Arjan Kuijper
Kristian Bredies
Thomas Pock
Horst Bischof (Eds.)

LNCS 7893

Scale Space and Variational Methods in Computer Vision

4th International Conference, SSVM 2013
Schloss Seggau, Leibnitz, Austria, June 2013
Proceedings

Springer

Lecture Notes in Computer Science 7893

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Arjan Kuijper Kristian Bredies
Thomas Pock Horst Bischof (Eds.)

Scale Space and Variational Methods in Computer Vision

4th International Conference, SSVM 2013
Schloss Seggau, Leibnitz, Austria, June 2-6, 2013
Proceedings

Springer

Volume Editors

Arjan Kuijper
Fraunhofer Gesellschaft
Institut für Graphische Datenverarbeitung
Fraunhoferstrasse 5
64283 Darmstadt, Germany
E-mail: arjan.kuijper@igd.fraunhofer.de

Kristian Bredies
University of Graz
Institute for Mathematics and Scientific Computing
Heinrichstrasse 36
8010 Graz, Austria
E-mail: kristian.bredies@uni-graz.at

Thomas Pock
Horst Bischof
Graz University of Technology
Institute for Computer Graphics and Vision
Inffeldgasse 16
8010 Graz, Austria
E-mail: {pock, bischof}@icg.tugraz.at

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-38266-6 e-ISBN 978-3-642-38267-3
DOI 10.1007/978-3-642-38267-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013937506

CR Subject Classification (1998): I.4, I.5, I.3.5, I.2.10, G.1, F.2.2

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

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

The 4th International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 2013) was held in Schloss Seggau, Leibnitz, in the vicinity of Graz, Austria. The biannual SSVM Conferences started in 2007 in Ischia, Italy (2007), and were followed by editions in Voss, Norway (2009), and Ein Gedi, Israel (2011).

This series of conferences originated from the biannual conferences on Scale Space held in 1997 in Utrecht, The Netherlands, and Variational, Geometric, and Level set Methods (VLSM) in 2001 in Vancouver, Canada. The aim of SSVM is to bring together these two different communities with common research interests: the one on scale space analysis and the one on variational, geometric, and level set methods and their applications in image interpretation and understanding. Just as in previous editions, the papers in these proceedings depict this successful combination.

Following the tradition of the previous SSVM conferences, we invited outstanding scientists to give keynote presentations and were happy to welcome:

- Gabriel Peyré (CNRS, CEREMADE, Université Paris-Dauphine): Inverse Problem Regularization with Weakly Decomposable Priors
- Martin Rumpf (University of Bonn): Variational Time Discretization of Geodesic Calculus in Shape Space
- Tony Lindeberg (KTH Royal Institute of Technology): A Framework for Invariant Visual Operations Based on Receptive Field Responses

From the 69 submitted papers, 19 were selected to be presented orally and 23 as posters. We would like to thank the authors for their contributions and the members of the Program Committee for their dedication and timely reviews.

We would like to sincerely thank Christine Haas from the sterreichische Computer Gesellschaft (OCG), Christiane Tronigger from Nethotels, and Sabine Tschernegg from Schloss Seggau for their help with the local arrangements.

March 2013

Arjan Kuijper
Kristian Bredies
Thomas Pock
Horst Bischof

Organization

Conference Chairs

Arjan Kuijper	Fraunhofer IGD, Germany
Kristian Bredies	University of Graz, Austria
Thomas Pock	Graz University of Technology, Austria
Horst Bischof	Graz University of Technology, Austria

Program Committee

Members

Luis Alvares	Universidad de Las Palmas de Gran Canaria, Spain
Jean-François Aujol	University of Bordeaux, France
Michael Breuß	BTU Cottbus, Germany
Thomas Brox	University of Freiburg, Germany
Freddy Bruckstein	Technion, Israel
Andres Bruhn	University of Stuttgart, Germany
Antonin Chambolle	Ecole Polytechnique, CMAP, France
Raymond Chan	Chinese University of Hong Kong, SAR China
Laurent Cohen	Ceremade, France
Remco Duits	Eindhoven University, The Netherlands
Jalal Fadili	ENSICAEN, France
Michael Felsberg	Linkopings Universitet, Sweden
Luc Florack	Eindhoven University of Technology, The Netherlands
Lewis Griffin	University College London, UK
Atsushi Imiya	Chiba University, Japan
SungHa Kang	Georgia Tech, USA
Ron Kimmel	Technion, Israel
Nahum Kiryati	Tel Aviv University, Israel
Stefan Kunis	University of Osnabrueck, Germany
François Lauze	University of Copenhagen, Denmark
Antonio Leitao	Federal University of Santa Catarina, Brazil
Dirk Lorenz	University of Braunschweig, Germany
Étienne Mémin	IRSIA, France
Jan Modersitzki	University of Lübeck, Germany
Mila Nikolova	Ecole Normale Superieur Cachan, France
Stanley Osher	UCLA, USA
Nikos Paragios	Ecole Centrale de Paris, France

Guy Rosman	Technion, Israel
Martin Rumpf	University of Bonn, Germany
Chen Sagiv	SagivTech Ltd., Israel
Otmar Scherzer	University of Vienna, Austria
Christoph Schnörr	University of Heidelberg, Germany
Carola Schönlieb	University of Cambridge, UK
Fiorella Sgallari	University of Bologna, Italy
Jon Sporring	University of Copenhagen, Denmark
Kim Steenstrup Pedersen	University of Copenhagen, Denmark
Gabriele Steidl	University of Kaiserslautern, Germany
Xue-Cheng Tai	University of Bergen, Norway
Bart ter Haar Romeny	Eindhoven University of Technology, The Netherlands
Joachim Weickert	Saarland University, Germany
Gershon Wolansky	Technion, Israel

Table of Contents

Image Denoising and Restoration

Image Enhancement and Texture Synthesis

Optical Flow and 3D Reconstruction

Scale Space and Partial Differential Equations

Image and Shape Analysis, Segmentation

Targeted Iterative Filtering

Freddie Åström[1,2], Michael Felsberg[1,2],
George Baravdish[3], and Claes Lundström[2,4]

[1] Computer Vision Laboratory, Linköping University, Sweden
[2] Center for Medical Image Science and Visualization, Linköping University, Sweden
[3] Department of Science and Technology, Linköping University, Sweden
[4] Sectra AB, Sweden
{freddie.astrom,michael.felsberg,george.baravdish,claes.lundstrom}@liu.se

Abstract. The assessment of image denoising results depends on the respective application area, *i.e.* image compression, still-image acquisition, and medical images require entirely different behavior of the applied denoising method. In this paper we propose a novel, nonlinear diffusion scheme that is derived from a linear diffusion process in a value space determined by the application. We show that application-driven linear diffusion in the transformed space compares favorably with existing nonlinear diffusion techniques.

1 Introduction

Many image processing techniques such, as denoising algorithms, aim to improve the quality of images. Naturally, the definition of quality is dependent on the situation where the images are used. The focus of this work is on denoising algorithms and our approach concentrates on that noise that actually will be visible to an observer, rather than data noise in general.

A widely applied denoising technique was introduced by Perona and Malik [1] who proposed a nonlinear partial differential equation (PDE) diffusion scheme. It extends the linear diffusion scheme which is based on the image gradient ∇u with an edge stopping function $g(|\nabla u|)$ *i.e.*

$$\begin{array}{ccc} \nabla u & \rightarrow & g(|\nabla u|)\nabla u \\ \text{Linear} & & \text{Perona and Malik} \end{array}$$

where a modification of the diffusion speed is based on the value domain of the image gradient. Another PDE model which has received much attention in recent years is the tensor-based diffusion scheme of Weickert [2]. These diffusion models require the determination of parameters often estimated from the input data. Thus the performance of these methods depend on the accuracy of the parameter estimation. A particular problem is that image structure of different scale can be present within the same value ranges, hence spatially varying contrast parameters are required.

In this work we show that by the use of an application dependent transformation to the input data space given by a function $m(u)$, we obtain a nonlinear

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 1–11, 2013.

diffusion formulation. The novel formulation modifies the value domain of u rather than the gradient domain as done in Perona and Malik diffusion *i.e.*

$$\begin{array}{ccc} \nabla u & \rightarrow & \nabla m(u) \\ \text{Linear} & & \text{Targeted diffusion} \end{array}$$

An energy functional is formulated where the regularization term is expressed using the mapping function $m(u)$ and the resulting Euler-Lagrange equation can be interpreted in terms of nonlinear diffusion. The difference between the edge-stopping function g and the mapping function m is that the former is an data-driven ad-hoc selection whereas m is application-driven.

Image processing tools that target specific regions of an image are relevant in many areas of computer vision, and include high dynamic range imaging [3,4], infrared imaging [5] and medical imaging [6]. One such region based diffusion filtering method was proposed by Kačur et al. [7] who generalized the Perona and Malik diffusion. They model the diffusion PDE with an additional nonlinear function on the range domain from which the gradient is computed. This allows the filtering process to be directed to regions containing particular image structures. Their framework reduces the filtering process in regions determined by the user, but the method still requires the determination of a parameter corresponding to an edge-stopping function within the region of filtering.

In this work our main contributions are

- A novel diffusion scheme is derived by using a mapping function in a variational formulation of standard image diffusion.
- Necessary and sufficient conditions are derived to determine if the solution given by the Euler Lagrange equation yield a minimum of the proposed energy functional.
- We show how the mean and variance of noise present in the signal domain is transformed by the mapping function.
- In experiments with computed tomography (CT) images of different noise levels, it is shown that the novel scheme compares favorably to nonlinear scalar diffusion on a data set of 400 images using the structural similarity index [8].

2 Image Diffusion

2.1 Linear Diffusion

The variational approach to isotropic image diffusion is to minimize the energy functional

$$E(u) = \int_{\Omega} (u - u^0)^2 \, d\boldsymbol{x} + \lambda \int_{\Omega} |\nabla u|^2 \, d\boldsymbol{x} \ , \tag{1}$$

where $\boldsymbol{x} \in \Omega$ and u^0 denotes the observed image. The constant λ is a positive scalar which determines the effect of the regularization. The domain Ω is a grid described by the image size in pixels, and $\nabla = \partial_{\boldsymbol{x}} = \left(\partial_{x_1}, ..., \partial_{x_n}\right)^T$ is the

gradient operator, and $\dim(\nabla) = n$ is the number of dimensions. Other types of regularization terms have previously been investigated [9,10]. To minimize $E(u)$, one finds the stationary point u by computing the Euler-Lagrange (E-L) equation

$$E_u(u) = 0 \quad \text{in} \quad \Omega, \quad \nabla u \cdot \boldsymbol{n} = 0 \quad \text{on} \quad \partial\Omega \ ,$$

where $\boldsymbol{n}$ is the normal vector on the boundary $\partial\Omega$. The E-L equation for (1) reads

$$\begin{cases} u - u^0 - \lambda \Delta u = 0 \text{ in } \ \Omega \\ \qquad\qquad \nabla u \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \end{cases} \tag{2}$$

where Δu is the Laplacian operator. We solve (2) by solving an initial value problem (IVP) and obtain the diffusion equation which has a closed form solution.

2.2 Nonlinear Diffusion

Before deriving the proposed diffusion scheme, we define the nonlinear scalar diffusion process of Perona and Malik (PM) [1] as

$$\begin{cases} u - u^0 - \lambda \operatorname{div}(g(|\nabla u|)\nabla u) = 0 \text{ in } \ \Omega \\ \qquad\qquad\qquad\qquad \nabla u \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \end{cases} \tag{3}$$

where $g(s) = (1 + (s/k)^2)^{-1}$ is a popular choice as the diffusivity function and k is a contrast parameter fixed to suppress the flux at edges and lines in the image. It will be seen that the diffusion process introduced in the subsequent section can be viewed as a nonlinear filter, closely related to PM-diffusion. We solve (3) by solving an IVP and obtain the diffusion equation.

Tensor-based nonlinear diffusion is achieved defining $T = w * \nabla u \nabla u^T$ where $*$ is a convolution operator and w is a Gaussian filter [2,11,12]. Then the diffusion tensor can be computed as $D(T) = O^T g(\Lambda) O$ where O are the eigenvectors and Λ the eigenvalues of T [13]. This gives the PDE

$$u - u^0 - \lambda \operatorname{div}(D(T)\nabla u) = 0 \quad . \tag{4}$$

3 Targeted Iterative Filtering

In order to simultaneously consider the signal domain and the application dependent transformation of an image, we express the regularization term of the energy functional (1) in the transformed domain. Let $m(u(\boldsymbol{x}))$ be a *mapping function* that maps $u(\boldsymbol{x})$ to its application domain, then define

$$E(u) = \int_\Omega (u - u^0)^2 d\boldsymbol{x} + \lambda \int_\Omega |\nabla m(u)|^2 d\boldsymbol{x} \tag{5}$$

where $m(u) \in \mathcal{C}^3(\Omega)$ and $\lambda > 0$ is a parameter determining the influence of the regularization term. In the subsequent sections we derive the necessary and sufficient conditions for the functional $E(u)$ to attain a local minimum (for details see supplementary material).

3.1 Necessary Conditions for Local Minimum

The variational derivative of the the regularization term of $E(u)$ is computed using the Gâteaux derivative

$$\langle \partial R, v \rangle = \lim_{\varepsilon \to 0} \frac{|\nabla m(u + \varepsilon v)|^2 - |\nabla m(u)|^2}{\varepsilon} \ ,$$

where $v \in \mathcal{C}^1(\Omega)$ is an arbitrary function such that $\partial_n v|_{\partial\Omega} = 0$. Using the chain rule $\nabla m(u) = m'(u)\nabla u$ we obtain

$$\langle \partial R, v \rangle = \lim_{\varepsilon \to 0} \frac{|\nabla u|^2 (m'(u+\varepsilon v)^2 - m'(u)^2) + m'(u+\varepsilon v)^2 (\varepsilon 2 \nabla u^t \nabla v + \varepsilon^2 |\nabla v|^2)}{\varepsilon}$$

With Green's identity and Neumann boundary conditions we obtain

$$\langle \partial R, v \rangle = \left(2|\nabla u|^2 m'(u) m''(u) - 2\mathrm{div}(m'(u)^2 \nabla u^t) \right) v \ .$$

Now observe that $\mathrm{div}(m'(u)^2 \nabla u) = 2m'(u)m''(u)|\nabla u|^2 + m'(u)^2 \Delta u$. Using this result, and since $v \neq 0$, the E-L equation reads

$$\begin{cases} u - u^0 - \lambda(\mathrm{div}(m'(u)^2 \nabla u) + m'(u)^2 \Delta u) = 0 \text{ in } \ \Omega \\ \qquad\qquad\qquad\qquad\qquad\qquad m'(u)^2 \nabla u \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \end{cases} \tag{6}$$

Since $m'(u)^2 \geq 0$ it is guaranteed that a solution of (6) exists. Compared to (3), the divergence operator is modulated with the squared steepness of the mapping function. Also, the Laplacian is weighted with the same factor. If and only if m is a globally linear function, (6) becomes identical to (2). The difference to nonlinear diffusion is easiest explained in terms of the Lagrangian: Replacing $g(|\nabla u|)$ with $m'(u)^2$ means to replace the robust error function with an intensity dependent factor.

3.2 Sufficient Conditions for Local Minimum

In this section we derive sufficient conditions for the solution of the E-L equation to be a minimum of the regularization term in (5). The result is summarized in the theorem below. We remark that if the mapping function is a strict monotone function, the regularization term in (5) is obviously convex and the necessary condition is also a sufficient condition. However, in the general case, m is not always a strict monotone function, and this is the case we consider here.

Theorem 1. *Let u^0 be an observed image in a domain $\Omega \subset \mathbb{R}^2$, and denote by $E(u)$ the functional*

$$E(u) = \int_\Omega (u - u^0)^2 \, d\boldsymbol{x} + \lambda \int_\Omega |\nabla m(u)|^2 \, d\boldsymbol{x}$$

where $u \in \mathcal{C}^2$ and $m(u) \in \mathcal{C}^3$. Let $\varepsilon > 0$ be arbitrary and consider the set

$$B_\varepsilon = \left\{ h, \nabla h \in L^2(\Omega) : \ ||h||^2_{L^2(\Omega)} \leq \varepsilon^2 / C_M, \quad ||\nabla h||^2_{L^2(\Omega)} \geq \varepsilon \right\}$$

where

$$C_M = \max_{\boldsymbol{x}\in\Omega} \left|[m'(u^*(\boldsymbol{x}))m'''(u^*(\boldsymbol{x})) - 3(m''(u^*(\boldsymbol{x})))^2]|\nabla u^*(\boldsymbol{x})|^2\right| \ .$$

Then u^* is a local minimum of $E(u)$ given by the solution of the E-L equation (6) if there exists $\boldsymbol{\xi} \in \Omega$ such that

$$(m'(u^*(\boldsymbol{\xi})))^2||\nabla h||^2_{L^2(\Omega)} > \varepsilon^2 \ , \tag{7}$$

for every $h \in B_\varepsilon$.

Proof. In order to find the sufficient condition for a minimum, define the regularization term in the functional as

$$J(u) = \int_\Omega |\nabla m(u)|^2 \, d\boldsymbol{x} = \int_\Omega (m'(u))^2 |\nabla u|^2 \, d\boldsymbol{x}$$

Given a function $\varphi \in \mathcal{C}^3$, a third order Taylor expansion at the point 0 is

$$\varphi(a) - \varphi(0) = a\varphi'(0) + \frac{a^2}{2}\varphi''(0) + \frac{a^3}{6}\varphi'''(a\theta) \quad 0 < \theta < 1 \ .$$

Let $h \in \mathcal{C}^1$, then define $\varphi(a) = J(u + ah)$, which determines the first variation δJ of $J(u)$ as

$$\delta J = \lim_{a\to 0} \frac{J(u+ah) - J(u)}{a} = \lim_{a\to 0} \frac{\varphi(a) - \varphi(0)}{a} = \varphi'(0)$$

In the same way the second variation $\delta^2 J$ follows. Since δJ is a linear functional in h and $\delta^2 J$ is a quadratic form in h define $L_1(h) = \delta J = \varphi'(0)$ and $L_2(h,h) = \delta^2 J = \varphi''(0)$. Given that φ is differentiable then so is J. If $a = 1$ then the Taylor expansion is given by

$$J(u(x) + h(x)) - J(u) = L_1(h) + L_2(h,h) + ||h||^2 \rho(h), \tag{8}$$

where $\rho(h) \to 0$, as $h \to 0$.

A necessary condition of u^* to be a minimum point of the functional $J(u)$ is

$$\varphi'(0) = L_1(h) = 2\int_\Omega [m'(u^*)m''(u^*)|\nabla u^*|^2 h + (m'(u^*))^2 \nabla u^* \cdot \nabla h] \, d\boldsymbol{x} = 0 \tag{9}$$

for every h in a neighborhood of u^*. According to the E-L equation the solution u^* must satisfy that

$$m'(u^*) \neq 0 \tag{10}$$

otherwise the trivial solution $J(u^*) = 0$ is obtained. Differentiating $\varphi'(a)$ and rewriting the E-L equation using condition (10) obtain $L_2(h,h)$ as

$$\frac{1}{2}L_2(h,h) = \int_\Omega [m'(u^*)m'''(u^*) - 3(m''(u^*))^2]|\nabla u^*|^2 h^2 \, d\boldsymbol{x} + \int_\Omega (m'(u^*))^2 |\nabla h|^2 \, d\boldsymbol{x}$$

Since $L_2(h,h) > 0$ implies a minimum, we consider the first integral. Assume $m \in \mathcal{C}^3$ and $u \in \mathcal{C}^1$, then there is an upper bound $C_M > 0$ such that

$$|[m'(u^*)m'''(u^*) - 3(m''(u^*))^2]|\nabla u^*|^2| \leq C_M.$$

Let $\varepsilon > 0$ and B_ε be a set defined by

$$B_\varepsilon = \left\{ h, \nabla h \in L^2(\Omega) : \ ||h||^2_{L^2(\Omega)} \leq \varepsilon^2/C_M, \quad ||\nabla h||^2_{L^2(\Omega)} \geq \varepsilon \right\}$$

Given that $h \in B_\varepsilon$, then the first integral of $L_2(h,h)$ reads

$$\int_\Omega [m'(u^*)m'''(u^*) - 3(m''(u^*))^2]|\nabla u^*|^2 h^2 \, d\boldsymbol{x} \geq -C_M \int_\Omega h^2 \, d\boldsymbol{x} \geq -\varepsilon^2.$$

Since $h \in B_\varepsilon$ we have

$$\int_\Omega (m'(u^*))^2 |\nabla h|^2 \, d\boldsymbol{x} \neq 0 \ .$$

By the mean value theorem of calculus there exists a $\boldsymbol{\xi} \in \Omega$ such that $m'(u^*(\boldsymbol{\xi})) \neq 0$ and

$$\int_\Omega (m'(u^*))^2 |\nabla h|^2 \, d\boldsymbol{x} = m'(u^*(\xi))^2 ||\nabla h||^2_{L^2(\Omega)}$$

Hence

$$\begin{aligned} L_2(h,h) &\geq 2 \int_\Omega (m'(u^*))^2 |\nabla h|^2 \, d\boldsymbol{x} - 2\varepsilon^2 \geq 2(m'(u^*(\boldsymbol{\xi})))^2 ||\nabla h||^2_{L^2(\Omega)} - 2\varepsilon^2 \\ &> 2\varepsilon \left[(m'(u^*(\boldsymbol{\xi})))^2 - \varepsilon \right] > 0 \end{aligned} \tag{11}$$

since $h \in B_\varepsilon$ and we can always chose $\varepsilon < (m'(u^*(\boldsymbol{\xi})))^2$ which is the sufficient condition for u^* to be a local minimum of $J(u)$. And the theorem follows. □

4 Noise Estimation in the Transformed Domain

Due to the nonlinear mapping function, m, it is of interest to investigate the transformation of the first and second statistical moments of the input signal. We assume that the image signal can be described by a linear model

$$u^0 = u_0 + \eta \ ,$$

where $\eta \sim \mathcal{N}(\mu, \sigma^2)$ and u^0 is the observed signal, u_0 is the noise-free signal and η is a noise component normally distributed with mean μ and variance σ^2. The mean value and the variance are estimated using a second order Taylor series of the mapping function, then the mean value and variance estimates

$$\hat{\mu}_m = m(u_0 + \mu) + \frac{1}{2} m''(u_0 + \mu) \, \sigma^2 \tag{12}$$

$$\hat{\sigma}^2_m = \Psi[m](u_0 + \mu) \, \sigma^2 \tag{13}$$

where

$$\Psi[m](u_0 + \mu) = m'(u^0)^2 - m(u^0)m''(u^0)$$

is the energy operator [14]. This shows that the mean value in the transformed domain will depend on the curvature of the transformation used, implying that the mapping will not preserve the average intensity level of the input space. Also that the noise variance in the signal domain is amplified by the energy operator. For complete derivations see supplementary material.

5 Application to Medical Imaging

For the purpose of evaluating the proposed application-driven diffusion scheme we consider the application of medical visualization. We make no claim on superiority over existing techniques in medical visualization, merely we limit ourselves to diffusion methods. The diffusion methods investigated are, the novel targeted filtering scheme (TF), linear diffusion (LD), nonlinear diffusion (PM) and tensor-based image diffusion (AD).

Visualizations in medical imaging are computed by transfer functions, which usually are piecewise linear [6]. However, sufficiently similar functions produce visualizations that are visually indistinguishable. We use combinations of sigmoid functions, see Fig. 1, since they are three times continuously differentiable.

5.1 Selection of Mapping Function

Let $m : \mathbb{R}^2 \to [0, 1]$ be the visualization mapped using a transfer function $m \in \mathcal{C}^3$ computed from two user defined thresholds $u(\boldsymbol{x}) = u_1$ to $u(\boldsymbol{x}) = u_2$. We define a sigmoid function

$$m(u(\boldsymbol{x}), a, b) = (1 + \exp(-(u(\boldsymbol{x}) - b)/a))^{-1} \ , \tag{14}$$

where $a = (u_2 - u_1)/4$ is the steepness of the sigmoid function and $b = (u_1 + u_2)/2$ defines the offset. For this choice of mapping function we show that the sufficient condition in (7) is satisfied. Then the lower bound of $(m'(u^*))^2$ is given by

$$(m'(u^*))^2 = \frac{1}{a^2} \frac{e^{\frac{2(b-u^*)}{a}}}{(1 + e^{\frac{b-u^*}{a}})^4} \geq \frac{1}{a^2} \frac{e^{\frac{2(b-1)}{a}}}{(e^{\frac{b}{a}} + e^{\frac{b}{a}})^4} \geq \frac{1}{a^2 16 e^{\frac{2(b+1)}{a}}}$$

thus the condition (11) is replaced with $(16a^2 e^{\frac{2(b+1)}{a}})^{-1} > \varepsilon$. Details on the determination of the lower bound can be found in the supplementary material.

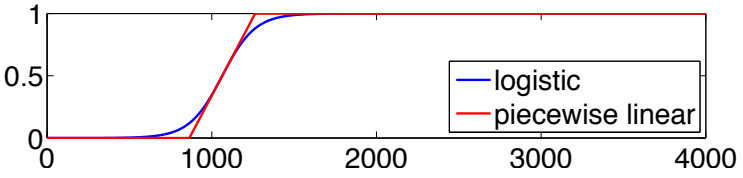


Fig. 1. Example of mapping function

5.2 Numerical Aspects

The derived E-L equation (6) is solved as an IVP problem discretizised using a standard forward Euler scheme. A forward and backward finite difference scheme is used to approximate the image derivatives.

The derivatives of the mapping function, m are computed analytically. However, before evaluating the derivatives of $m(u)$, the signal u is regularized with a small Gaussian filter. Also to remedy the fact that different propagation speeds are obtained for different slopes of the mapping function, derivatives are normalized to attain a global maximum of 1. The implementation is available here [15].

5.3 Experiment Setup

For the evaluation, we add zero mean Gaussian noise to a set of computed tomography (CT) images. The motivation for using additive noise is due to the projection data obtained from the CT scanner contains multiplicative noise. In the CT reconstruction the logarithm of the data is taken, thus multiplicative noise can be modeled as additive noise. All images were scaled to an 8-bit quantisation representation and zero mean Gaussian noise with standard deviation $\sigma = \{5, 10, 15\}$ was added to the test images in the signal domain. According to (13), the noise levels in the visualization domain is $\sigma_m = \{51.19, 102.38, 153.36\}$ using a mapping function with endpoints $u_1 = 864$ and $u_2 = 1264$ where the endpoints are represented using Hounsfield units.

All diffusion methods were set to iterate the solution until the peak signal to noise (PSNR) value no longer increases. The steplength was set to 0.05 for all methods except for the proposed method which utilizes the slope of the mapping function as its steplength $\lambda = \min(1/(u_2 - u_1)), 0.25)$ where 0.25 is the maximum steplength to ensure stability in the case of linear diffusion [2].

The PM and AD contrast parameter was set using the estimated noise levels σ_{est} based on [13] and computed according to [16] as $k = (e-1)(e-2)^{-1}\sigma_{est}^2$.

The peak signal to noise measure (PSNR) and the structural similarity index (SSIM) [8] was used to evaluate the performance of the proposed algorithm.

5.4 Results

Table 1 shows the SSIM and PSNR values obtained in the visualization domain for a dataset of 400 CT images. Comparing the filtering methods with respect to the error measures, then the error values are in favor of the proposed targeted filtering method (TF) higher noise levels. Here it is important to note the fundamental difference between TF and PM. The performance of PM is determined based the estimation of a contrast parameter for the nonlinear mapping function, whereas TF is not. The only parameter required to be determined in TF (as with all iterative methods) is the stopping time to avoid trivial solutions. Thus, disregarding the stopping time, *TF is a non-parametric non-linear diffusion scheme* which behaves similarly to PM diffusion.

Table 1. SSIM and PSNR values. $\hat{\sigma}_m$ was computed according to (13).

	σ	$\hat{\sigma}_m$	LD	PM	TF	AD
SSIM	5	51.19	0.89 ± 0.005	0.92 ± 0.004	0.93 ± 0.003	0.94 ± 0.003
	10	102.38	0.84 ± 0.004	0.87 ± 0.005	0.89 ± 0.005	0.87 ± 0.006
	15	153.56	0.82 ± 0.006	0.83 ± 0.005	0.86 ± 0.006	0.82 ± 0.005
PSNR	5	51.19	28.44 ± 0.37	30.82 ± 0.56	30.76 ± 0.51	32.18 ± 0.72
	10	102.38	25.92 ± 0.45	27.68 ± 0.49	28.13 ± 0.53	27.88 ± 0.49
	15	153.56	24.81 ± 0.54	25.74 ± 0.53	26.82 ± 0.59	25.38 ± 0.51

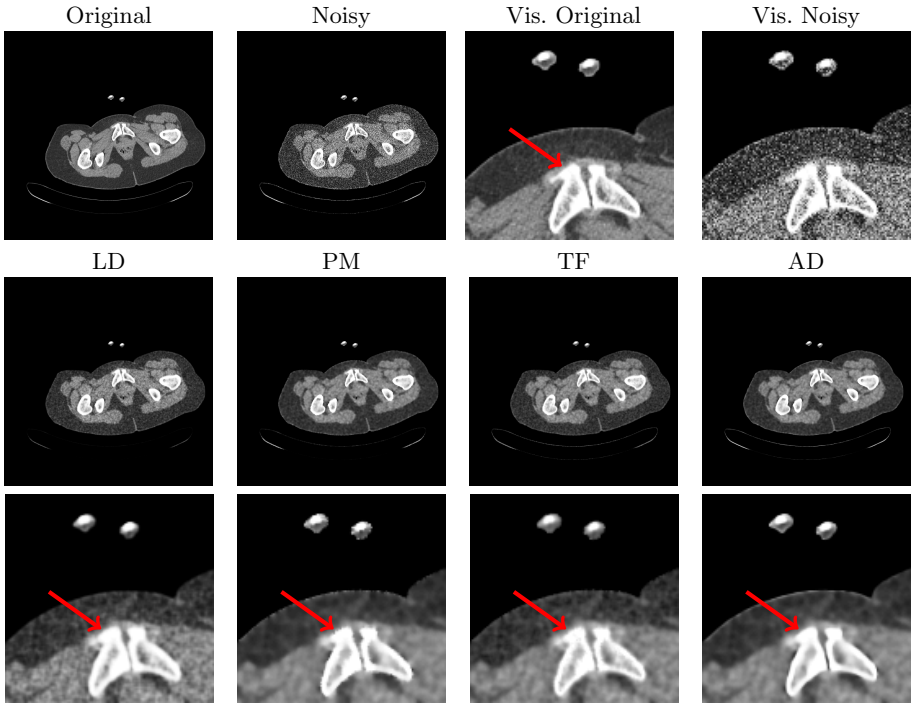


Fig. 2. Slice 250. Noise level $\sigma = 5$. *Details best viewed on monitor.*

Figure 2 and 3 visualize the corresponding images of slices 250 with noise level $\sigma = 5$ and 350 with noise level $\sigma = 10$. In addition to the visualizations, respective details are depicted. Visually, the proposed diffusion scheme produces superior results close to edges compared to LD and PM diffusion indicated by the arrows in both figures. LD oversmooths the image and PM simply retains noise close to edges. AD preserves edges well and produces high PSNR and SSIM values but approximately homogeneous regions appear oversmoothed. In Fig. 3 it is clear that regions indicated by the arrows have been retained in the proposed method whereas the other diffusion techniques have removed the structure.

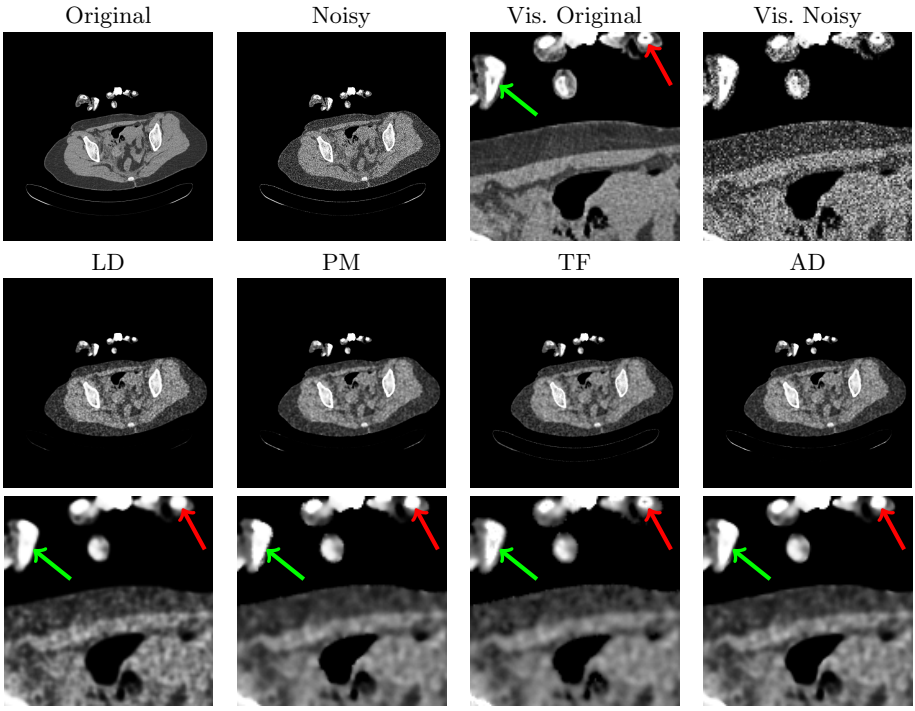


Fig. 3. Slice 350. Noise level $\sigma = 10$. *Details best viewed on monitor.*

6 Conclusion

The performance of image denoising methods has to be assessed with respect to the respective application. In our case, we considered the application of denoising medical images and limit ourselves to diffusion methods. The relevant quality criteria is the result of the visualization after applying a mapping function. We have used the mapping function to derive a novel nonlinear diffusion scheme for targeted iterative diffusion and evaluated the method on a data set of CT images with different noise levels. The proposed method is non-parametric in the sense that it is application-driven rather than data-driven.

Acknowledgment. This research has received funding from the Swedish Research Council through a grant for the project *Visualization-adaptive Iterative Denoising of Images*, from the ECs 7th Framework Programme (FP7/ 2007-2013), grant agreement 247947 (GARNICS). From Vinnova project *Online laboratory for medical image analysis*, and Swedish Foundation for Strategic Research, grant SM10-0022. We thank Hanno Scharr at Forschungszentrum Jülich, Germany, for discussion on the implementation of the anisotropic diffusion scheme.

References

1. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions, PAMI 12, 629–639 (1990)
2. Weickert, J.: Anisotropic Diffusion in Image Processing. ECMI Series. Teubner-Verlag, Stuttgart (1998)
3. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: SIGGRAPH 1997, pp. 369–378 (1997)
4. DiCarlo, J.M., Wandell, B.A.: Rendering high dynamic range images. In: Proceedings of the SPIE: Image Sensors, vol. 3965, pp. 392–401 (2000)
5. Vollmer, M., Möllmann, K.: Infrared Thermal Imaging: Fundamentals, Research and Applications. John Wiley & Sons (2010)
6. Prokop, M., Galanski, M.: Spiral and Multislice Computed Tomography of the Body. Thieme Verlag (2003)
7. Kačur, J., Mikula, K.: Slowed anisotropic diffusion. In: ter Haar Romeny, B., Florack, L., Koenderink, J., Viergever, M. (eds.) Scale-Space 1997. LNCS, vol. 1252, pp. 357–360. Springer, Heidelberg (1997)
8. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. 13(4), 600–612 (2004)
9. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D 60(1-4), 259–268 (1992)
10. Baravdish, G., Svensson, O.: Image reconstruction with p(x)-parabolic equation. In: ICIPE 2011, Orlando Florida (2011)
11. 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)
12. Bigun, J., Granlund, G.H.: Optimal Orientation Detection of Linear Symmetry. In: Proceedings of the IEEE First International Conference on Computer Vision, pp. 433–438 (1987)
13. Felsberg, M.: Autocorrelation-driven diffusion filtering. IEEE Transactions on Image Processing 20(7), 1797–1806 (2011)
14. Felsberg, M., Jonsson, E.: Energy tensors: Quadratic, phase invariant image operators. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 493–500. Springer, Heidelberg (2005)
15. Åström, F.: Implementation of Targeted Iterative Filtering. In: SSVM 2013 (2013), `http://liu.diva-portal.org/smash/record.jsf?pid=diva2:608779`
16. Förstner, W.: Image preprocessing for feature extraction in digital intensity, color and range images. In: Geomatic Method for the Analysis of Data in the Earth Sciences. LNES, vol. 95, pp. 165–189 (2000)

Generalized Gradient on Vector Bundle – Application to Image Denoising

Thomas Batard and Marcelo Bertalmío*

Department of Information and Communication Technologies
University Pompeu Fabra, Barcelona, Spain
{thomas.batard,marcelo.bertalmio}@upf.edu

Abstract. We introduce a gradient operator that generalizes the Euclidean and Riemannian gradients. This operator acts on sections of vector bundles and is determined by three geometric data: a Riemannian metric on the base manifold, a Riemannian metric and a covariant derivative on the vector bundle. Under the assumption that the covariant derivative is compatible with the metric of the vector bundle, we consider the problems of minimizing the L2 and L1 norms of the gradient. In the L2 case, the gradient descent for reaching the solutions is a heat equation of a differential operator of order two called connection Laplacian. We present an application to color image denoising by replacing the regularizing term in the Rudin-Osher-Fatemi (ROF) denoising model by the L1 norm of a generalized gradient associated with a well-chosen covariant derivative. Experiments are validated by computations of the PSNR and Q-index.

Keywords: Generalized gradient, Riemannian manifold, Vector bundle, Total variation, Color image denoising, Rudin-Osher-Fatemi model.

1 Introduction

Total variation regularization methods have been widely used for image denoising tasks. Given an image $I_0 \colon \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R} \in BV(\Omega)$ corrupted by additive white Gaussian noise of standard deviation σ, the seminal model of Rudin-Osher-Fatemi (ROF) [19] estimates the denoised image as the solution of the following variational problem

$$\underset{I \in BV(\Omega)}{\arg\min} \int_\Omega \frac{1}{2}\lambda \left(I - I_0\right)^2 + \|\nabla I\| \, d\Omega \tag{1}$$

where λ is a tuning parameter. The first term in formula (1) is the attached data term and the second one is the regularizing term. Since then, this model has been extended in several ways (see e.g. [4],[9],[14],[15],[16],[17],[18],[24],[25], for local methods based on a modification of the regularizing term, and [8],[10] for nonlocal methods).

* This work was supported by European Research Council, Starting Grant ref. 306337. The second author acknowledges partial support by Spanish grants AACC, ref. TIN2011-15954-E, and Plan Nacional, ref.TIN2012-38112.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 12–23, 2013.

In this paper, we construct a new regularizing term from a generalization of the Euclidean and Riemannian gradient operators, as well as the Jacobian, on vector bundles. Then, the ROF denoising model based on this new operator generalizes the Euclidean approach of [19] and its multidimensional extension [4], as well as the Riemannian ROF denoising model in [17]. The key idea is to treat the term ∇I as a vector-valued differential 1-form $\nabla^E I$, that we call **connection gradient** of I, where the operator ∇^E is a covariant derivative (also called connection). Given Riemannian metrics on the base manifold and vector bundle, a metric on the space of vector-valued differential 1-forms might be constructed, and consequently the norm of the connection gradient $\nabla^E I$ might be considered. Then, for particular choices of metrics and covariant derivative, the norm of $\nabla^E I$ corresponds to the norm of the Euclidean or Riemannian gradient.

In this paper, we focus on connection gradients where the covariant derivative is compatible with the metric of the vector bundle. In this context, the covariant derivative ∇^E has an adjoint operator ∇^{E^*} and we show that both L1 and L2 norms minimization problems extend the Euclidean and Riemannian approaches in a natural way. Indeed, we show that the gradient descent flow for reaching the sections minimizing the L2 norm of the connection gradient is the heat equation of a generalized Laplacian. Moreover, we show that the critical points of the L1 norm of the connection gradient satisfy the equation $\nabla^{E^*}\left(\nabla^E I/\|\nabla^E I\|\right) = 0$.

The outline of the paper is the following. Sect. 2 is mainly theoretical. We first introduce the notion of connection gradient and its norm. Then, we restrict to the case where the covariant derivative is compatible with the metric of the vector bundle, and consider the L1 and L2 norms minimization of the connection gradient. In Sect. 3, we present an application to color image denoising by considering the L1 norm of a suitable connection gradient as the regularizing term of a ROF denoising model. We test our denoising method on the Kodak database [11] and compute both PSNR and Q-index [23]. Results show that our method provides better results than the split Bregman method [9] applied to ROF functional.

2 Generalized Gradient on Vector Bundle

2.1 Definitions and Examples

We refer to [21] for an introduction to differential geometry of fiber bundles. Given a vector bundle E, we denote by $\Gamma(E)$ the set of smooth sections of E.

Connection Gradient

Definition 1. *Let E be a vector bundle of rank m over a Riemannian manifold (M, g) of dimension n. Let ∇^E be a covariant derivative and h be a definite positive metric on E. Given $\varphi \in \Gamma(E)$, we call the term $\nabla^E\varphi \in \Gamma(T^*M \otimes E)$ the **connection gradient** of φ.*

The metrics g on TM and h on E induce a definite positive metric $\langle\ ,\ \rangle$ on $T^*M \otimes E$. Then, we define the norm of the connection gradient of φ as

$$\|\nabla^E \varphi\| := \sqrt{\langle \nabla^E \varphi, \nabla^E \varphi \rangle} = \sqrt{\sum_{i,j=1}^{n} g^{ij}\, h\left(\nabla^E_{\partial/\partial x_i} \varphi, \nabla^E_{\partial/\partial x_j} \varphi\right)} \tag{2}$$

where $(\partial/\partial x_1, \cdots, \partial/\partial x_n)$ is the frame of TM induced by a coordinates system $(x_1, \cdots, x_n)$ of M.

Example 1. Let $E = C^\infty(M)$ be the vector bundle of rank 1 of smooth functions on a Riemannian manifold (M, g). Let ∇^E be the trivial covariant derivative on E and h be the definite positive metric on E given by the scalar multiplication. Then, the connection gradient of a function f is its differential $df \in \Gamma(T^*M)$.

The musical isomorphism $\sharp\colon T^*M \longmapsto TM$ maps df onto the Riemannian gradient $\nabla_g f$ of f, of components $g^{ij}\, \partial f/\partial x_i$ in the frame $\{\partial/\partial x_j\}$. Moreover the norm of df coincides with the norm of $\nabla_g f$ since we have

$$\|df\| := \sqrt{\langle df, df \rangle} = \sqrt{\sum_{i,j=1}^{n} g^{ij} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j}}, \tag{3}$$

Connection Compatible with the Metric

Definition 2. *Let E be a vector bundle over a Riemannian manifold (M, g), equipped with a definite positive metric h. A covariant derivative ∇^E on E is compatible with the metric h if it satisfies*

$$dh(\varphi, \psi) = h(\nabla^E \varphi, \psi) + h(\varphi, \nabla^E \psi) \tag{4}$$

for any $\varphi, \psi \in \Gamma(E)$.

Example 2. On the vector bundle of smooth functions on a Riemannian manifold, the trivial covariant derivative is compatible with the metric given by the scalar multiplication on the fibers.

Assuming that E is associated with the principal bundle $P_{\mathrm{SO}}(E)$ of orthonormal frame fields of E, we have the following result.

Proposition 1 (see e.g. Lawson et al. [13] (Prop. 4.4 p.103)). *There is a one-one correspondence between connection 1-forms on $P_{SO}(E)$ and covariant derivatives on E that are compatible with the metric.*

Under the choice of a local trivializing section of $P_{\mathrm{SO}}(E)$, i.e. a local orthonormal frame with respect to h of the vector bundle E, a connection 1-form is a $\mathfrak{so}(n)$-valued 1-form on M, i.e. $\omega \in \Gamma(T^*M \otimes \mathfrak{so}(n))$. More precisely, we have

$$\nabla^E_X \varphi = d_X \varphi + \omega(X)(\varphi) \tag{5}$$

for any $X \in \Gamma(TM)$.

Connection Laplacian. Let $\nabla^{T^*M\otimes E}$ be the covariant derivative on $T^*M\otimes E$ defined as

$$\nabla^{T^*M\otimes E}(\eta\otimes\varphi)=\nabla^{T^*M}\eta\otimes\varphi+\eta\otimes\nabla^E\varphi$$

where ∇^{T^*M} is the covariant derivative on T^*M induced by the Levi-Civita covariant derivative on (TM,g) and ∇^E is a covariant derivative on E compatible with a definite positive metric h. The **adjoint** $\nabla^{E^*}\colon \Gamma(T^*M\otimes E)\longrightarrow\Gamma(E)$ of the operator $\nabla^E\colon\Gamma(E)\longrightarrow\Gamma(T^*M\otimes E)$ is the operator

$$\nabla^{E^*}=-Tr\,\nabla^{T^*M\otimes E} \tag{6}$$

where Tr denotes the contraction with respect to the metric g. In others words, the following equality is satisfied

$$\int_M h(\nabla^{E^*}\eta,\varphi)\,dM=\int_M\langle\eta,\nabla^E\varphi\rangle\,dM \tag{7}$$

assuming that φ has compact support.

Example 3. On the vector bundle of smooth functions on a Riemannian manifold (M,g), the adjoint $d^*\colon\Gamma(T^*M)\longrightarrow C^\infty(M)$ of the trivial covariant derivative $d\colon C^\infty(M)\longrightarrow\Gamma(T^*M)$ is the operator

$$d^*\eta=-\sum_{i,j}\left(g^{ij}\partial_{x_i}\eta(\partial/\partial x_j)-\sum_k\Gamma_{ij}^k\,\eta(\partial/\partial x_k)\right)$$

where Γ_{ij}^k are the Christoffel symbols of (M,g).

Definition 3. *The **connection Laplacian** Δ^E is the second order differential operator on $\Gamma(E)$ defined as $\Delta^E=\nabla^{E^*}\nabla^E$.*

In the frame $(\partial/\partial x_1,\cdots,\partial/\partial x_n)$ of (TM,g), we have

$$\Delta^E=-\sum_{ij}g^{ij}\left(\nabla^E_{\partial/\partial x_i}\nabla^E_{\partial/\partial x_j}-\sum_k\Gamma_{ij}^k\nabla^E_{\partial/\partial x_k}\right)$$

Example 4. The **Laplace-Beltrami operator** Δ_g is the connection Laplacian (up to a sign) associated to the trivial covariant derivative d on the vector bundle of smooth functions on a Riemannian manifold (M,g), i.e.

$$\Delta_g=-\sum_{ij}g^{ij}\left(\partial_{x_i}\partial_{x_j}-\sum_k\Gamma_{ij}^k\,\partial_{x_k}\right)$$

2.2 L2 Minimization of Connection Gradient and Dirichlet Energy

Let E be a vector bundle over a Riemannian manifold (M,g) equipped with a definite positive metric h and a covariant derivative ∇^E compatible with h. We have the following result.

Proposition 2 (Lawson et al. [13] Prop. 8.1 p.154). *The operator Δ^E is non-negative and essentially self-adjoint. Furthermore,*

$$\int_M h(\Delta^E \varphi, \psi)\, dM = \int_M \langle \nabla^E \varphi, \nabla^E \psi \rangle\, dM \tag{8}$$

for all $\varphi, \psi \in \Gamma(E)$ provided that one of φ or ψ has compact support. If M is compact, then $\Delta^E \varphi = 0$ if and only if $\nabla^E \varphi = 0$.

We observe that the right term of equality (8) corresponds to the Gâteaux derivative in the direction ψ of the energy

$$E(\varphi) = \int_M \|\nabla^E \varphi\|^2 dM \tag{9}$$

Hence the critical points of the energy (9) satisfy $\Delta^E \varphi = 0$. Moreover, they correspond to the minimum of the energy.

Example 5. A non trivial covariant derivative ∇^{opt} compatible with the metric is constructed in [2] in the context of color image processing. Then, the gradient descent for reaching the sections minimizing the corresponding energy (9) is compared with the Beltrami flow in [20], that may be viewed as the gradient descent for reaching the sections minimizing the energy (9) associated with the trivial covariant derivative. Experiments show that colors and edges and better preserved with the non trivial covariant derivative. The authors explain this behaviour by the fact that the solutions, i.e. the sections φ satisfying $\nabla^{opt} \varphi = 0$, are not necessarily constant.

Heat Equation and Heat Kernel of Connection Laplacian. The gradient descent method for reaching sections minimizing the energy (9) corresponds to the heat equation of the connection Laplacian Δ^E

$$\frac{\partial \varphi}{\partial t} + \Delta^E \varphi = 0 \tag{10}$$

Results about heat equation and heat kernel of connection Laplacian are well-etablished (see e.g. [3]).

Given a connection Laplacian Δ^E and $\varphi_0 \in \Gamma(E)$, there exists a smooth map called **heat kernel** of Δ^E and denoted by K such that the operator $e^{-t\Delta^E}$ defined by

$$(e^{-t\Delta^E} \varphi_0)(x) = \int_M K_t(x, y) \varphi_0(y)\, dM$$

satisfies the heat equation (10).

The heat kernel of a connection Laplacian has a series expansion of the form

$$\left(\frac{1}{4\pi t}\right)^{\frac{n}{2}} e^{-d(x,y)^2/4t}\, \Psi(d(x,y)^2) \sum_{i=0}^{+\infty} t^i \Phi_i(x, y, \Delta^E)\, J(x, y)^{-\frac{1}{2}} \tag{11}$$

where $\Phi_i(x, y, \Delta^E) \in \mathrm{End}(E_y, E_x)$, n is the dimension of the base manifold M, and d stands for the geodesic distance on (M, g). The function Ψ is such that the term $\Psi(d(x, y)^2)$ equals 1 if y is inside a normal neighborhood of x and 0 otherwise. At last, J are the Jacobians of the coordinates changes from usual coordinates systems to normal coordinates systems.

The leading term of the series (11) is

$$\left(\frac{1}{4\pi t}\right)^{n/2} e^{-d(x,y)^2/4t}\, \Psi(d(x, y)^2)\, \tau(x, y)\, J(x, y)^{-1/2} \tag{12}$$

where $\tau(x, y)$ is the parallel transport map on E associated to ∇^E along the unique geodesic joining x and y.

Example 6. In [22], convolution with the leading term (12) associated to the Laplace-Beltrami operator was applied to anisotropic diffusion of color images. It was extended in [1] to connection Laplacians with no trivial connections.

Parallel Section and Harmonic Map. Harmonic maps between two Riemannian manifolds $\sigma\colon (M, g) \longrightarrow (N, Q)$ are defined as critical points of the **Dirichlet energy**

$$E(\sigma) = \int_M Tr(\sigma^* h)\, dM = \int_M \|d\sigma\|^2\, dM \tag{13}$$

The Euler-Lagrange equations of the functional (13) are

$$\tau(\sigma)\colon = Tr\, \nabla^{T^*M \otimes \sigma^{-1}(TN)} = 0$$

Note that the Dirichlet energy (13) is at the core of the Beltrami framework of Sochen et al. (see e.g. [20]).

Theorem 1 (Konderak [12]). *Let E be a vector bundle over a compact Riemannian manifold (M, g) equipped with a metric h and a covariant derivative ∇^E compatible with h. Let $\tilde{h}$ be the Sasaki metric on E associated to (h, ∇^E, g). Then $\sigma \in \Gamma(E)$ is a harmonic map $\sigma\colon (M, g) \longrightarrow (E, \tilde{h})$ if and only if it is parallel, i.e. $\nabla^E \sigma = 0$.*

Hence, the sections minimizing the energy (9) are harmonic maps with respect to the Sasaki metric on the vector bundle.

2.3 Total Variation on Vector Bundle

Definition 4. *Let E be a vector bundle over a compact Riemannian manifold (M, g) equipped with a Riemannian metric h, and a covariant derivative ∇^E compatible with h. We define the total variation TV of $\varphi \in \Gamma(E)$ as*

$$TV(\varphi) = \int_M ||\nabla^E \varphi||\, dM \tag{14}$$

Proposition 3. *The critical points of (14) are the sections φ satisfying*

$$-Tr\,\nabla^{T^*M\otimes E}\left(\frac{\nabla^E\varphi}{\|\nabla^E\varphi\|}\right)=0 \tag{15}$$

Proof. Let ψ be a section with compact support. The first variation of $TV(\varphi)$ in the direction ψ is

$$\delta\,TV(\varphi;\psi)=\int_M\left\langle\frac{\nabla^E\varphi}{\|\nabla^E\varphi\|},\nabla^E\psi\right\rangle dM=-\int_M\left\langle Tr\,\nabla^{T^M\otimes E}\left(\frac{\nabla^E\varphi}{\|\nabla^E\varphi\|}\right),\psi\right\rangle dM$$

since $-Tr\,\nabla^{T^M\otimes E}$ is the adjoint of ∇^E. As ψ has compact support, it follows

$$\delta\,TV(\varphi;\psi)=0\Longrightarrow -Tr\,\nabla^{T^*M\otimes E}\left(\frac{\nabla^E\varphi}{\|\nabla^E\varphi\|}\right)=0 \qquad \square$$

Remark 1. Formula (15) is not defined where $\nabla^E\varphi$ vanishes. One way to tackle this problem is to consider a regularized Total Variation

$$TV^{reg}(\varphi)=\int_M\sqrt{||\nabla^E\varphi||^2+\beta}\;dM,\qquad \beta>0 \tag{16}$$

Example 7. Let E be a vector bundle of rank m equipped with the trivial connection and Euclidean scalar product over a compact Euclidean manifold of dimension n. Then, for $\varphi\in\Gamma(E)$, we have

$$TV(\varphi)=\int_M\sqrt{\sum_{i=1}^n\sum_{j=1}^m\left(\frac{\partial\varphi^j}{\partial x_i}\right)^2}\,dM \tag{17}$$

Formula (17) corresponds to the total variation defined by Blomgren et al. [4]. In particular, for $n=2$ and $m=1$, this is the total variation of Rudin et al. [19]. Hence, these two approaches may be viewed as Euclidean restrictions of (14).

3 Application to Color Image Denoising

3.1 ROF Denoising Model on Vector Bundle: The General Case

Continuous Setting. Let E be a vector bundle of rank 3 equipped with a definite positive metric h and covariant derivative ∇^E compatible with h over a Riemannian manifold (M,g) of dimension 2. Let $I_0\in BV(E)$ be a color image corrupted by additive Gaussian noise of deviation σ. We propose the denoising model

$$\underset{I\in BV(E)}{\arg\min}\int_M\frac{1}{2}\lambda\|I-I_0\|^2d\Omega+\|\nabla^E I\|\,dM \tag{18}$$

where $d\Omega$ denotes the Euclidean measure on M and λ is a Lagrange multiplier associated with the noise level.

The gradient descent for reaching solutions of (18) is

$$\frac{\partial I}{\partial t}=-\lambda(I-I_0)+Tr\,\nabla^{T^*M\otimes E}\left(\frac{\nabla^E I}{\|\nabla^E I\|}\right),\quad I_{|t=0}=I_0 \tag{19}$$

Discrete Setting. We follow the approach in [19] where forward and backward finite difference operators are used for discretizing the trivial covariant derivative d and its adjoint $-div$. The key idea is to use the discrete version of the adjoint operator definition, which is written as follows in the context of connection gradient

$$\int_M -div\,\eta\,\varphi\,d\Omega = \int_M \langle \eta,\, d\varphi \rangle\, d\Omega$$

for $\varphi \in C^\infty(M)$ and $\eta \in \Gamma(T^*M)$. Then, using forward differences for discretizing $d\varphi$ implies that $div\,\eta$ must be discretized using backward differences.

We extend this approach by using the general definition of adjoint operator (7). Let us give the explicit expressions in the case of base manifold of dimension 2 and vector bundle of rank 3. Let $\varphi = \sum_{j=1}^{3} \varphi^j e_j \in \Gamma(E)$ where (e_1, e_2, e_3) is orthonormal with respect to h. Under the forward finite difference operators for approximating $d\varphi^j$, we have

$$\nabla^E \varphi_{\,m,n} = \sum_{i=1}^{2}\sum_{j=1}^{3}\left[(\varphi^j_{m+\delta_{i1},n+\delta_{i2}} - \varphi^j_{m,n}) + \sum_{k=1}^{3} \varphi^k_{m,n}\,\Upsilon^j_{ik\;m,n}\right] dx_i \otimes e_j \quad (20)$$

where δ is the Kronecker symbol. Then, using the discrete form of (7) given by

$$\sum_{m,n} h_{m,n}(\nabla^{E^*}\eta_{\,m,n}, \varphi_{m,n}) = \sum_{m,n} \langle \eta_{\,m,n}, \nabla^E \varphi_{\,m,n}\rangle_{m,n}$$

we obtain, for $\eta = \sum_{i=1}^{2}\sum_{j=1}^{3} \eta^{ij} dx_i \otimes e_j$, the expression $\nabla^{E^*}\eta_{\,m,n} =$

$$\sum_{j=1}^{3}\left[\sum_{i,k=1}^{2}(g^{ik}_{m-\delta_{k1},n-\delta_{k2}}\eta^{ij}_{m-\delta_{k1},n-\delta_{k2}} - g^{ik}_{m,n}\eta^{ij}_{m,n}) + \sum_{r,s=1}^{2}\sum_{p=1}^{3}\eta^{rp}_{m,n}\,g^{rs}_{m,n}\,\Upsilon^p_{sj\;m,n}\right] e_j \quad (21)$$

Hence, as in the Euclidean case [19], forward finite difference operators on ∇^E imply backward finite difference operators on ∇^{E^*}.

3.2 ROF Denoising Model on Vector Bundle: An Example

Connection Gradient Suitable for Color Image Processing. Let $I_0 = (I_0^1, I_0^2, I_0^3)$ be a color image defined on a domain Ω of $\mathbb{R}^2$.

We construct a surface S embedded in $(\mathbb{R}^5, \|\,\|_2)$ parametrized by

$$\varphi\colon (x_1, x_2) \longmapsto (x_1, x_2, \mu\, I_0^1(x_1, x_2), \mu\, I_0^2(x_1, x_2), \mu\, I_0^3(x_1, x_2)), \qquad \mu > 0$$

in the fixed orthonormal frame $(e_1, e_2, e_3, e_4, e_5)$ of $(\mathbb{R}^5, \|\,\|_2)$.

Let E be the vector bundle of $\mathbb{R}^5$-valued functions over the Euclidean manifold $(\Omega, \|\,\|_2)$. Let $(Z_1, Z_2, N_1, N_2, N_3)$ be an orthonormal frame field of E with

respect to the Euclidean norm, where $Z_1, Z_2 \in \Gamma(TS)$. Let ∇^E be the covariant derivative on E given by the connection 1-form $\omega \equiv 0$ in the frame $(Z_1, Z_2, N_1, N_2, N_3)$. Denoting by P the change frame field from $(e_1, e_2, e_3, e_4, e_5)$ to $(Z_1, Z_2, N_1, N_2, N_3)$, ω is given in the frame $(e_1, e_2, e_3, e_4, e_5)$ by

$$P\, dP^{-1} \tag{22}$$

As the connection 1-form ω is $\mathfrak{so}(\mathfrak{n})$-valued, the covariant derivative ∇^E is compatible with the Euclidean metric on $\mathbb{R}^5$.

Remark 2. The orthonormal frame field $(Z_1, Z_2, N_1, N_2, N_3)$ of E varies at each point of Ω since the vector fields Z_1 and Z_2 are required to be tangent vector fields of the surface S. Moreover, it is not unique since the vector fields Z_1 and Z_2 are defined up to rotations in the tangent planes of S. By construction, the frame $(Z_1, Z_2, N_1, N_2, N_3)$ takes into account the local variations of I_0.

Algorithm. We test the denoising model (18) with this connection gradient.

The algorithm is the following:

1. Consider the (discrete) surface S parametrized by

$$\varphi\colon (m,n) \longmapsto (m, n, \mu\, I_0^1(m,n), \mu\, I_0^2(m,n), \mu\, I_0^3(m,n)), \qquad \mu > 0$$

2. Construct an orthonormal moving frame $(Z_1, Z_2, N_1, N_2, N_3)$ using Gram-Schmidt process at each point (m,n) with the assumption that $Z_1, Z_2\,(m,n) \in T_{m,n}S$, and denote by $P(m,n)$ the basis change from $(e_1, e_2, e_3, e_4, e_5)$ to $(Z_1, Z_2, N_1, N_2, N_3)(m,n)$.
3. Embed I_0 into the frame $(e_1, e_2, e_3, e_4, e_5)$: $(I_0^1, I_0^2, I_0^3) \longrightarrow (0, 0, I_0^1, I_0^2, I_0^3)$.
4. Compute the components of I_0 in the frame $(Z_1, Z_2, N_1, N_2, N_3)$: $(J_0^1, J_0^2, J_0^3, J_0^4, J_0^5)^T := P^{-1}(0, 0, I_0^1, I_0^2, I_0^3)^T$.
5. Perform the Euclidean ROF denoising algorithm on $(J_0^1, J_0^2, J_0^3, J_0^4, J_0^5)^T$ with stopping criteria

$$\frac{1}{|\Omega| \times 3} \sum_{x \in \Omega} \|J_t(x) - J_0(x)\|^2 \geq \sigma^2$$

or

$$\left| \frac{1}{|\Omega| \times 3} \sum_{x \in \Omega} \|J_{t+dt}(x) - J_0(x)\|^2 - \frac{1}{|\Omega| \times 3} \sum_{x \in \Omega} \|J_t(x) - J_0(x)\|^2 \right| \leq 0.0005$$

whichever happens first.
6. Compute the components of the result in the frame $(e_1, e_2, e_3, e_4, e_5)$: $(I_t^1, I_t^2, I_t^3, I_t^4, I_t^5)^T := P(J_t^1, J_t^2, J_t^3, J_t^4, J_t^5)^T$, and return the function (I_t^3, I_t^4, I_t^5).

Experiments. We run the algorithm on the Kodak database [11], for $\sigma = 5, 10, 15, 20, 25$. We take $\mu = 0.0075$ for $\sigma = 5$, $\mu = 0.005$ for $\sigma = 10$, $\mu = 0.0045$ for $\sigma = 15$, $\mu = 0.004$ for $\sigma = 20$ and $\mu = 0.0035$ for $\sigma = 25$.

The time step dt is 0.1. On Fig. 1, we test our denoising method and show that it removes the noise efficiently. On Fig. 2, we compare our denoising method with the split Bregman denoising method [9] tested online [7]. We observe that our method preserves more the details of the image. We claim that it comes from the choice of the moving frame $(Z_1, Z_2, N_1, N_2, N_3)$, that takes into account the local variations of the image. On Fig. 3, we compute the average increases of PSNR as well as the average percent increases of Q-index [23] over the Kodak database for both methods (we define the Q-index of a color image as the mean of the Q-index on each component). Results show that our method improves the split Bregman method.

Fig. 1. Example of our denoising method. Left: image corrupted by additive white noise with $\sigma = 25$. right: denoised image.

Fig. 2. Comparison of our denoising method with split Bregman: From left to right: original image, image corrupted by additive white noise with $\sigma = 25$, split Bregman denoising result, our denoising result.

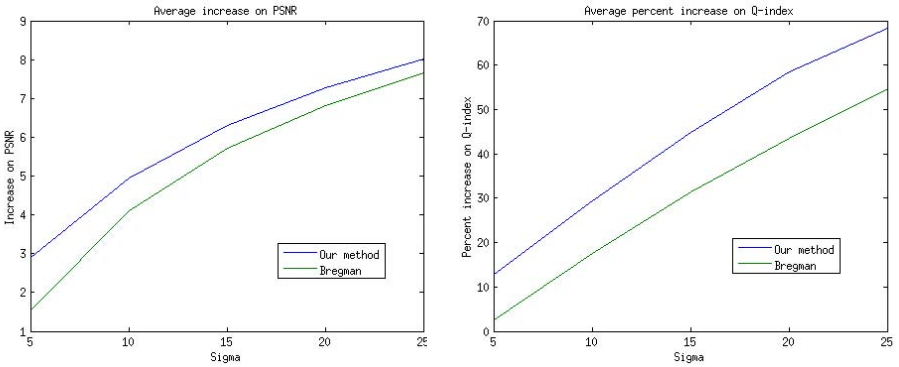


Fig. 3. Comparison of our method with split Bregman. Left: PSNR increase for each method. Right: Percent increase on Q-index for each method. Values averaged over Kodak database.

4 Conclusion

In this paper, we introduced a generalization of the Euclidean and Riemannian gradient operators in the context of vector bundles. We presented an application to image denoising by replacing the Euclidean gradient in the regularizing term of the Rudin-Osher-Fatemi denoising model by a generalized gradient on a vector bundle. By the gradient operator we considered, the denoising method is decomposed into 2 steps: first, a projection of the image on the tangent and normal parts of a surface describing the image; then, an Euclidean ROF denoising method of the image projection in this moving frame. The relevance of the method is justified by the PSNR and Q-index measures. In some sense, the denoising method preserves the first order local geometry of the image.

We would like to point out that the step 2 of our denoising method might be extended to any denoising method. In particular, we expect that nonlocal denoising methods like Non Local Means [5] and BM3D [6] would increase significantly both PSNR and Q-index. More generally, inspired by [8] where the Euclidean ROF model [19] is extended to a nonlocal model by the construction of a nonlocal gradient operator, we expect that our vector bundle ROF model extends to a nonlocal model by the construction of a nonlocal connection gradient operator.

References

1. Batard, T.: Heat Equations on Vector Bundles - Application to Color Image Regularization. J. Math. Imaging Vis. 41(1-2), 59–85 (2011)
2. Batard, T., Sochen, N.: A Class of Generalized Laplacians Devoted to Multi-Channel Image Processing. To Appear in J. Math. Imaging Vis., doi 10.1007/s10851-013-0426-7
3. Berline, N., Getzler, E., Vergne, M.: Heat Kernels and Dirac Operators. Springer (2004)

4. Blomgren, P., Chan, T.F.: Color TV: Total Variation Methods for Restoration of Vector-Valued Images. IEEE Trans. Image Processing 7(3), 304–309 (1998)
5. Buades, A., Coll, B., Morel, J.-M.: A Review of Image Denoising Algorithms, with a new one. Multiscale Model. Simul. 4(2), 490–530 (2005)
6. Dabov, K., Foi, V., Katkovnik, V., Egiazarian, K.: Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. IEEE Trans. Image Processing 16(8), 2080–2095 (2007)
7. Getreuer, P.: Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman. Image Processing On Line (2012)
8. Gilboa, G., Osher, S.: Nonlocal Operators with Applications to Image Processing. Multiscale Model. Simul. 7(3), 1005–1028 (2008)
9. Goldstein, T., Osher, S.: The Split Bregman Method for L1 Regularized Problems. SIAM J. Imaging Sciences 2, 323–343 (2009)
10. Jin, Y., Jost, J., Wang, G.: A New Nonlocal H^1 Model for Image Denoising. To Appear in J. Math. Imaging Vis., doi 10.1007/s10851-012-0395-2
11. Kodak, `http://r0k.us/graphics/kodak/`
12. Konderak, J.J.: On Sections of Fiber Bundles which are Harmonic Maps. Bull. Math. Soc. Sci. Math. Roumanie 90(4), 341–352 (1999)
13. Lawson, H.B., Michelson, M.-L.: Spin Geometry. Princeton University Press (1989)
14. Lysaker, M., Osher, S., Tai, X.-C.: Noise Removal using Smoothed Normals and Surface Fitting. IEEE Trans. Image Processing 13(10), 1345–1357 (2004)
15. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An Iterative Regularization Method for Total Variation-based Image Restoration. Multiscale Model. Simul. 4(2), 460–489 (2004)
16. Rahman, T., Tai, X.-C., Osher, S.J.: A TV-Stokes Denoising Algorithm. In: Sgallari, F., Murli, A., Paragios, N. (eds.) SSVM 2007. LNCS, vol. 4485, pp. 473–483. Springer, Heidelberg (2007)
17. Rosman, G., Tai, X.-C., Dascal, L., Kimmel, R.: Polyakov Action Minimization for Efficient Color Image Processing. In: Kutulakos, K.N. (ed.) ECCV 2010 Workshops, Part II. LNCS, vol. 6554, pp. 50–61. Springer, Heidelberg (2012)
18. Rosman, G., Wang, Y., Tai, X.-C., Kimmel, R., Bruckstein, A.M.: Fast Regularization of Matrix-Valued Images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 173–186. Springer, Heidelberg (2012)
19. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear Total Variation Based Noise Removal Algorithms. Physica D 60, 259–268 (1992)
20. Sochen, N., Kimmel, R., Malladi, R.: A General Framework for Low Level Vision. IEEE Trans. Image Processing 7(3), 310–318 (1998)
21. Spivak, M.: A Comprehensive Introduction to Differential Geometry. Publish or Perish, 2nd edn. (1990)
22. Spira, A., Kimmel, R., Sochen, N.: A Short-time Beltrami Kernel for Smoothing Images and Manifolds. IEEE Trans. Image Processing 16, 1628–1636 (2007)
23. Wang, Z., Bovik, A.: A Universal Image Quality Index. IEEE Signal Processing Letters 9(3), 81–84 (2002)
24. 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)
25. Zhu, W., Chan, T.F.: Image Denoising using Mean Curvature of Image Surface. SIAM J. Imaging Sciences 5(1), 1–32 (2012)

Expert Regularizers for Task Specific Processing

Guy Gilboa

Department of Electrical Engineering, Technion,
Israel Institute of Technology,
Haifa 32000, Israel

Abstract. This study is concerned with constructing expert regularizers for specific tasks. We discuss the general problem of what is desired from a regularizer, when one knows the type of images to be processed. The aim is to improve the processing quality and to reduce artifacts created by standard, general-purpose, regularizers, such as total-variation or nonlocal functionals.

Fundamental requirements for the theoretic expert regularizer are formulated. A simplistic regularizer is then presented, which approximates in some sense the ideal requirements.

1 Introduction

In many cases one knows the type of images and objects which are to be processed. Therefore it makes sense to incorporate this knowledge when solving an image processing problem. In the variational approach this means selecting the proper regularizer which best fits the signal. Thus avoiding, as much as possible, artifacts caused by the regularizer, such as over-smoothing, reduction in contrast, removal of small details, corners, lines or textural patterns. We will investigate the general problem of how such regularizers can be constructed, propose some desired properties and suggest a simple procedure to construct an expert regularizer (ER) for specific type of signals based on region similarity and a collection of typical images. We first summarize briefly the main regularizers which are currently being used.

1.1 A Brief Overview on Regularizers in Image Processing

Regularizers have been introduced to increase the robustness of a solution in problems which are often ill posed and are therefore significantly affected by noise, such as deconvolution, optical flow [7] or image-registration [14].

Very roughly, regularizers can be classified into three large categories: (i) General model-based, (ii) Specific model-based, (iii) Self-similarity. The first two can be considered parametric methods, whereas the third is a non-parametric (data-driven) one.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 24–35, 2013.

General Model-Based. This category is for general images. Images are approximated by a certain mathematical model. The most notable regularizer in this class is the total-variation (TV) functional [28]

$$J_{TV}(u) = \int_{\Omega} |\nabla u(x)| dx, \tag{1}$$

which is excellent for piece-wise constant images and copes well also with piece-wise smooth images (introduces staircasing). This simplified model of images retains edges well, but cannot distinguish well enough between textures and noise. Other artifact associated with total-variation are some reduction in contrast and oversmoothing of corners. Many other general regularizers were proposed in the literature. Higher order derivatives were used to reduce staircasing and improve the response in gradual luminance changes [6,11,18].

To avoid contrast reduction and erosion of fine details several approaches were taken: An iterative procedure of TV regularizations was suggested, understood as Bregman iterations [23]. In the limit of this procedure (as the regularization parameter approaches zero) a continuous formulation is obtained, [10], keeping better contrast and detail. A different approach retained the standard TV functional and changed the fidelity term from L^2 to norms which better favor oscillatory patterns, such as G or H^{-1} norms [22,3,25].

Nonlinear diffusions and many related PDE's can be viewed as steepest descent of gradient-based regularizers (see more in [30,2]), which belong to the general piece-wise smooth image model.

Specific Model-Based. Some regularizers were designed for specific tasks, using an underlying mathematical model. Special regularizers were proposed for preferred convex shapes [24], rectangles [5], lines and vessels-adapted regularization [4,19] and more.

Self-similarity. This category was introduced more recently, following the non local means denoising approach [9]. Here the method is non-parametric, where the model stems from the image data itself. It is based on a generic characteristic of self-similarity: images tend to exhibit regions which are similar to one another. Self-similarity regularizers were recently proposed, e.g. [20,16,17,21,1]. The simplest generic self-similarity regularizer, termed *nonlocal* H^1 [17,21], is:

$$J_{NL-H1}(u) := \int_{\Omega}\int_{\Omega} (u(x) - u(y))^2 w(x,y) dy dx. \tag{2}$$

1.2 The New Proposed Approach

The approach that will be presented here is task-specific and non-parametric. That is, the smoothing properties are data-driven and not dictated in an *a priori* manner. In Section 2 the ideal model is described, where several general desired requirements for such regularizers are stated. In Section 3 a regularizer, which

is essentially based on similarity to external data, is proposed. It is shown in what aspects the proposed functional could be considered an expert regularizer. In Section 4 some examples are shown.

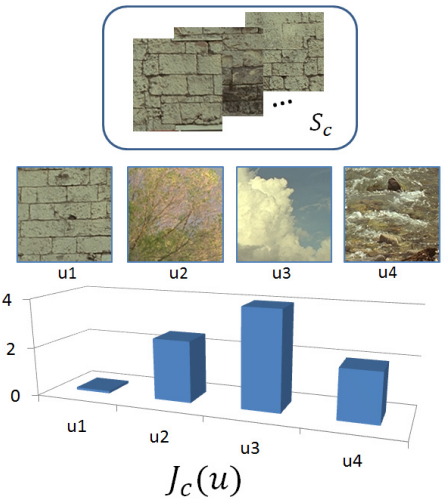


Fig. 1. Illustration of an ideal regularizer response

2 Ideal Expert Regularizers

2.1 Definitions and Characteristics

We first define the generic requirements desired from an expert regularizer.

Definition 1 (Categorial Set). *Let a categorial set S_c be the set of all possible images belonging to a certain category $\mathcal{C}$.*

Category $\mathcal{C}$ here can be quite particular, like "Vehicles from 40-45 meters, as seen by a surveillance camera" or "Computed Tomography (CT) images of liver tissues" to more general concepts such as Clouds, Grass or Sand.

Definition 2 (Expert Regularizer). *Let an expert regularizer J_c be a functional which is designed to regularize images belonging to category $\mathcal{C}$ with minimal degradations and regularization artifacts.*

This is a very general definition, which leaves some degrees of freedom. We advocate the following properties:

Desired Properties. For an expert regularizer J_c the following properties are desired:

(i) $J_c(u) \geq 0$, $J_c(u) = 0$ iff $u \in S_c$.
(ii) If $\exists u_0 \in S_c, \|u_0 - u\|_{L^2} \leq \delta$ then $J_c(u) \leq \epsilon$.
(iii) If $\forall u_0 \in S_c, \|u_0 - u\|_{L^2} \geq A$, then $J_c(u) \geq kA$,

where δ, ϵ, k and A are all positive. We would like δ to be as small as possible, and typically ϵ would be a function of δ, $\epsilon(\delta)$. These parameters characterize J_c and S_c and are independent of any specific images u, u_0. Naturally, various other variations can be suggested, such as using norms other than L^2.

Motivation of Properties: Property (i) states that J_c is a positive functional (energy) which attains its global minimum (zero) if and only if the image belongs to the category. Thus, the attractors are images in the category and not the constant function of traditional regularizers. In other words, we have a *nontrivial null-space*, whereas total variation (and other convex gradient-based functionals) attain zero only for constants. This property should reduce artifacts associated with classical regularizers, which are caused by over-simplifying the image model. Properties (ii) and (iii) state that images close to the category will have a low energy and images far from any instance in the category will attain a high energy.

Note that property (i) yields that for an input image $f \in S_c$ an optimization of the form: $E = \min_u \{J_c(u) + \lambda \|f - u\|^2\}$ will map f to itself ($u = f$, "artifact free") attaining $E = 0$, for any value of λ. See illustration in Fig. 2. An earlier mention of non-trivial steady-states and their usefulness to preserve textures appears in [8], but it is not in the ER context, where no external data is used (therefore its power to preserve typical category properties is more limited).

3 Approximate Regularizers

Obtaining expert regularizers is not a trivial task and there may be various ways to tackle this problem. A MRF approach to learn image priors can be seen, for instance, in [26]. We present here a simple formulation that can give a good approximation to the desired properties defined above. Let a category C be approximated by a subset $\hat{S}_c \subset S_c$ of images.

The main idea is to separate the task into two: given the input image to be regularized, select affinities from the external data set $\hat{S}_c$ which are most relevant. Then use a convex functional for the regularization. See [12] for using patch-based similarity measures of external data to improve segmentation.

Let $g_c(y)$ be a function over a region Ω_c representing category $\mathcal{C}$ (practically it consists of many images belonging to the category. To simplify the formulation we can think of it as a single very large region, where all the images are concatenated to each other, and the boundary pixels are not considered). Let $w(x, y)$ be a non-negative function ($w(x, y) \geq 0$) which defines the similarity between point x (in Ω) to point y (in Ω_c). We examine the following functional:

$$J(u) := \frac{1}{|\Omega|} \int_{\Omega} \int_{\Omega_c} (u(x) - g_c(y))^2 w(x, y) dy dx, \tag{3}$$

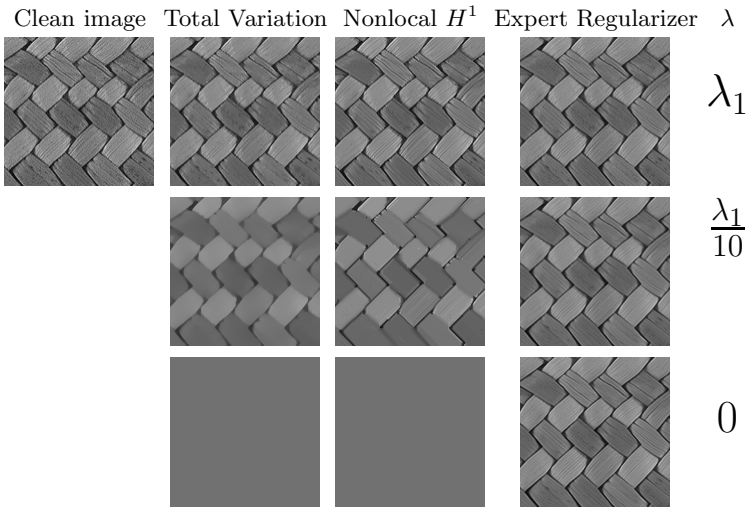


Fig. 2. Regularization of a clean image by various methods. Top left - clean image. Regularization results for different fidelity weight λ using 3 regularizers: TV, Eq. (1), Nonlocal H^1, Eq. (2), and an approximated ER proposed here. To reduce artifacts, the attractor in ER is an admissible instance in the category and not the constant function, as in traditional convex regularization.

where $|\Omega|$ denotes the area of Ω (as the value of $J(u)$ can be of interest to us, especially when comparing different images, we want to normalize the image size). The Euler-Lagrange is

$$J'(u) = \frac{2}{|\Omega|} \int_{\Omega_c} (u(x) - g_c(y))w(x,y)dy. \tag{4}$$

For a given input image f, we can use the standard L^2 fidelity term:

$$J_{fid}(u,f) = \frac{1}{|\Omega|} \int_{\Omega} (u(x) - f(x))^2 dx. \tag{5}$$

A total energy for denoising can be defined by:

$$E(u,f) = J(u) + \lambda J_{fid}(u,f). \tag{6}$$

The solution in this case is very simple. Having fixed $w(x,y)$, this energy is convex and the local minimum coincides with the global one. The Euler-Lagrange condition is:

$$J'(u) + \frac{2\lambda}{|\Omega|}(u - f) = 0.$$

Using (4) we reach the solution:

$$u(x) = \frac{\int_{\Omega_c} g_c(y)w(x,y)dy + \lambda f(x)}{\int_{\Omega_c} w(x,y)dy + \lambda}. \tag{7}$$

3.1 A Simple ER Approximation

One can construct a simple approximation of an expert regularizer using (3) provided the weights $w(x, y)$ are constructed properly.

Constructing the Weights. Let W be a neighborhood of radius r around the origin, $\tilde{x} \in W$ if $|\tilde{x}| \leq r$, with area $|W|$ (in the discrete case a square is often used, or "L^∞ radius"). We define the following square distance measure:

$$d^2_W(f(x), g(y); x, y) := \frac{1}{|W|} \int_W (f(x + \tilde{x}) - g(y + \tilde{x}))^2 d\tilde{x}. \tag{8}$$

We denote for short

$$d^2_W(x, y) := d^2_W(u, g_c; x, y) = \frac{1}{|W|} \int_W (u(x + \tilde{x}) - g_c(y + \tilde{x}))^2 d\tilde{x}.$$

For every point x, we search for a domain $\mathit{\Pi}(x; y) \subset \Omega_c$ of fixed size $\mathcal{K}$ with the minimal distance. Let

$$\mathit{\Pi}(x) = \mathit{\Pi}(x; y) := \{y \,|\, d^2_W(x, y) \leq t\},$$

where t is the minimal value obtaining $|\mathit{\Pi}(x)| = \mathcal{K}$. The weights $w(x, y)$ are computed by

$$w(x, y) = \begin{cases} \frac{1}{\mathcal{K}} e^{-d^2_W(x,y)/h^2}, & y \in \mathit{\Pi}(x) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where h is a parameter characterizing the distance relevancy; in denoising it can be assigned the value $h = \sigma$.

In the discrete case this translates to a K-nearest neighbors search, based on the distance $d_W(x, y)$ where the area size $\mathcal{K}$ is proportional to the number of discrete neighbors K.

3.2 Which Expert Regularizer Properties Hold?

The regularizer (3) can be considered a rough approximation of the ideal expert-regularizer. We show below which ER properties (stated in Section 2.1) can be satisfied, under some conditions, and which cannot.

We first define a regularity condition of the categorial set.

Definition 3 (Regular Set). *A set S_c is regular if for some $\mathcal{K} > 0$, $\epsilon > 0$ and $d^2_W(g_c(y), g_c(\tilde{y}); y, \tilde{y})$, there exists $\mathit{\Pi}(y)$, $|\mathit{\Pi}(y)| \geq \mathcal{K}$, such that $\forall y \in \Omega_c$,*

$$\frac{1}{\mathcal{K}} \int_{\mathit{\Pi}(y)} (g_c(y) - g_c(\tilde{y}))^2 d\tilde{y} \leq \epsilon.$$

This essentially means that there are no large outliers (very rare instances which do not repeat). In other words, for all types of signals in the set there are at least a few similar instances, where "few" is defined by $\mathcal{K}$ and "similar" is defined by ϵ.

Proposition 1. *If $u \in \hat{S}_c$ and $\hat{S}_c$ is a regular set, then $J(u) \leq \epsilon$.*

Proof. From the definition of the weight, Eq. (9), we have $w(x,y) \leq \frac{1}{\mathcal{K}}$. Moreover, since $w(x,y) = 0$ for all $y \notin \Pi(x)$, (see Eq. (9)), the second integration domain in Eq. (3), Ω_c, can be replaced by $\Pi(x)$, yielding

$$J(u) \leq \frac{1}{|\Omega|\mathcal{K}} \int_{\Omega} \int_{\Pi(x)} (u(x) - g_c(y))^2 dy dx. \qquad (*)$$

As $u \in \hat{S}_c$ we can use the regularity condition of $\hat{S}_c$ to obtain $\int_{\Pi(x)}(u(x) - g_c(y))^2 dy \leq \mathcal{K}\epsilon$. □

Here we examine images which are close to the category. We denote $\Pi|_u$ as $\Pi(x)$ which is based on u. Here we require the following condition: (C) $\int_{\Pi|_u}(u(x) - g_c(y))^2 dy \leq \int_{\Pi|_{u_0}}(u(x) - g_c(y))^2 dy$. In the case of infinitesimal patch size this would naturally follow, since one takes the closest points in the definition of Π.

Proposition 2. *If $u_0 \in \hat{S}_c$, $\|u_0 - u\|_{L^2} \leq \delta$, $\hat{S}_c$ is a regular set and condition (C) holds, then $J(u) \leq \epsilon_1$.*

Proof. We use the above upper bound on $J(u)$, $(*)$, and show a bound on the inner integral. We expand $(u(x) - g_c(y))^2$ to $((u(x) - u_0(x)) + (u_0(x) - g_c(y)))^2$. Then we use the bounds $\int_{\Pi|_{u_0}}(u(x) - u_0(x))^2 dy \leq \int_{\Pi|_{u_0}} dy \|u_0 - u\|^2_{L^2} \leq \mathcal{K}\delta^2$, $\int_{\Pi|_{u_0}}(u_0(x) - g_c(y))^2 dy \leq \mathcal{K}\epsilon$ (regularity condition), condition (C) and the Cauchy-Schwarz inequality to obtain $J(u) \leq \epsilon_1 = (\delta + \sqrt{\epsilon})^2$. □

A variation of Property (iii) in Section 2.1 cannot be proved with this regularizer. Although, in most cases, images which are significantly different from the category will attain a high regularizer value, it is possible to find specific contradicting cases, as illustrated in Fig. 3 and formalized in the following:

Proposition 3. *$\exists u$, $\hat{S}_c$ such that $\forall u_0 \in \hat{S}_c$, $\|u_0 - u\|_{L^2} \geq A$, $J(u) = 0$.*

Proof. Examine the following simple example: $\hat{S}_c$ consists of a single image u_0 of area $|\Omega| = 1$ with half plain $u_0 = 0$ and half $u_0 = 2A$ (see Fig. 3). The zero image $u = 0$ satisfies $\|u_0 - u\|_{L^2} = \sqrt{2}A > A$. For $|W| + \mathcal{K} < 0.5$ we can find $\Pi(x) \subset \Omega$ for which $u_0|_{\Pi(x)} = 0$, hence $\int_{\Pi(x)}(u(x) - g_c(y))^2 dy = 0$, $\forall x$ and $J(u) = 0$. □

3.3 Point Space versus Region Space

In the standard regularization functionals (such as (1), (2) and (3)) one works in point space, the Euler-Lagrange is done point-wise and the result $u(x)$ is the regularized image. We would like to examine a region-based regularization. In the discrete setting this is understood as pixel versus patch processing. See related studies addressing this issue, e.g. in [13,15,29]. In a region approach a small local region around each point is regularized and the image is then reconstructed. Our

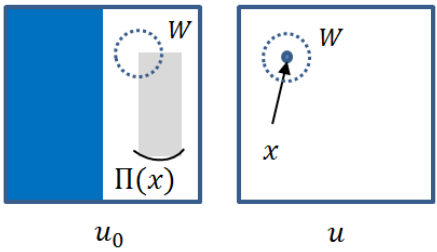


Fig. 3. Illustration of Proposition 3

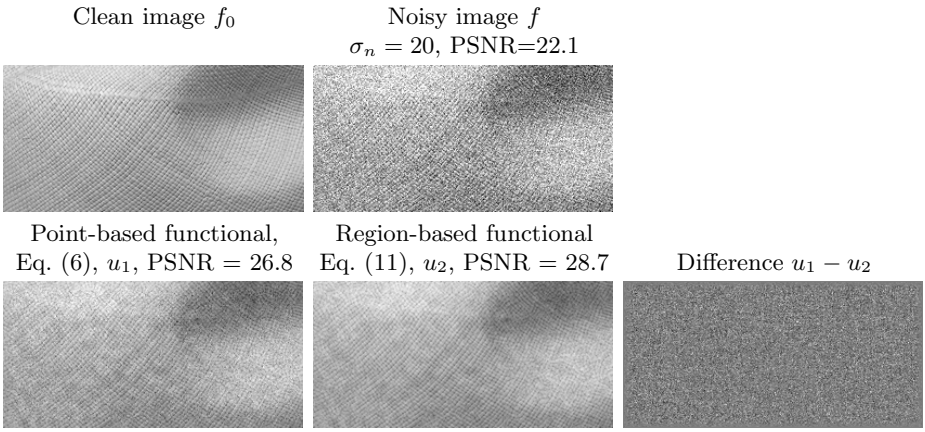


Fig. 4. Point vs. region functionals. Region functionals retain more coherent structural results.

experiments, shown below, indicate that more coherent results are obtained, which better fit the category, with improved SNR. Note that this is performed in addition to the region-based similarity of $w(x, y)$.

Using the window W defined in Section 3.1, we map an image $f(x)$, $x \in \Omega$ to $F(x, \tilde{x})$, $(x, \tilde{x}) \in (\Omega \times W)$ by $F(x, \tilde{x}) := f(x + \tilde{x})$, $x \in \Omega$, $\tilde{x} \in W$. We thus define $G(y, \tilde{x})$, $(y, \tilde{x}) \in (\Omega_c \times W)$ by $G(y, \tilde{x}) := g(y + \tilde{x})$, $y \in \Omega_c$, $\tilde{x} \in W$. Appropriate boundary conditions (such as mirror) should be set to define the values at the boundary of Ω (or Ω_c). An associated weight can be added $\tilde{w}(\tilde{x}) \geq 0$, $\int_W \tilde{w}(\tilde{x})d\tilde{x} = 1$ with higher values near the origin. We can now define the following regularizer:

$$J_W(U) := \frac{1}{|\Omega|} \int_\Omega \int_{\Omega_c} \int_W (U(x, \tilde{x}) - G_c(y, \tilde{x}))^2 \tilde{w}(\tilde{x}) w(x, y) d\tilde{x} dy dx, \qquad (10)$$

where $U(x, \tilde{x})$, $(x, \tilde{x}) \in (\Omega \times W)$ is the regularized function of regions. The associated (region-based) Euler-Lagrange is: $J'_W(U)(x, \tilde{x}) = \frac{2\tilde{w}(\tilde{x})}{|\Omega|} \int_{\Omega_c} (U(x, \tilde{x}) -$

$G_c(y,\tilde{x}))w(x,y)dy$. Using a region-equivalent fidelity term $J_{fid,W}(U,F) = \frac{1}{|\Omega|}\int_\Omega \int_W (U(x,\tilde{x}) - F(x,\tilde{x}))^2 \tilde{w}(\tilde{x})d\tilde{x}dx$, and energy

$$E_W(U,F) = J_W(U) + \lambda J_{fid,W}(U,F), \tag{11}$$

the solution for the minimizer is:

$$U(x,\tilde{x}) = \frac{\int_{\Omega_c} G_c(y,\tilde{x})w(x,y)dy + \lambda F(x,\tilde{x})}{\int_{\Omega_c} w(x,y)dy + \lambda}. \tag{12}$$

Having solved for $U(x,\tilde{x})$ we would like to reconstruct $u(x)$. For every point x we now have an entire set representing this point: $\left\{\hat{x}\,|\,x = \hat{x} + \tilde{x},\ \tilde{x} \in W\right\}$. The simplest way to reconstruct $u(x)$ is by weighted integration:
$u(x) = \int_{x=\hat{x}+\tilde{x}} U(\hat{x},\tilde{x})\tilde{w}(\tilde{x})d\tilde{x}$, $\tilde{x} \in W$. By replacing $\hat{x} = x - \tilde{x}$ we get,

$$u(x) = \int_W U(x-\tilde{x},\tilde{x})\tilde{w}(\tilde{x})d\tilde{x}. \tag{13}$$

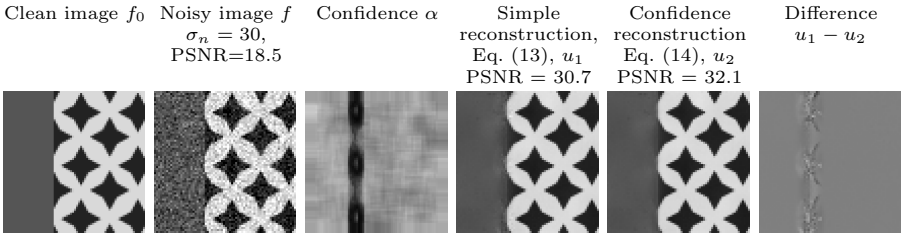


Fig. 5. Standard reconstruction vs. confidence-based reconstruction. When confidence is considered, boundaries between classes are regularized better.

Confidence-Based Integration. A more advanced way to reconstruct $u(x)$ from $U(x,\tilde{x})$ is to consider the local quality of the regularization. Some regions may be more effectively regularized than others. A confidence function $0 \leq \alpha(x) \leq 1$ is associated with $U(x,\tilde{x})$. Then $u(x)$ can be reconstructed by

$$u(x) = \int_W U(x-\tilde{x},\tilde{x})\frac{\alpha(x-\tilde{x})}{\int_W \alpha(x-\check{x})d\check{x}}d\tilde{x}. \tag{14}$$

We use the following confidence measure: $\alpha(x) = \int_{\Pi(x)} w(x,y)dy$. The superior effect of this approach can be seen in Fig. 5. As category boundaries are reached, confidence is low, and information on the most adequate regularization is gathered from regions within the window W that are farther from the boundary.

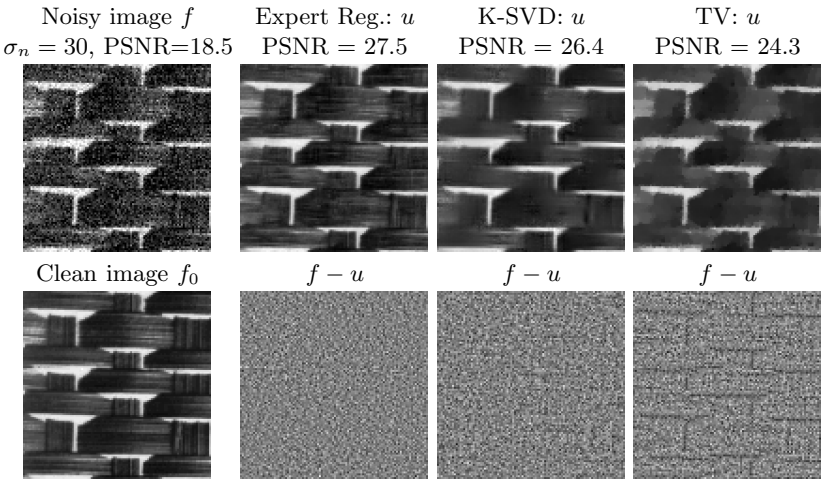


Fig. 6. Denoising examples 1: Expert regularizer, K-SVD and TV

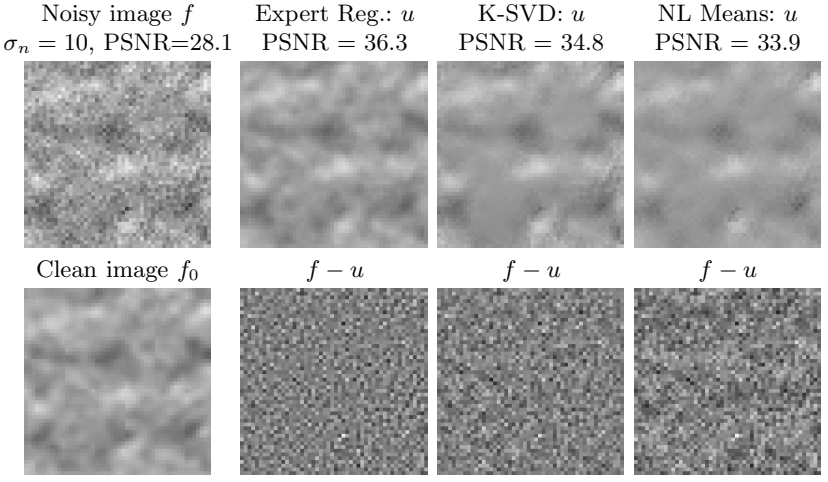


Fig. 7. Denoising examples 2: Expert regularizer, K-SVD and nonlocal means

4 Examples

In Figs. 6 and 7 examples of denoising results are shown, with comparison to total variation [28], non-local means [9] and K-SVD [15] (using the implementation of [27]). We show here simple cases of only a single category in the image. More elaborate algorithms are required for multiple-category images, using a pre-classification stage. A few images of similar nature were collected to create

the representative category set $\hat{S}_c$. The expert regularizer retains well the texture and characteristics of the clean version. In terms of SNR, it competes well with state-of-the-art methods.

5 Discussion and Conclusion

A new paradigm is suggested for regularization, when additional knowledge is at hand on the category of images to be processed.

In the ideal case, one would like that admissible images in the category would be unaffected by the regularizer, thus avoiding artifacts. This in turn suggests a few properties on expert regularizers, which are different then generic approaches.

A basic way to form an approximated ER was described, by precomputing weights, which define the attractor for the processed image, composed of parts of instances of the external category data. The problem then becomes convex and a simple solution yields the regularization result.

The approximation is a simplified approach, which does not take into account more complex interactions between regions or larger structural context. We gave examples of a single category case. In order to automatically select between several categories, one can use the regularizer value as an indicator (a sort of classifier) and select the one with minimal value.

The approach presented here gave encouraging results, further research is planned to investigate more comprehensive methods for solving these regularization problems.

References

1. Arias, P., Caselles, V., Facciolo, G.: Analysis of a variational framework for exemplar-based image inpainting. Multiscale Model. & Simul. 10(2), 473–514 (2012)
2. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing. Applied Mathematical Sciences, vol. 147. Springer (2002)
3. Aujol, J.F., Chambolle, A.: Dual norms and image decomposition models. IJCV 63(1), 85–104 (2005)
4. Bayram, I., Kamasak, M.E.: A Directional Total Variation. In: Proceedings of 20th European Signal Processing Conference, EUSIPCO 2012 (2012)
5. Berkels, B., Burger, M., Droske, M., Nemitz, O., Rumpf, M.: Cartoon extraction based on anisotropic image classification. In: Vision, Modeling, and Visualization Proceedings, pp. 293–300 (2006)
6. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imaging Sciences 3(3), 492–526 (2010)
7. Brox, T., Bruhn, A., Papenberg, 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)
8. Brox, T., Kleinschmidt, O., Cremers, D.: Efficient nonlocal means for denoising of textural patterns. IEEE Trans. Image Processing 17(7), 1083–1092 (2008)
9. Buades, A., Coll, B., Morel, J.-M.: A review of image denoising algorithms, with a new one. SIAM Multiscale Modeling and Simulation 4(2), 490–530 (2005)

10. Burger, M., Gilboa, G., Osher, S., Xu, J.: Nonlinear inverse scale space methods. Comm. in Math. Sci. 4(1), 179–212 (2006)
11. Chan, T.F., Esedoglu, S., Park, F.E.: A fourth order dual method for staircase reduction in texture extraction and image restoration problems. In: ICIP, pp. 4137–4140 (2010)
12. Coupé, P., Manjón, J.V., Fonov, V., Pruessner, J., Robles, M., Louis Collins, D.: Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. Neuroimage 54(2), 940–954 (2011)
13. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Processing 16(8), 2080–2095 (2007)
14. Droske, M., Rumpf, M.: A variational approach to non-rigid morphological registration. SIAM Appl. Math. 64(2), 668–687 (2004)
15. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Processing 15(12), 3736–3745 (2006)
16. Gilboa, G., Osher, S.: Nonlocal linear image regularization and supervised segmentation. SIAM Multiscale Modeling and Simulation 6(2), 595–630 (2007)
17. Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. Multiscale Modeling & Simulation, 1005–1028 (2008)
18. Hu, Y., Jacob, M.: Higher degree total variation (hdtv) regularization for image recovery. IEEE Transactions on Image Processing 21(5), 2559–2571 (2012)
19. Hutter, J., Grimm, R., Forman, C., Hornegger, J., Schmitt, P.: Vessel Adapted Regularization for Iterative Reconstruction in MR Angiography. In: Proceedings, 20th Annual Meeting, Int. Soc. Magnetic Resonance in Medicine (ISMRM) (2012)
20. Kindermann, S., Osher, S., Jones, P.: Deblurring and denoising of images by nonlocal functionals. SIAM Multiscale Modeling & Simul. 4(4), 1091–1115 (2005)
21. Lou, Y., Zhang, X., Osher, S., Bertozzi, A.: Image recovery via nonlocal operators. Journal of Scientific Computing 42(2), 185–197 (2010)
22. Meyer, Y.: Oscillating patterns in image processing and in some nonlinear evolution equations. The 15th Dean Jacquelines B. Lewis Memorial Lectures (March 2001)
23. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation based image restoration. SIAM Journal on Multiscale Modeling and Simulation 4, 460–489 (2005)
24. Osher, S., Esedoglu, S.: Decomposition of images by the anisotropic rudin-osher-fatemi model. Comm. Pure Appl. Math 57, 1609–1626 (2003)
25. Osher, S., Sole, A., Vese, L.: Image decomposition and restoration using total variation minimization and the H^{-1} norm. SIAM Multiscale Modeling and Simulation 1(3), 349–370 (2003)
26. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: IEEE Computer Society Conference on CVPR 2005, pp. 860–867 (2005)
27. Rubinstein, R., Zibulevsky, M., Elad, M.: Double sparsity: Learning sparse dictionaries for sparse signal approximation. IEEE Transactions on Image Processing 58(3), 1553–1564 (2010)
28. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
29. Tschumperlé, D., Brun, L.: Non-local regularization and registration of multi-valued images by pdes and variational methods on higher dimensional spaces. In: Mathematical Image Processing, pp. 181–197 (2011)
30. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner-Verlag, Stuttgart (1998)

A Spectral Approach to Total Variation

Guy Gilboa

Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa 32000, Israel

Abstract. The total variation (TV) functional is explored from a spectral perspective. We formulate a TV transform based on the second time derivative of the total variation flow, scaled by time. In the transformation domain disks yield impulse responses. This transformation can be viewed as a spectral domain, with somewhat similar intuition of classical Fourier analysis. A simple reconstruction formula from the TV spectral domain to the spatial domain is given. We can then design low-pass, high-pass and band-pass TV filters and obtain a TV spectrum of signals and images.

Keywords: Scale Space, total variation, image analysis.

1 Introduction

The total variation (TV) functional is today a fundamental regularizing tool in image processing. It is employed for denoising and deconvolution [30,12,26,28,27,20], optical-flow [8], tomographic reconstruction [31], texture and image analysis [7,4,3,35,21] and more. Since its introduction in [30] in the context of image processing many studies have been devoted to its analysis and interpretation, e.g. [12,26,13,14]. We attempt in this paper to further enhance the intuition and applicability of this functional to feature extraction and image analysis by formulating a spectral framework, where one can decompose and reconstruct images using the basic TV elements of the image.

Spectral analysis has been used extensively in the analysis and processing of signals modelled as stationary random processes (see e.g. [24,33]). For more complex non-stationary signals, such as images and speech, harmonic analysis methods were developed in the form of wavelets [17,25,18], spectral graph theory [15] and diffusion maps [16]. We explore a way to provide spectral information for total variation analysis.

In [32] Steidl et al have shown the close relations, and equivalence in a 1D discrete setting, of the Haar wavelets to both TV regularization [30] and TV flow [1]. This was later developed for a 2D setting in [37]. The development of features in the scale space framework [38,22,29,36] and the emergence of critical points were studied for example in [22,9,23,34,13,21]. This work relies on the established theory of the TV flow proposed by Andreu et al in [1] and further developed in [2,6,32,10,5,19] and the references therein.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 36–47, 2013.

2 The TV Spectral Framework

The scale-space approach is a natural way to define scale:

$$u_t = -p,\ u|_{t=0} = f,\ p \in \partial_u J(u), \tag{1}$$

where $\partial_u J(u)$ denotes the subdifferential of some regularizing functional $J(u)$.

We are interested in the total variation functional:

$$J(u) = \int_\Omega |Du|, \tag{2}$$

where Du denotes the distributional gradient of u. It is therefore natural to examine the total variation scale-space, known as total-variation flow [1], formally written as:

$$\begin{array}{ll} \frac{\partial u}{\partial t} = \mathrm{div}\left(\frac{Du}{|Du|}\right), & \text{in } (0,\infty)\times\Omega \\ \frac{\partial u}{\partial n} = 0, & \text{on } (0,\infty)\times\partial\Omega \\ u(0,x) = f(x), & \text{in } x \in \Omega, \end{array} \tag{3}$$

where Ω is the image domain (a bounded set in $\mathcal{R}^N$ with Lipschitz continuous boundary $\partial\Omega$). We assume f has sufficient spatial regularity.

We now give our line of thought how the transform was derived. Similar results may probably be obtained using other, more formal, approaches.

In Fourier analysis, the sine and cosine functions (or exponents with imaginary arguments) are the basic functions of the transform. They form impulses in the Fourier domain. How can this be generalized to the total variation domain? We begin by examining some atom-like elements in the TV sense. It is well known that disks are elementary structures for the TV functional. For instance, they satisfy the eigenvalue problem in $\mathcal{R}^N$: $\partial_u J(u) = \lambda u$ (where $\lambda \in \mathcal{R}$), which implies their shape stays the same during the entire evolution (their height decreases until they disappear). Analytic solutions for disk regularizations and evolutions were obtained for the TV regularization model [26,34], TV-flow [1,2,6], inverse-scale-space evolutions [11] and more.

Let us recall the analytic solution of a simple case: evolution of a single disk in two dimensions. The indicator function of a disk of radius r in $\mathcal{R}^2$ is:

$$I(x) = \begin{cases} 1, |x| < r \\ 0, \text{otherwise} \end{cases}$$

For a disk of height h, $hI(x)$, we have that $\partial_u J(u) = \frac{2}{r} I(x)$ for all t until the disk disappears. We denote by $t_d = \frac{hr}{2}$ the disappearance time.

The solution of the TV flow for $u(t)$ is therefore

$$u(t) = \begin{cases} (h - \frac{2}{r}t)I(x), & 0 \le t < t_d \\ 0, & \text{otherwise} \end{cases}$$

The first and second derivatives in time are:

$$u_t(t) = \begin{cases} -\frac{2}{r} I(x), & 0 \le t < t_d \\ 0, & \text{otherwise} \end{cases}$$

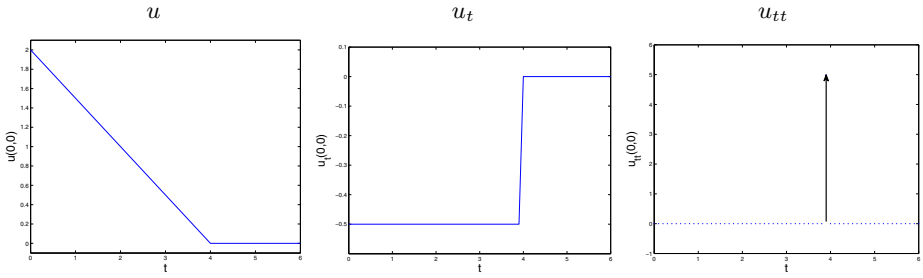


Fig. 1. Illustrating the evolution of a disk in $\mathcal{R}^2$. The value is within $|x| < r$, for example at $(x_1 = 0, x_2 = 0)$. The second derivative is an impulse at time t_d. [here we set $r = 4$, $h = 2$ and therefore $t_d = 4$].

$$u_{tt}(t) = \frac{2}{r}\delta(t - t_d)I(x),$$

where $\delta(t)$ denotes an impulse (Dirac delta) at $t = 0$. See Fig. 1 for an illustration.

We observe that u_{tt} yields an impulse of an elementary structure and is, therefore, a good candidate for a spectral representation. We would also like that the response will be invariant with respect to time. We normalize by multiplying it by the evolution time t. It will be seen later that this yields a straightforward reconstruction formula.

2.1 TV Transform

Let the TV transform be defined by

$$\phi(t) = u_{tt}t, \tag{4}$$

where $t \in (0, \infty)$ is the time parameter of the TV-flow, Equation (3), and u_{tt} is the second derivative in time of u in that flow.

Having defined $\phi(t) \in L^1(\Omega)$, we now need the inverse transform, which reconstructs a signal from all $\phi(t)$ responses. The reconstruction formula is very simple and is defined as:

$$w(x) = \int_0^\infty \phi(t)dt + \bar{f}, \tag{5}$$

where $\bar{f} = \frac{1}{\Omega}\int_\Omega f(x)dx$ is the mean value of the initial condition. Naturally, if we do not manipulate the spectral domain for filtering, we expect to reconstruct the image of the initial condition f, as stated in the following:

Theorem 1. *For $\phi(t)$ defined in* (4)*, the reconstruction formula* (5) *recovers $f \in BV(\Omega) \cap L^\infty(\Omega)$, that is $w(x) = f(x)$.*

Proof. We examine the left-term on the right hand side of Eq. (5). Integration by parts yields

$$\int_0^\infty \phi(t)dt = \int_0^\infty u_{tt}tdt = u_t t|_0^\infty - u|_0^\infty.$$

We use the property of finite extinction time of the TV flow. A two-dimensional proof by energy methods is given in [2] Th. 5. A more recent proof for all dimensions using energy estimates and Sobolev inequalities is given in [19] Th. 2.4, 2.5. In essence, this property means that for some $t_1 \in (0, \infty)$ we have $u(t) \equiv const, \forall t > t_1$. Therefore also $u_t(t) \equiv 0$ in a similar time range. The expression $u_t \in -\partial_u J(u)$ is finite for all $t \in [0, \infty)$ so that $u_t t|_{t=0} = 0$. We can therefore conclude that the left term $u_t t|_0^\infty = 0$. For Neumann boundary conditions the mean is unchanged, therefore $u|_{t\to\infty} = \bar{f}$. Using the initial condition we have $u|_0^\infty = \bar{f} - f$. □

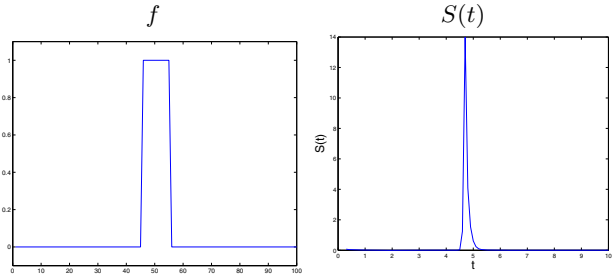


Fig. 2. A single one-dimensional disk and the corresponding numerical spectral response $S(t)$

Definition 1 (TV Spectral Response). *The TV spectral response for* $t \in (0, \infty)$ *is defined as:*

$$S(t) = \|\phi(t; x)\|_{L^1} = \int_\Omega |\phi(t; x)|dx.$$

The spectral response roughly corresponds to the amplitude of the response in a Fourier domain (see Fig. 3). If the response is high, a large "quantity" of the element $\phi(t)$ is contained in the image. If it is low, this element can be considered negligible. A response for one dimensional disk, as computed discretely, is depicted in Fig. 2. We will show in our experiments that, as can be expected, elements with high spectral response compose the main features of the image.

2.2 Spectral Filtering

Let $H(t)$ be a filter defined in the TV spectral domain as a real valued function of t. The filtered response $\phi_H(t)$ in the spectral domain is defined by:

$$\phi_H(t) = \phi(t)H(t). \tag{6}$$

The filtered response in the spatial domain is then the corresponding reconstruction procedure

$$f_H(x) = \int_0^\infty \phi_H(t)dt + \bar{f}, \tag{7}$$

An ideal filter in Fourier analysis eliminates completely energy of undesired frequencies while perfectly retaining frequencies in the desired range. We can now define analogous ideal filters in the TV spectral sense:

Definition 2 (Ideal Spectral Filters). *Let $t_1, t_2 \in [0, \infty)$. We define the following ideal spectral filters:*

(i) Ideal low-pass filter:

$$H_{LPF,t_1}(t) = \begin{cases} 0, 0 \le t < t_1 \\ 1, t_1 \le t < \infty \end{cases}$$

(ii) Ideal high-pass filter:

$$H_{HPF,t_1}(t) = \begin{cases} 1, 0 \le t < t_1 \\ 0, t_1 \le t < \infty \end{cases}$$

(iii) Ideal band-pass filter:

$$H_{BPF,t_1,t_2}(t) = \begin{cases} 0, 0 \le t < t_1 \\ 1, t_1 \le t < t_2 \\ 0, t_2 \le t < \infty \end{cases}$$

(iv) Ideal band-stop filter:

$$H_{BSF,t_1,t_2}(t) = \begin{cases} 1, 0 \le t < t_1 \\ 0, t_1 \le t < t_2 \\ 1, t_2 \le t < \infty \end{cases}$$

2.3 Feature Extraction

The spectral response $S(t)$ can be used to characterize an image. It informs us of the dominant scales and can be used when comparing images or as features for a machine learning algorithms. See Figs. 5, 6 for the spectral response and selected elements $\phi(t)$ of two image examples.

3 Examples

Examples demonstrating the qualitative properties of this transform are shown below.

In Fig. 3 a 1D example is shown and compared with classical low-pass-filtering in the Fourier domain. In the classical linear setting (bottom row) we have: $f_2 = \sin 2\pi\varphi_1 + 0.2 \sin 2\pi\varphi_2$, (in this specific example $\varphi_1 = 0.025$, $\varphi_2 = 0.15$).

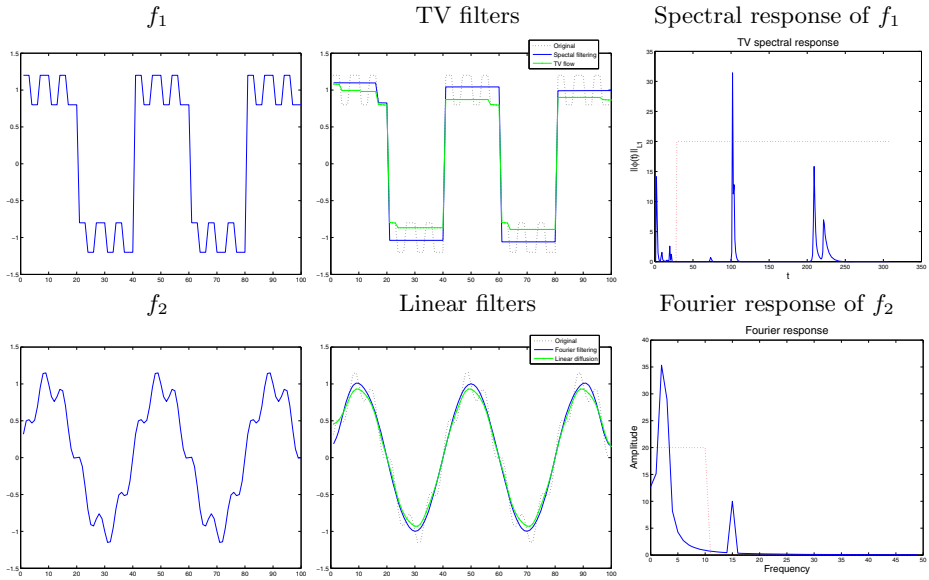


Fig. 3. One dimensional example of ideal low pass filtering versus scale-space low pass filtering. Top row, processing f_1 (left), middle - response by spectral filtering (full blue line), and by TV flow (dotted green line). On the top right the spectral response is shown. On the bottom row an analogue linear case filters f_2 with Fourier ideal LPF (full blue line) versus linear diffusion (dotted green line).

We compare two linear low-pass filters (LPF) - an ideal LPF and linear diffusion. The ideal LPF (shown on bottom, right, dotted line) keeps all low frequencies and sets to zero all frequencies above the threshold. The diffusion processes attenuates more softly the frequencies near the threshold (as it is not an ideal LPF). We observe that the ideal LPF retains the low frequency with better contrast.

A signal with similar properties, adapted for the TV case, is shown in Fig. 3 top row: $f_1 = \text{sign}(\sin 2\pi\varphi_1) + 0.2\,\text{sign}(\sin 2\pi\varphi_2)$. The spectral response $S(t)$ shows three active bands ($t < 30$ high oscillations, $t \approx 100$ low oscillations, and $200 < t < 250$ low amplitude step). TV flow is compared to ideal TV LPF, as defined above with filter threshold $t_1 = 30$. The filter response is illustrated in a dotted line at the top right. Note that in the TV spectral setting high frequencies are on the left side (small t values) as oppose to Fourier domain.

One can observe the very sharp transitions of the ideal LPF using the spectral filtering. Note that filtering with ideal LPF may result in too sharp transitions which can produce some reconstruction artifacts. This can be the case both in the linear and TV settings.

In Fig. 4 four circles of different sizes are processed. The ideal LPF is compared to TV-flow. In both cases the extent of filtering is such that the smallest circle completely vanishes. One can observe that the ideal LPF retains almost perfectly the larger three circles, whereas TV-flow erodes their contrast considerably.

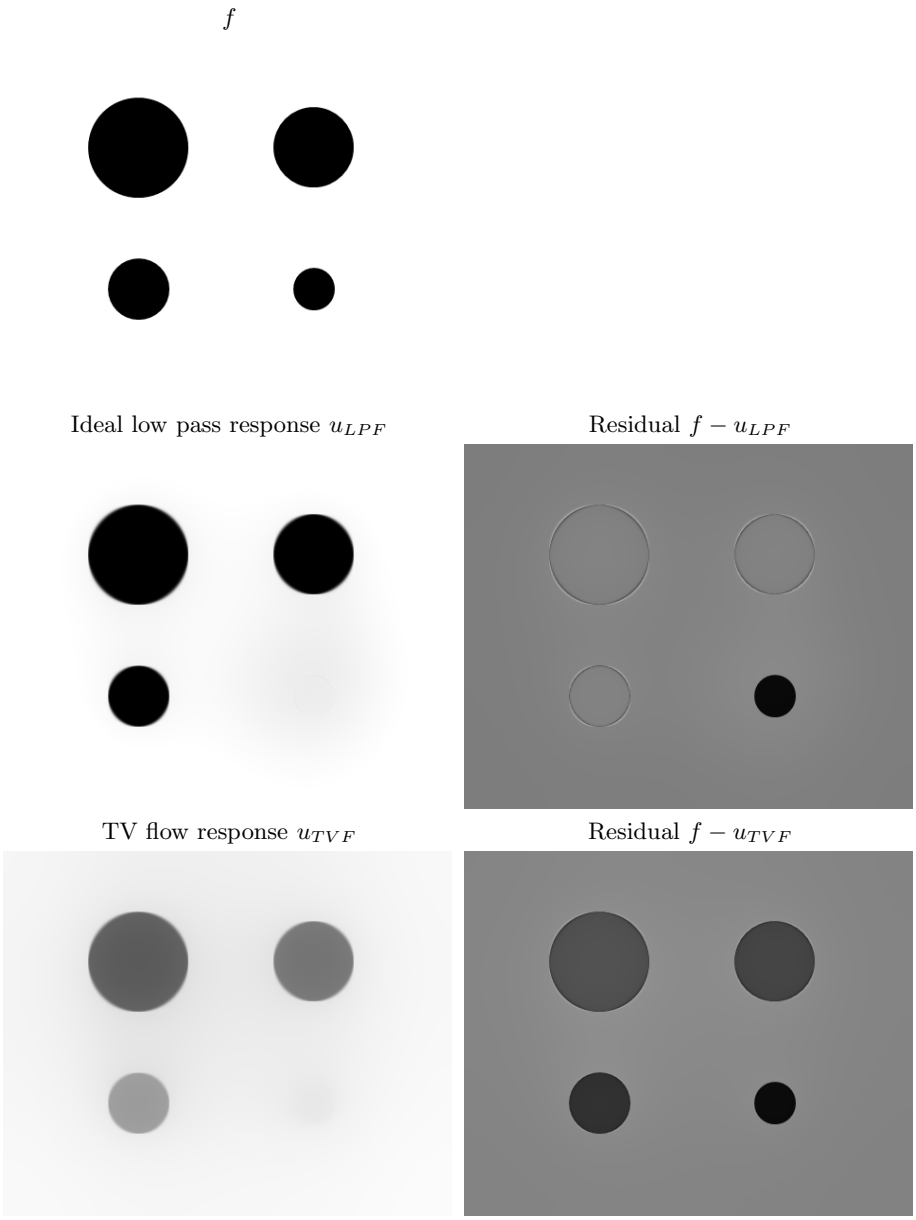


Fig. 4. Comparison between the ideal low-pass filter response and TV-flow. In both cases the response is shown for the minimal extent of filtering in which the smallest circle completely vanishes. One sees the considerable reduction of contrast of the larger circles in the TV-flow versus the sharp and stable results of the ideal TV LPF.

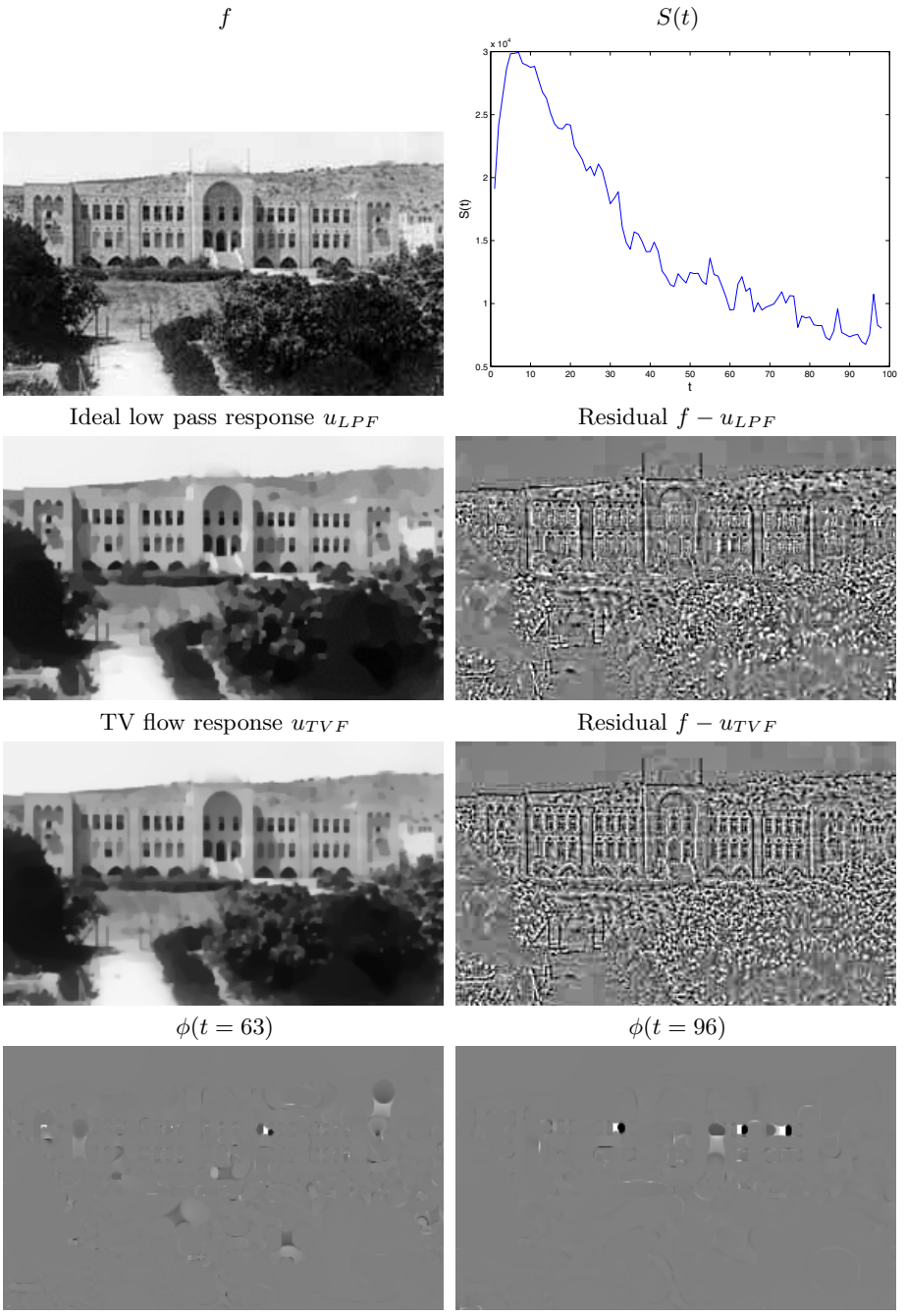


Fig. 5. Old Technion image. Results of ideal low pass filtering. This is compared to TV-flow with equivalent filtering in the L^2 sense (the norm of the residual, $\|f - u\|_{L^2}$, is the same). In addition, two examples of $\phi(t)$ are shown for different t values.

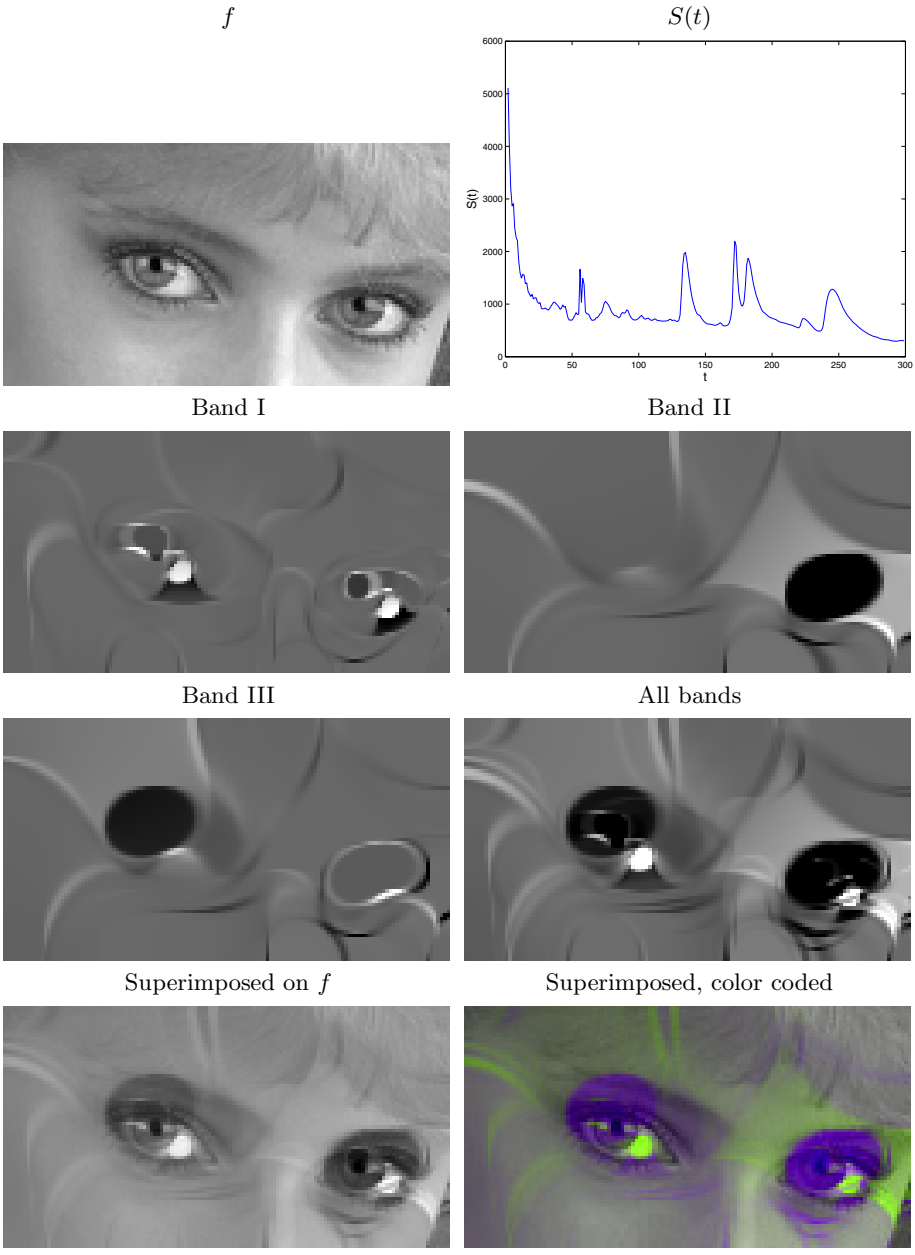


Fig. 6. Feature extraction example. Salient features ate depicted as spectral peaks (top right). The first three spectral peaks are shown as Bands I-III. These bands are reconstructed together at the third row, right. This reconstruction is then superimposed on the image to show the localization of the bands. Bottom right - a color coded visualization of the image with the selected bands.

In Fig. 5 an image of a building with landscape is examined. The ideal LPF response is shown along with a standard TV-flow filtering. In both cases the L^2 norm of the residual $f - u$ is the same. The ideal LPF exhibits sharper features. In addition two spectral elements $\phi(t)$ are shown. One can observe that the spatial response for any $\phi(t)$ is highly localized with very particular structures that emerge. The responses for the building windows (seen as black and white structure on the bottom row) highly resemble 2D Haar wavelets, which can be related to the analysis of [32,37]. Other structures can be related to the explicit solutions of structures which retain their characteristic function, as analyzed in [6].

In Fig. 6 a possible direction for image analysis is shown. The first most salient peaks in the spectrum are examined (around times 60, 130, 170). We band-pass filter them, as the response is not fully concentrated near a singular time point. The composed three bands are shown on the third row, right. They are superimposed back on the original image. It is shown that they contain meaningful and well localized features with semantic meaning (in this case the eyes). Therefore they may serve as good candidates for image features in higher-level vision algorithms (e.g. face detection).

4 Conclusion

In this study a TV transform and a corresponding reconstruction formula were presented. This transform yields large response to all image structure which disappear at highly concentrated time intervals during the TV flow evolution. We can regard these structure as the "atoms" of the image, with respect to the total variation functional and gain a spectral understanding in the TV sense.

We have shown numerically that these structures are well localized spatially and often represent significant image features with semantic meaning. Thus they can serve for image analysis and as input features to higher-level vision processing.

Extensions of this framework and relations to other TV-based formulations should be further investigated. For example, it may be the case that inverse-scale-space [11] can be interpreted as TV spectral low-pass filtering. Also other scale-spaces and regularization procedure, not based on the TV-functional, may be generalized using a similar approach.

References

1. Andreu, F., Ballester, C., Caselles, V., Mazn, J.M.: Minimizing total variation flow. Differential and Integral Equations 14(3), 321–360 (2001)
2. Andreu, F., Caselles, V., Daz, J.I., Mazon, J.M.: Some qualitative properties for the total variation flow. Journal of Functional Analysis 188(2), 516–547 (2002)
3. Aujol, J.F., Chambolle, A.: Dual norms and image decomposition models. IJCV 63(1), 85–104 (2005)

4. Aujol, J.F., Gilboa, G., Chan, T., Osher, S.: Structure-texture image decomposition – modeling, algorithms, and parameter selection. International Journal of Computer Vision 67(1), 111–136 (2006)
5. Bartels, S., Nochetto, R.H., Abner, J., Salgado, A.J.: Discrete total variation flows without regularization. arXiv preprint arXiv:1212.1137 (2012)
6. Bellettini, G., Caselles, V., Novaga, M.: The total variation flow in R^N. Journal of Differential Equations 184(2), 475–525 (2002)
7. Berkels, B., Burger, M., Droske, M., Nemitz, O., Rumpf, M.: Cartoon extraction based on anisotropic image classification. In: Vision, Modeling, and Visualization Proceedings, pp. 293–300 (2006)
8. Brox, T., Bruhn, A., Papenberg, 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)
9. Brox, T., Weickert, J.: A tv flow based local scale estimate and its application to texture discrimination. Journal of Visual Communication and Image Representation 17(5), 1053–1073 (2006)
10. Burger, M., Frick, K., Osher, S., Scherzer, O.: Inverse total variation flow. Multiscale Modeling & Simulation 6(2), 366–395 (2007)
11. Burger, M., Gilboa, G., Osher, S., Xu, J.: Nonlinear inverse scale space methods. Comm. In: Math. Sci. 4(1), 179–212 (2006)
12. Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. Numerische Mathematik 76(3), 167–188 (1997)
13. Chan, T.F., Esedoglu, S.: Aspects of total variation regularized l 1 function approximation. SIAM Journal on Applied Mathematics 65(5), 1817–1837 (2005)
14. Chan, T.F., Shen, J.: A good image model eases restoration - on the contribution of Rudin-Osher-Fatemi's BV image model (2002); IMA preprints 1829
15. Chung, F.R.K.: Spectral graph theory, vol. 92. Amer. Mathematical Society (1997)
16. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and Computational Harmonic Analysis 21(1), 5–30 (2006)
17. Daubechies, I., et al.: Ten lectures on wavelets, vol. 61. SIAM (1992)
18. Donoho, D.L.: De-noising by soft-thresholding. IEEE Transactions on Information Theory 41(3), 613–627 (1995)
19. Giga, Y., Kohn, R.V.: Scale-invariant extinction time estimates for some singular diffusion equations. Hokkaido University Preprint Series in Mathematics (963) (2010)
20. Gilboa, G., Sochen, N., Zeevi, Y.Y.: Estimation of optimal PDE-based denoising in the SNR sense. IEEE Trans. on Image Processing 15(8), 2269–2280 (2006)
21. Gilboa, G., Sochen, N., Zeevi, Y.Y.: Variational denoising of partly-textured images by spatially varying constraints. IEEE Trans. on Image Processing 15(8), 2280–2289 (2006)
22. Koenderink, J.J.: The structure of images. Biol. Cybern. 50, 363–370 (1984)
23. Luo, B., Aujol, J.F., Gousseau, Y.: Local scale measure from the topographic map and application to remote sensing images. Multiscale Modeling & Simulation 8(1), 1–29 (2009)
24. Marple Jr., S.L., Carey, W.M.: Digital spectral analysis with applications. The Journal of the Acoustical Society of America 86, 2043 (1989)
25. Meyer, Y.: Wavelets-algorithms and applications. Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation, 142 p., 1 (1993)
26. Meyer, Y.: Oscillating patterns in image processing and in some nonlinear evolution equations. The 15th Dean Jacquelines B. Lewis Memorial Lectures (March 2001)

27. Nikolova, M.: A variational approach to remove outliers and impulse noise. JMIV 20(1-2), 99–120 (2004)
28. Osher, S., Sole, A., Vese, L.: Image decomposition and restoration using total variation minimization and the H^{-1} norm. SIAM Multiscale Modeling and Simulation 1(3), 349–370 (2003)
29. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. PAMI 12(7), 629–639 (1990)
30. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
31. Sidky, E.Y., Pan, X.: Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. Physics in Medicine and Biology 53(17), 4777 (2008)
32. Steidl, G., Weickert, J., Brox, T., Mrázek, P., Welk, M.: On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. SIAM Journal on Numerical Analysis 42(2), 686–713 (2004)
33. Stoica, P., Moses, R.L.: Introduction to spectral analysis, vol. 89. Prentice Hall, Upper Saddle River (1997)
34. Strong, D., Chan, T.F.: Edge-preserving and scale-dependent properties of total variation regularization. Inverse Problems, 19(6), S165–S187 (2003)
35. Vese, L., Osher, S.: Modeling textures with total variation minimization and oscillating patterns in image processing. Journal of Scientific Computing 19, 553–572 (2003)
36. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner-Verlag, Stuttgart (1998)
37. Welk, M., Steidl, G., Weickert, J.: Locally analytic schemes: A link between diffusion filtering and wavelet shrinkage. Applied and Computational Harmonic Analysis 24(2), 195–224 (2008)
38. Witkin, A.P.: Scale space filtering. In: Proc. Int. Joint Conf. On Artificial Intelligence, pp. 1019–1023 (1983)

Convex Generalizations of Total Variation Based on the Structure Tensor with Applications to Inverse Problems*

Stamatios Lefkimmiatis[1], Anastasios Roussos[2], Michael Unser[1], and Petros Maragos[3]

[1] Biomedical Imaging Group, EPFL, Lausanne, Switzerland
[2] School of EECS, Queen Mary University of London, United Kindgdom
[3] School of ECE, National Technical University of Athens, Greece

Abstract. We introduce a generic convex energy functional that is suitable for both grayscale and vector-valued images. Our functional is based on the eigenvalues of the structure tensor, therefore it penalizes image variation at every point by taking into account the information from its neighborhood. It generalizes several existing variational penalties, such as the Total Variation and vectorial extensions of it. By introducing the concept of patch-based Jacobian operator, we derive an equivalent formulation of the proposed regularizer that is based on the Schatten norm of this operator. Using this new formulation, we prove convexity and develop a dual definition for the proposed energy, which gives rise to an efficient and parallelizable minimization algorithm. Moreover, we establish a connection between the minimization of the proposed convex regularizer and a generic type of nonlinear anisotropic diffusion that is driven by a spatially regularized and adaptive diffusion tensor. Finally, we perform extensive experiments with image denoising and deblurring for grayscale and color images. The results show the effectiveness of the proposed approach as well as its improved performance compared to Total Variation and existing vectorial extensions of it.

1 Introduction

This work deals with image reconstruction problems such as denoising and deblurring. We adopt their classical formulation as linear inverse problems: Let $\boldsymbol{u}(\boldsymbol{x}) = [u_1(\boldsymbol{x}) \dots u_M(\boldsymbol{x})] : \Omega \rightarrow \mathbb{R}^M$ be a generic vector-valued image with M channels that we seek to estimate. We consider that the observed image $\boldsymbol{v}$ is a degraded version of $\boldsymbol{u}$ according to the model: $\boldsymbol{z} = \mathrm{A}\boldsymbol{u} + \varepsilon$, where A is a linear operator and ε is the measurement noise. Following the common variational approach, we estimate $\boldsymbol{u}$ by minimizing a cost functional. This functional is typically the sum of a *data term* and a *regularization term*. The former measures the consistency between the estimate and the measurements, while the latter promotes certain solutions. A *regularization parameter* $\tau \geq 0$ balances the contributions of the two terms.

* S.L. and A.R. contributed equally and have joint first authorship. S.L. and M.U. were supported (in part) by the Hasler Foundation and the Indo-Swiss Joint Research Program. A.R. was supported by the ERC Starting Grant 204871-HUMANIS. P.M. was partially supported by the Greek research grant COGNIMUSE under the ARISTEIA action.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 48–60, 2013.

A widely used choice for the regularizer is the *Total Variation* (TV) [1], which is applied on grayscale images u (M=1) and is defined as:

$$\mathrm{TV}(u) = \int_{\Omega} \|\nabla u\|_2 \, \mathrm{d}\boldsymbol{x}. \tag{1}$$

TV owes its popularity to its ability to reconstruct images with well-preserved and sharp edges. This is due to the fact that it involves the gradient magnitude $\|\nabla u\|_2$ and it thus undergoes an L^1-type of behavior that does not over-penalize high variations of u. Its downside, however, is that it oversmooths homogeneous regions and creates strong staircase artifacts [2]. This behavior stems from its tendency to favor piecewise-constant solutions. Another drawback of TV is that the gradient magnitude, employed to penalize the image variation at every point $\boldsymbol{x}$, is too simple as an image descriptor; it relies only on $\boldsymbol{x}$ without taking into account the information from its neighborhood.

TV has been extended to general vector-valued image data in several ways, see e.g. [3–7]. Another related regularizer is the Beltrami functional [8], which has been recently generalized and unified with the Mumford-Shah functional [9]. In [10], TV is extended in an anisotropic way by incorporating the structure tensor of the image. But as in the image-driven anisotropic regularization of [5], this tensor is considered fixed and computed by the observed image. In all the above cases, the regularizers integrate a penalty of image variation that, as in TV, is completely local. On the contrary, in [11] a non-local version of TV is proposed, while in [12] an extension of the Beltrami framework that uses image patches is introduced. In [13] the authors propose a generic regularizer for vector-valued images that is based on the eigenvalues of the structure tensor, therefore it also takes into account the vicinity of each point. They show that its minimization is connected to tensor-based anisotropic diffusions.

In this work, to overcome the limitations of TV we adopt more sophisticated descriptors of image variations that generalize the gradient magnitude. We build upon the work of [13] and propose a generic convex energy functional that is based on the eigenvalues of the structure tensor. However, the current work departs from [13] in several ways. First, we provide more intuition about why the usage of the structure tensor's eigenvalues leads to effective generalizations of Total Variation. Also, the focus of [13] was in gradient descent flows of solely the regularizers, whereas in this work we combine the regularizers with data terms and we focus on the minimum rather than the flow towards the minimization. Further, we prove convexity of the proposed regularizers and we design an efficient algorithm for their minimization, which copes with their non-differentiability. Finally, in [13] the regularizers were applied only on image denoising, whereas our regularization framework is applied on more general linear inverse problems.

To the best of our knowledge, this is the first work that establishes a connection between **1)** generic anisotropic diffusion that is based on a spatially regularized and adaptive diffusion tensor (in the sense that this tensor contains convolutions with a kernel and is steered by the structure tensor field of the evolving image, as e.g. in [6, 14]) and **2)** minimization of **convex** regularizers that can be incorporated in an optimization framework and implemented efficiently using convex optimization algorithms.

2 Structure Tensor-Based Regularization

2.1 Directional Variation and Structure Tensor Revisited

In this section, we revisit and reformulate the well-established theory behind the structure tensor [14, 15], in a way that better motivates the regularizers that we will propose. The *vectorial directional derivative* of the vector-valued image $\boldsymbol{u}$ in an arbitrary 2D direction $\boldsymbol{n}$ ($\|\boldsymbol{n}\|_2$=1) is: $\partial\boldsymbol{u}/\partial\boldsymbol{n} = (J\boldsymbol{u})\boldsymbol{n}$, where $J\boldsymbol{u}$ is the *Jacobian matrix* of $\boldsymbol{u}$:

$$J\boldsymbol{u} = \begin{bmatrix} \nabla u_1 & \dots & \nabla u_M \end{bmatrix}^T . \tag{2}$$

The magnitude of the directional derivative $\|\partial\boldsymbol{u}/\partial\boldsymbol{n}\|_2$ yields a measure of the amount of change of the image $\boldsymbol{u}$ at the direction $\boldsymbol{n}$ for any specific point $\boldsymbol{x}$. This measure is typically unreliable since it is computed by concentrating completely at the point $\boldsymbol{x}$. In order to be more robust and capture also the behavior of the image $\boldsymbol{u}$ in the neighborhood of $\boldsymbol{x}$, we consider the weighted *root mean square* (RMS) of $\|\partial\boldsymbol{u}/\partial\boldsymbol{n}\|_2$, which we call (local) *directional variation*:

$$\mathrm{RMS}_K \left\{\|\partial\boldsymbol{u}/\partial\boldsymbol{n}\|_2\right\} = \sqrt{K * \|\partial\boldsymbol{u}/\partial\boldsymbol{n}\|_2^2} = \sqrt{\boldsymbol{n}^T (S_K\boldsymbol{u})\, \boldsymbol{n}}\,. \tag{3}$$

In the above equation $K(\boldsymbol{x})$ is a non-negative, rotationally symmetric convolution kernel (*e.g.*, a 2D Gaussian) that performs the weighted averaging and $S_K\boldsymbol{u}$ is the so-called *structure tensor* of the image $\boldsymbol{u}$ defined as:

$$S_K\boldsymbol{u} = K * \left(J\boldsymbol{u}^T J\boldsymbol{u}\right) . \tag{4}$$

Similarly to [6, 13], in the above definition we do not consider any pre-smoothing of the image before computing its Jacobian, since the single convolution with K seems sufficient for the needs of image regularization. Let $\lambda_+ \geq \lambda_-$ be the **eigenvalues** of $S_K(\boldsymbol{u})$ and $\boldsymbol{\theta}_+$, $\boldsymbol{\theta}_-$ be the corresponding unit **eigenvectors**. Also, let $\omega \in (-\pi, \pi]$ be the angle between the direction vector $\boldsymbol{n}$ and the eigenvector $\boldsymbol{\theta}_+$. Using the eigendecomposition of $S_K(\boldsymbol{u})$, we can express the directional variation (3) as a function of the angle ω:

$$V(\omega) \triangleq \mathrm{RMS}_K \left\{\|\partial\boldsymbol{u}/\partial\boldsymbol{n}\|_2\right\} = \sqrt{\lambda_+ \cos^2\omega + \lambda_- \sin^2\omega}\,. \tag{5}$$

If we consider the parametric equation $\boldsymbol{X}(\omega)$=$(\sqrt{\lambda_+}\cos\omega, \sqrt{\lambda_-}\sin\omega)$ of an ellipse with semi-major axis $\sqrt{\lambda_+}$ and semi-minor axis $\sqrt{\lambda_-}$, $V(\omega)$ can be interpreted as the distance of any point $\boldsymbol{X}(\omega)$ from the center of the ellipse. Therefore, $\sqrt{\lambda_+}$ corresponds to the **maximum** of the directional variation $V(\omega)$ (which is achieved for ω=0,π), whereas $\sqrt{\lambda_-}$ to the **minimum** of $V(\omega)$ (which is achieved for ω=$\pm\pi/2$).

2.2 Proposed Class of Regularizers

Based on the above analysis, we conclude that the vector $\sqrt{\boldsymbol{\lambda}} \triangleq (\sqrt{\lambda_+}, \sqrt{\lambda_-})$ is a synopsis of the function of local directional variation $V(\omega)$: it consists of the upper and lower bounds of this function. Therefore, we propose to generalize the Total Variation (1)

via replacing the gradient magnitude $\|\nabla u\|_2$ by ℓ_p norms of $\sqrt{\boldsymbol{\lambda}}$. More precisely, we propose the following type of regularizers, with $p \geq 1$:

$$E_p(\boldsymbol{u}) = \int_\Omega \left\|\sqrt{\boldsymbol{\lambda}}\right\|_p \mathrm{d}\boldsymbol{x} = \int_\Omega \left\|(\sqrt{\lambda_+}, \sqrt{\lambda_-})\right\|_p \mathrm{d}\boldsymbol{x}\,. \tag{6}$$

These norms measure the local variation of the image at each point more robustly than the gradient magnitude used in TV, as they take into account the variations in its neighborhood. At the same time, they incorporate richer information, since they depend not only on the maximum but also on the minimum of the directional variation. For instance, the response of these measures behaves differently at image edges than image corners.

We note that all the regularizers of the type (6) generalize TV. The reason is that for M=1 (grayscale images), if $K(\boldsymbol{x})$ is chosen to be the Dirac delta $\delta(\boldsymbol{x})$ (degenerated case where no convolution takes place at the computation of the structure tensor), then $\lambda_+ = \|\nabla u\|_2^2$ and λ_- is always 0. Therefore $\left\|\sqrt{\boldsymbol{\lambda}}\right\|_p = \|\nabla u\|_2$ for any $p \geq 1$.

Next, we describe some interesting cases of the proposed regularizers (6). For the following three cases of $E_p(\boldsymbol{u})$, the corresponding norms describe specific measures of the directional variation $V(\omega)$:

– p=1: $\left\|\sqrt{\boldsymbol{\lambda}}\right\|_1 = \sqrt{\lambda_+} + \sqrt{\lambda_-}$ corresponds (up to the scale factor 1/2) to the **mid-range** of $V(\omega)$, i.e. the average of minimum and maximum values of $V(\omega)$.
– p=2: $\left\|\sqrt{\boldsymbol{\lambda}}\right\|_2 = \sqrt{\lambda_+ + \lambda_-}$ corresponds (up to the scale factor $1/\sqrt{2}$) to the **RMS value** of $V(\omega)$, as it can be easily verified using Eq. (5).
– p=∞: $\left\|\sqrt{\boldsymbol{\lambda}}\right\|_\infty = \sqrt{\lambda_+}$ corresponds to the **maximum** of $V(\omega)$.

Invariance Properties. Since our regularizers are generalizations of TV, one should expect that they also share the same invariance properties. Two of the most favorable ones are the rotation invariance and contrast covariance (1-homogeneity), which according to Proposition 1 are indeed preserved (see Supplementary Material for the proof).

Proposition 1. *The energy functional* (6) *is rotation invariant and contrast covariant.*

2.3 Connections to Tensor-Based Anisotropic Diffusion and Previous Work

The proposed class of regularizers $E_p(\boldsymbol{u})$ (6) is a special case of the more generic form proposed in [13]: $E(\boldsymbol{u}) = \int_\Omega \psi\left(\lambda_+, \lambda_-\right) \mathrm{d}\boldsymbol{x}$. This special case corresponds to cost functions of the form $\psi\left(\lambda_+, \lambda_-\right) = \|(\sqrt{\lambda_+}, \sqrt{\lambda_-})\|_p$.

In order to make the cost function in the proposed regularizers differentiable, let us consider the relaxation $E_{p,\epsilon}(\boldsymbol{u})$ that arises by setting $\psi\left(\lambda_+, \lambda_-\right) = \varphi_{p,\epsilon}(\lambda_+, \lambda_-) \triangleq \left\|(\sqrt{\epsilon + \lambda_+}, \sqrt{\epsilon + \lambda_-})\right\|_p$, where $\epsilon > 0$ is a small constant. Note that we need this relaxation only to establish connections to anisotropic diffusion and not for the actual optimization, since our optimization algorithm, described in Section 4, can cope with the non-differentiability of the functionals. By applying [13, Theorem 1], we find the relation of minimizing the proposed regularizers with anisotropic diffusion:

Corollary 1. *The functional gradient of $E_{p,\epsilon}$ w.r.t. each image component u_i is:*

$$\frac{\delta E_{p,\epsilon}}{\delta u_i} = -\mathrm{div}\left(D \nabla u_i\right),\; D = K * \left(2\frac{\partial \varphi_{p,\epsilon}}{\partial \lambda_+}\boldsymbol{\theta}_+ \otimes \boldsymbol{\theta}_+ + 2\frac{\partial \varphi_{p,\epsilon}}{\partial \lambda_-}\boldsymbol{\theta}_- \otimes \boldsymbol{\theta}_-\right). \tag{7}$$

This gradient is a nonlinear anisotropic diffusion term, where the diffusion tensor D contains convolutions with the kernel K and depends on the structure tensor of the image. For the following characteristic choices of p, the diffusion tensor is given as:

– p=1: $D = K*\Big(\frac{1}{\sqrt{\epsilon+\lambda_+}}\boldsymbol{\theta}_+\otimes\boldsymbol{\theta}_+ + \frac{1}{\sqrt{\epsilon+\lambda_-}}\boldsymbol{\theta}_-\otimes\boldsymbol{\theta}_-\Big)$. This tensor is adapting on the image structures in a conceptually similar way to tensor-based anisotropic diffusion methods, such as [6, 14]: 1) in the homogeneous regions (small λ_+,λ_-) it is strong and isotropic, 2) near the edges (large λ_+, small λ_-) it is weaker and mainly oriented by the edges, whereas 3) near the corners (large λ_+,λ_-) it is even weaker.

– p=2: $D = \Big(K * \frac{1}{\sqrt{2\epsilon+K*\sum_i \|\nabla u_i\|_2^2}}\Big) I_{2\times 2}$. This tensor is always isotropic, it thus corresponds to a diffusion coefficient. Similarly to nonlinear diffusion methods, such as [16, 17], this coefficient is strong in the homogeneous regions, whereas weaker near edges.

– p=∞: $D = K * \Big(\frac{1}{\sqrt{\epsilon+\lambda_+}}\boldsymbol{\theta}_+ \otimes \boldsymbol{\theta}_+\Big)$. This tensor is always highly anisotropic and oriented perpendicular to image edges.

Further Relations to Previous Work. As already stated, the proposed regularizers are special cases of the more generic functional of [13]. Furthermore, the special subcase of $p = 1$ corresponds to the so-called **Tensor Total Variation** of [13]. In addition, several other variational methods emerge as special cases of the proposed regularizers. The kernel K that corresponds to all these cases is the Dirac delta $\delta(\boldsymbol{x})$, which means that the regularization does not exploit information from the neighborhood of each point and is thus less coherent. As already described in Section 2.2, if we set $K(\boldsymbol{x})$=$\delta(\boldsymbol{x})$ and M=1 (grayscale images) then for all choices of $p\geq$1 we recover the **Total Variation** [1]. The case of $K(\boldsymbol{x})$=$\delta(\boldsymbol{x})$, $M>$1 and p=2 corresponds to the usually called **Vectorial TV** (TV-F) [3, 4], which is the most common extension of TV to vector-valued images. Finally, the case $K(\boldsymbol{x})$=$\delta(\boldsymbol{x})$, $M>$1 and p=∞ corresponds to the method of [7], which the authors call **Natural Vectorial TV** (TVJ).

3 Patch-Based Jacobian and the Discrete Structure Tensor

In this section, we introduce a generalization of the Jacobian of an image, based on local weighted patches (see e.g. [12, 13]). This new operator, which we call *patch-based Jacobian*, contains weighted shifted versions of the Jacobian of $\boldsymbol{u}$, whose weights are determined by the convolution kernel K. Then, we employ it to express the structure tensor in a novel way, which finally leads us to derive an equivalent definition of the proposed regularizers. This alternative definition provides more intuition, facilitates the proof of convexity and opens the way for an efficient optimization strategy.

Hereafter, we will focus on the discrete formulation of the image reconstruction problem. We consider that the discretized vector-valued image $\boldsymbol{u}$ is defined on a rectangular grid with unary steps and that the corresponding intensities of each channel m of $\boldsymbol{u}$ ($m=1,..,M$) are rasterized in the vector $\boldsymbol{u}_m$ of size N. By combining all the image channels, we have that $\boldsymbol{u} \in \mathbb{R}^{NM}$. We use the index n=1, .., N to refer to a specific pixel of the grid and we denote by $\boldsymbol{x}_n$ the coordinates of that pixel. Furthermore, we consider

that the convolution kernel K (see Eq. (4)) has been discretized and truncated in order to have compact support $\mathcal{S}=\{-L_K,..,L_K\}^2$, where L_K is a non-negative integer.

We define the *patch-based Jacobian* of an image $\boldsymbol{u}$ as the linear mapping $\boldsymbol{J}_K : \mathbb{R}^{NM} \mapsto \mathcal{X}$, where $\mathcal{X} \triangleq \mathbb{R}^{N\times(LM)\times 2}$ and $L = (2L_K+1)^2$. For each pixel n we denote by $[\boldsymbol{J}_K\boldsymbol{u}]_n$ the element of $\boldsymbol{J}_K\boldsymbol{u}$ that corresponds to that pixel and we construct it by: **1)** taking the discrete versions of the $M \times 2$ Jacobian matrices (2) of $\boldsymbol{u}$ for all the pixels $\{\boldsymbol{x}_n - \boldsymbol{y} : \boldsymbol{y} \in \mathcal{S}\}$ in the $\mathcal{S}$-neighborhood of pixel $\boldsymbol{x}_n$, **2)** weighting these matrices with the window function $\boldsymbol{w}[\boldsymbol{y}] \triangleq \sqrt{K[\boldsymbol{y}]}$ and **3)** stacking all these matrices vertically in the matrix $[\boldsymbol{J}_K\boldsymbol{u}]_n$, whose dimension is $(LM) \times 2$. Formally, the patch-based Jacobian can be defined as:

$$[\boldsymbol{J}_K\boldsymbol{u}]_n^T = \begin{bmatrix} [P_{\boldsymbol{y}_1}\circ\boldsymbol{D}_h\boldsymbol{u}_1]_n \cdots [P_{\boldsymbol{y}_L}\circ\boldsymbol{D}_h\boldsymbol{u}_1]_n \cdots [P_{\boldsymbol{y}_1}\circ\boldsymbol{D}_h\boldsymbol{u}_M]_n \cdots [P_{\boldsymbol{y}_L}\circ\boldsymbol{D}_h\boldsymbol{u}_M]_n \\ [P_{\boldsymbol{y}_1}\circ\boldsymbol{D}_v\boldsymbol{u}_1]_n \cdots [P_{\boldsymbol{y}_L}\circ\boldsymbol{D}_v\boldsymbol{u}_1]_n \cdots [P_{\boldsymbol{y}_1}\circ\boldsymbol{D}_v\boldsymbol{u}_M]_n \cdots [P_{\boldsymbol{y}_L}\circ\boldsymbol{D}_v\boldsymbol{u}_M]_n \end{bmatrix}, \quad (8)$$

where $\boldsymbol{D}_h$, $\boldsymbol{D}_v$ are the two components of the discrete gradient, the shift vectors $\boldsymbol{y}_l$ ($l=1,\ldots,L$) are the elements of the lattice $\mathcal{S}$, and $P_{\boldsymbol{y}_l}$ are weighted shift operators. The latter are designed to properly handle the image boundaries according to the assumed extension (*e.g.*, mirroring) and are defined as:

$$[P_{\boldsymbol{y}_l} \circ \boldsymbol{D}_h\boldsymbol{u}_m]_n = \boldsymbol{w}[\boldsymbol{y}_l]\, \boldsymbol{D}_h\{\boldsymbol{u}_m\}[\boldsymbol{x}_n - \boldsymbol{y}_l] \ . \quad (9)$$

Next, we equip the space $\mathcal{X}$ (which is the target space of $\boldsymbol{J}_K$) with the inner product $\langle\cdot\,,\,\cdot\rangle_{\mathcal{X}}$ and norm $\|\cdot\|_{\mathcal{X}}$. To define them, let $\boldsymbol{X}, \boldsymbol{Y} \in \mathcal{X}$, with $\boldsymbol{X}_n, \boldsymbol{Y}_n \in \mathbb{R}^{(LM)\times 2}\,\forall\ n = 1,\ldots,N$. Then we have:

$$\langle \boldsymbol{X}\,,\,\boldsymbol{Y}\rangle_{\mathcal{X}} = \textstyle\sum_{n=1}^{N} \mathrm{tr}\left(\boldsymbol{Y}_n^T\boldsymbol{X}_n\right) \quad (10) \qquad \text{and} \qquad \|\boldsymbol{X}\|_{\mathcal{X}} = \sqrt{\langle \boldsymbol{X}\,,\,\boldsymbol{X}\rangle_{\mathcal{X}}}\,, \quad (11)$$

where $\mathrm{tr}\,(\cdot)$ is the trace operator. For the Euclidean space $\mathbb{R}^{NM}$ we use the standard inner product $\langle\cdot\,,\,\cdot\rangle_2$ and norm $\|\cdot\|_2$.

The adjoint of $\boldsymbol{J}_K$ is the discrete linear operator $\boldsymbol{J}_K^* : \mathcal{X} \mapsto \mathbb{R}^{NM}$, defined by:

$$\langle \boldsymbol{Y}\,,\,\boldsymbol{J}_K\boldsymbol{u}\rangle_{\mathcal{X}} = \langle \boldsymbol{J}_K^*\boldsymbol{Y}\,,\,\boldsymbol{u}\rangle_2 \ . \quad (12)$$

The following Proposition expresses $\boldsymbol{J}_K^*$ in a form that facilitates its computation (see Supplementary Material for the proof).

Proposition 2. *The adjoint operator $\boldsymbol{J}_K^*$ of the patch-based Jacobian is given by:*

$$[\boldsymbol{J}_K^*\boldsymbol{Y}]_{(n,m)} = \textstyle\sum_{l=1}^{L} -\boldsymbol{div}\left[P_{\boldsymbol{y}_l}^* \circ \boldsymbol{Y}^{((m-1)L+l,:)}\right]_n, \quad (13)$$

*where **div** is the discrete divergence, P^* is the adjoint of the shift operator P, and $\boldsymbol{Y}_n^{(k,:)}$ corresponds to the k-th row of the n-th matrix component, $\boldsymbol{Y}_n \in \mathbb{R}^{(LM)\times 2}$, of $\boldsymbol{Y}$.*

Having introduced the necessary tools, we can now express the structure tensor in a novel way. This is done in Proposition 3 (see Supplementary Material for a proof).

Proposition 3. *Let $[\boldsymbol{S}_K\boldsymbol{u}]_n$ be the discretized structure tensor at pixel n, which is defined by adopting discrete derivatives and discrete convolution in* (2) *and* (4)*, respectively. Then, it can be written in terms of the patch-based Jacobian as:*

$$[\boldsymbol{S}_K\boldsymbol{u}]_n = [\boldsymbol{J}_K\boldsymbol{u}]_n^T\,[\boldsymbol{J}_K\boldsymbol{u}]_n \ . \quad (14)$$

Since λ_+, λ_- are the eigenvalues of $[\boldsymbol{S}_K \boldsymbol{u}]_n$, the singular values of $[\boldsymbol{J}_K \boldsymbol{u}]_n$ are $\sqrt{\lambda_+}, \sqrt{\lambda_-}$. This connection permits us to use Schatten norms [18] and the patch-based Jacobian so as to write the proposed regularizers (6) (after discretization) as:

$$E_p(\boldsymbol{u}) = \sum_{n=1}^{N} \left\| [\boldsymbol{J}_K \boldsymbol{u}]_n \right\|_{\mathcal{S}_p}, \text{ with } p \geq 1. \tag{15}$$

Note that for a matrix $\boldsymbol{Z}$, its **Schatten norm** of order p ($\mathcal{S}_p$ norm) denoted by $\|\boldsymbol{Z}\|_{\mathcal{S}_p}$, is defined as $\|\boldsymbol{\sigma}(\boldsymbol{Z})\|_p$, with $\boldsymbol{\sigma}(\boldsymbol{Z})$ the vector of the singular values of $\boldsymbol{Z}$. This equivalent formulation of $E_p(\boldsymbol{u})$ provides more intuition about the fact that the proposed regularizers are effective generalizations of TV. More precisely, $[\boldsymbol{J}_K \boldsymbol{u}]_n$ encodes the vectorial variation of the image $\boldsymbol{u}$ in the vicinity of the pixel n. Therefore, the Schatten norms of this matrix provide different measures of the local variation of $\boldsymbol{u}$, by taking into account its neighborhood in a weighted manner. In addition, an important contribution of the above result is that the expression (6), which involves the eigenvalues of the nonlinear structure tensor, has been transformed to the expression (15) that is much easier to handle, since it depends on a norm of a linear operator acting on $\boldsymbol{u}$. We refer to the proposed regularizers as STV-[k] (Structure tensor Total Variation) where the character [k] denotes the order of the Schatten norm. For example, for the cases of p=1,2 and ∞ we use the notations STV-N (**N**uclear norm), STV-F (**F**robenius norm) and STV-S (**S**pectral norm) respectively.

It has now become straight-forward to show the following important result:

Proposition 4. *The regularizer $E_p(\boldsymbol{u})$ is convex w.r.t $\boldsymbol{u}$ $\forall p \geq 1$.*

Proof. The regularizer of Eq. (15) is clearly convex since it results as the composition of a norm (mixed ℓ_1-$\mathcal{S}_p$ norm; see (17) for its definition) and the linear operator $\boldsymbol{J}_K$.

4 Energy Minimization Strategy

4.1 Proximal Map Evaluation

In this section we propose an efficient algorithm that provides a numerical solution to the following problem, for any $p \geq 1$:

$$\underset{\boldsymbol{u} \in \mathbb{R}^{NM}}{\arg\min} \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{z}\|_2^2 + \psi(\boldsymbol{u}), \text{ with } \psi(\boldsymbol{u}) \triangleq \tau E_p(\boldsymbol{u}) + \iota_{\mathcal{C}}(\boldsymbol{u}), \tag{16}$$

where $\mathcal{C}$ is a convex set that represents additional constraints on the solution and $\iota_{\mathcal{C}}$ is its indicator function: $\iota_{\mathcal{C}}(\boldsymbol{u})$ takes the value 0 for $\boldsymbol{u} \in \mathcal{C}$ and ∞ otherwise. Note that the case of no constraints is simply the special case $\mathcal{C}$=$\mathbb{R}^{NM}$. The solution of (16) corresponds to evaluating the proximal map [19] of the function ψ at $\boldsymbol{z}$ and arises in most linear inverse imaging problems, including the ones considered by this work.

To proceed with our minimization approach, we write the energy E_p in the compact form $E_p(\boldsymbol{u}) = \|\boldsymbol{J}_K \boldsymbol{u}\|_{1,p}$, where $\|\cdot\|_{1,p}$ corresponds to the mixed ℓ_1-$\mathcal{S}_p$ norm, which for an argument $\boldsymbol{X} = \left[\boldsymbol{X}_1^T, \ldots, \boldsymbol{X}_N^T\right]^T \in \mathcal{X}$ is defined as

$$\|\boldsymbol{X}\|_{1,p} = \sum_{n=1}^{N} \|\boldsymbol{X}_n\|_{\mathcal{S}_p}. \tag{17}$$

Next, we rely on the following lemma to derive a dual formulation of our problem.

Lemma 1 ([20]). *Let $p \geq 1$, and let q be the conjugate exponent of p, i.e., $\frac{1}{p} + \frac{1}{q} = 1$. Then, the mixed norm $\|\cdot\|_{\infty,q}$ is dual to the mixed norm $\|\cdot\|_{1,p}$.*

Using Lemma 1 and the fact that the dual of the dual norm is the original norm [21], we write (17) in the equivalent form:

$$\|\boldsymbol{X}\|_{1,p} = \max_{\boldsymbol{\Omega} \in \mathcal{B}_{\infty,q}} \langle \boldsymbol{\Omega}\,,\, \boldsymbol{X} \rangle_{\mathcal{X}}\ , \tag{18}$$

where $\mathcal{B}_{\infty,q}$ denotes the ℓ_∞-$\mathcal{S}_q$ unit-norm ball, defined as the set

$$\mathcal{B}_{\infty,q} \triangleq \left\{ \boldsymbol{\Omega} = \left[\boldsymbol{\Omega}_1^T, \ldots, \boldsymbol{\Omega}_N^T\right]^T \in \mathcal{X} : \|\boldsymbol{\Omega}_n\|_{\mathcal{S}_q} \leq 1, \forall n = 1, \ldots, N \right\}. \tag{19}$$

Note that from (19), it is clear that the orthogonal projection onto $\mathcal{B}_{\infty,q}$ can be obtained by projecting separately each submatrix $\boldsymbol{\Omega}_n$ onto a unit-norm $\mathcal{S}_q$ ball ($\mathcal{B}_{\mathcal{S}_q}$).

Combining (12) and (18) we re-write (16) as

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u} \in \mathcal{C}} \tfrac{1}{2} \|\boldsymbol{u} - \boldsymbol{z}\|_2^2 + \tau \max_{\boldsymbol{\Omega} \in \mathcal{B}_{\infty,q}} \langle \boldsymbol{J}_K^* \boldsymbol{\Omega}\,,\, \boldsymbol{u} \rangle_2\ . \tag{20}$$

This formulation naturally leads us to the following minimax problem:

$$\min_{\boldsymbol{u} \in \mathcal{C}} \max_{\boldsymbol{\Omega} \in \mathcal{B}_{\infty,q}} \mathcal{L}\left(\boldsymbol{u}, \boldsymbol{\Omega}\right)\ , \tag{21}$$

where $\mathcal{L}\left(\boldsymbol{u}, \boldsymbol{\Omega}\right) = \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{z}\|_2^2 + \tau \left\langle \boldsymbol{J}_K^* \boldsymbol{\Omega}\,,\, \boldsymbol{u} \right\rangle_2$. The function $\mathcal{L}$ is strictly convex in $\boldsymbol{u}$ and concave in $\boldsymbol{\Omega}$, and thus, we have the guarantee that a saddle-value of $\mathcal{L}$ is attained [21]. Therefore, the order of the minimum and the maximum in (21) does not affect the solution and $\hat{\boldsymbol{u}}$ can be equivalently obtained by solving the problem:

$$\max_{\boldsymbol{\Omega} \in \mathcal{B}_{\infty,q}} \min_{\boldsymbol{u} \in \mathcal{C}} \left(\tfrac{1}{2} \|\boldsymbol{u} - (\boldsymbol{z} - \tau \boldsymbol{J}_K^* \boldsymbol{\Omega})\|_2^2 + \tfrac{1}{2} \|\boldsymbol{z}\|_2^2 - \tfrac{1}{2} \|(\boldsymbol{z} - \tau \boldsymbol{J}_K^* \boldsymbol{\Omega})\|_2^2 \right). \tag{22}$$

The inner minimization in (22) has an exact solution:

$$\hat{\boldsymbol{u}} = \mathit{\Pi}_{\mathcal{C}} \left(\boldsymbol{z} - \tau \boldsymbol{J}_K^* \hat{\boldsymbol{\Omega}} \right), \tag{23}$$

where $\mathit{\Pi}_{\mathcal{C}}$ denotes the orthogonal projection onto the convex set $\mathcal{C}$, while $\hat{\boldsymbol{\Omega}}$ is the maximizer of the dual problem:

$$\max_{\boldsymbol{\Omega} \in \mathcal{B}_{\infty,q}} \left(\phi\left(\boldsymbol{\Omega}\right) \triangleq \tfrac{1}{2} \|\mathit{\Pi}_{\mathcal{C}}\left(\boldsymbol{c}\right) - \boldsymbol{c}\|_2^2 + \tfrac{1}{2} \|\boldsymbol{z}\|_2^2 - \tfrac{1}{2} \|\boldsymbol{c}\|_2^2 \right), \tag{24}$$

where $\boldsymbol{c} = \boldsymbol{z} - \tau \boldsymbol{J}_K^* \boldsymbol{\Omega}$. Contrary to the primal problem (16), where the function to be minimized is not continuously differentiable, the dual one in (24) involves the function ϕ which is smooth and has a well defined gradient. To compute it, we use the result in [22, Lemma 4.1], according to which the gradient of a function $h\left(\boldsymbol{x}\right) = \|\boldsymbol{x} - \mathit{\Pi}_{\mathcal{C}}\left(\boldsymbol{x}\right)\|_2^2$ is equal to: $\nabla h\left(\boldsymbol{x}\right) = 2\left(\boldsymbol{x} - \mathit{\Pi}_{\mathcal{C}}\left(\boldsymbol{x}\right)\right)$. Based on that, we get:

$$\nabla \phi\left(\boldsymbol{\Omega}\right) = \tau \boldsymbol{J}_K \mathit{\Pi}_{\mathcal{C}}\left(\boldsymbol{z} - \tau \boldsymbol{J}_K^* \boldsymbol{\Omega}\right). \tag{25}$$

Algorithm 1: Evaluation of the proximal map of $\psi(\boldsymbol{u})$.

Input: $\boldsymbol{z}, \tau > 0, p \geq 1, \mathit{\Pi}_{\mathcal{C}}$.
Initialization: $\boldsymbol{\Psi}_1 = \boldsymbol{\Omega}_0 = \mathbf{0} \in \mathcal{X}, t_1 = 1$.
while *stopping criterion is not satisfied* **do**
  $\boldsymbol{v} = \mathit{\Pi}_{\mathcal{C}}\left(\boldsymbol{z} - \tau \boldsymbol{J}_K^* \boldsymbol{\Psi}_n\right)$;
  $\boldsymbol{\Omega}_n \leftarrow \mathit{\Pi}_{\mathcal{S}_q}\left(\boldsymbol{\Psi}_n + \frac{1}{8\tau}\boldsymbol{J}_K \boldsymbol{v}\right)$;
  $t_{n+1} \leftarrow \frac{1+\sqrt{1+4t_n^2}}{2}$;
  $\boldsymbol{\Psi}_{n+1} \leftarrow \boldsymbol{\Omega}_n + \left(\frac{t_n - 1}{t_{n+1}}\right)(\boldsymbol{\Omega}_n - \boldsymbol{\Omega}_{n-1})$;
  $n \leftarrow n + 1$;
end
return $\hat{\boldsymbol{u}} = \mathit{\Pi}_{\mathcal{C}}\left(\boldsymbol{z} - \tau \boldsymbol{J}_K^* \boldsymbol{\Omega}_{n-1}\right)$;

Then, we use (25) to design a gradient-based algorithm that solves (24). The solution of our primal problem (16) is obtained in two steps: **1)** we find the maximizer of the dual objective function (24), and **2)** we obtain the solution using (23).

Since (24) does not have a closed-form solution ($\boldsymbol{J}_K$ has not a stable inverse), we employ Nesterov's iterative method [23] for smooth functions. This is a gradient-based scheme that exhibits state-of-the art convergence rates of one order higher than the standard gradient-ascent method. A detailed description of the overall algorithm is provided in Algorithm 1. Note that for the implementation of Algorithm 1, we need to perform a projection of a matrix onto a Schatten norm ball. This is discussed in the next section.

4.2 Efficient Projection of Rectangular Matrices

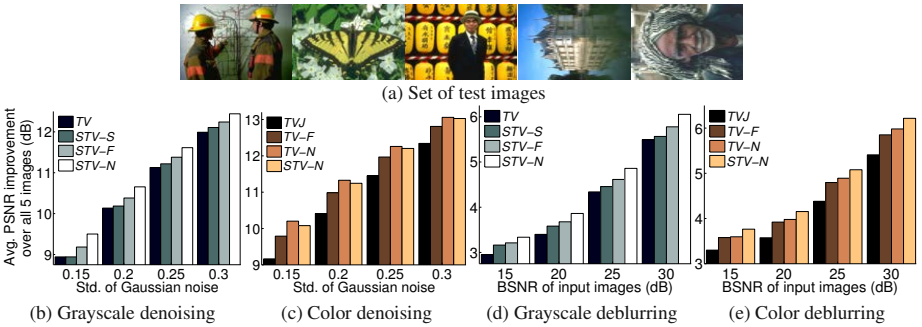
Let $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ with an SVD decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ and $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ with $n = \min(n_1, n_2)$. According to [20, Proposition 1], the projection of $\boldsymbol{X}$ onto the unit-norm $\mathcal{S}_q$ ball is computed as:

$$\mathit{\Pi}_{\mathcal{S}_q}(\boldsymbol{X}) = \boldsymbol{U}\boldsymbol{\Sigma}_q\boldsymbol{V}^T, \tag{26}$$

where $\boldsymbol{\Sigma}_q = \operatorname{diag}(\boldsymbol{\sigma}_q)$ and $\boldsymbol{\sigma}_q$ is the projection of the singular values of $\boldsymbol{\Sigma}$ onto the ℓ_q unit-norm ball $\mathcal{B}_q = \left\{\boldsymbol{\sigma} \in \mathbb{R}_+^n : \|\boldsymbol{\sigma}\|_q \leq 1\right\}$. The projection in (26) requires the singular vectors and singular values of $\boldsymbol{X}$. In our case $n_2 = 2 < n_1$, and we compute the projection in an efficient way as described next. First, we note that the matrix $\boldsymbol{X}^T\boldsymbol{X}$ is $n_2 \times n_2$ symmetric with an eigenvalue decomposition $\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T$. Therefore, for $n_2 = 2$ both $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ can be computed in closed form. Now, if $\boldsymbol{\Sigma}^+$ is the pseudoinverse matrix of $\boldsymbol{\Sigma}$, defined as: $\boldsymbol{\Sigma}^+ = \operatorname{diag}\left(\sigma_1^{-1}, \ldots, \sigma_k^{-1}, 0, \ldots, 0\right)$, with σ_k the smallest nonzero singular value, then $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{V}\boldsymbol{\Sigma}^+$. Using this result we write (26) as:

$$\mathit{\Pi}_{\mathcal{S}_q}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{\Sigma}_q\boldsymbol{V}^T, \tag{27}$$

and we avoid the computation of $\boldsymbol{U}$. We note that the same idea was explored in [7] for efficiently computing the projection step that arises in the minimization of TVJ.



(a) Set of test images

(b) Grayscale denoising (c) Color denoising (d) Grayscale deblurring (e) Color deblurring

Fig. 1. Performance measures for different regularization methods

4.3 General Linear Inverse Problems

Algorithm 1 applies only in cases where no linear operator is involved in the data term. For general inverse problems, under the proposed regularization framework, one needs to solve a minimization problem of the form:

$$\underset{\boldsymbol{u}\in\mathbb{R}^{NM}}{\arg\min}\, \tfrac{1}{2}\left\|\boldsymbol{A}\boldsymbol{u}-\boldsymbol{z}\right\|_2^2 + \tau E_p\left(\boldsymbol{u}\right) + \iota_{\mathcal{C}}\left(\boldsymbol{u}\right), \forall p \geq 1, \tag{28}$$

where $\boldsymbol{A}$ is a linear degradation operator, which for most practical cases is ill-conditioned. To solve this type of problems we employ the MFISTA algorithm [22], which exhibits state-of-the-art convergence rates. Nevertheless, our algorithm is still a critical part, since the main step of MFISTA requires the evaluation of the proximal map that we investigated in Section 4.1. For a detailed description of the MFISTA approach we refer the readers to the Supplementary Material accompanying this paper.

5 Experimental Results

To evaluate the effectiveness of the proposed generic regularization framework, we report results for the problems of gray/color image denoising and deblurring. For both linear inverse problems we use the set of images shown in Fig. 1(a), taken from the Berkeley BSDS500 dataset. In the image denoising setting we consider four different standard deviations of Gaussian noise, σ_w=$\{0.15, 0.2, 0.25, 0.3\}$. In the image deblurring setting we consider a Gaussian blur kernel, which has a support of 9×9 pixel and a standard deviation σ_b=6 pixels, and four noise levels corresponding to BSNR=$\{15, 20, 25, 30\}$ dB respectively, where BSNR is the *Blurred Signal to Noise Ratio*, defined as $\text{BSNR} = \text{var}\left(\boldsymbol{A}\boldsymbol{u}\right)/\sigma_w^2$.

In Figs. 1(b)-1(e) we report the average performance, in terms of *Peak Signal to Noise Ratio* (PSNR), over all tested images. For the grayscale experiments, we compare TV against three variants of our functional (STV-S, STV-F, STV-N). For the color case, we compare the results we obtained with our STV-N regularizer against those obtained using TVJ [7] and TV-F [3]. In these comparisons, we also include the variant of STV-N where no smoothing is involved in the computation of the structure tensor (TV-N), which is also a novel regularizer. For the sake of consistency among comparisons,

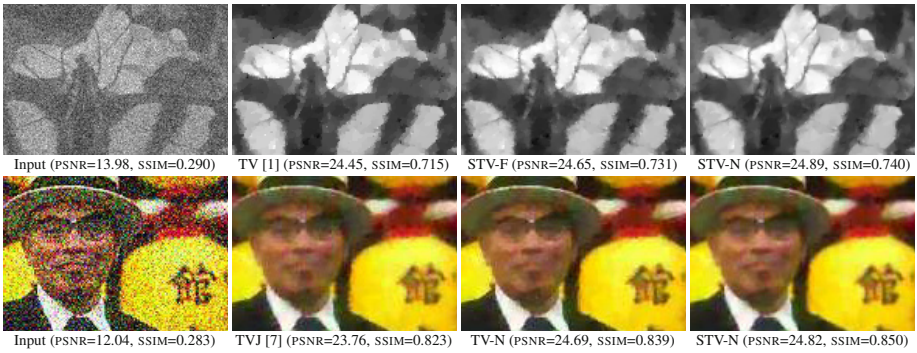


Fig. 2. Grayscale *(first row)* and Color *(second row)* image denoising examples. The PSNR and the Structural Similarity index (SSIM) are also reported.

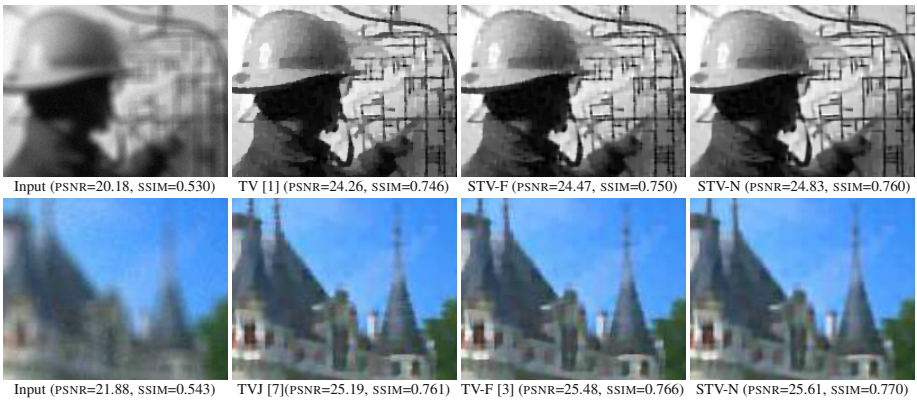


Fig. 3. Grayscale *(first row)* and Color *(second row)* image deblurring examples. The PSNR and SSIM measures are also reported.

the reported results for each regularizer were obtained using the individualized regularization parameter that gives the best PSNR performance. Moreover, all reconstructions are performed under box constraints, meaning that the restored intensities must lie in the convex set $\mathcal{C} = \left\{\boldsymbol{u} \in \mathbb{R}^N | u_n \in [0\,,1]\,\forall n = 1, \ldots, N\right\}$. Finally, in all the STV regularizers, we choose the structure tensor's convolution kernel to be a Gaussian with a support of 3×3 pixels.

From the reported results, we observe that in the grayscale case the best performance for both image denoising and deblurring is achieved by STV-N. On the other hand, TV has the worst performance, especially in deblurring, since in denoising its performance is very close to STV-S. In the color denoising experiments, TV-N performs slightly better than STV-N, and both are superior than the competitive regularizers. However, when we consider the image deblurring problem, STV-N behaves better than TV-N and provides the best results. This can be attributed to the fact that deblurring is a more ill-conditioned problem and, thus, the use of a convolution kernel K is more critical.

Finally, apart from the quantitative comparisons, conclusions for the effectiveness of the proposed approach can be drawn by a visual inspection of the results provided in Figs. 2-3. From these examples we can verify that the proposed STV regularizers perform better in reducing the staircase effects of other Total Variation methods and better reconstruct the edges and the other image structures.

6 Conclusions

In this work we introduced a family of regularizers that is based on the eigenvalues of the structure tensor. In image denoising and deblurring problems these regularizers can more accurately restore image edges than TV and its vectorial extensions, and, thus, lead to improved results. Furthermore, based on a novel formulation of the structure tensor, we proved convexity for the regularizers and designed an efficient primal-dual algorithm for their minimization. Since TV-based reconstructions are used in a host of imaging applications, an interesting research direction is to investigate whether our regularizers can also lead to an improved performance in other inverse problems, as well. This will be the subject of future work.

References

1. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
2. Chan, T., Marquina, A., Mulet, P.: High-order total variation-based image restoration. SIAM J. Sci. Comput. 22, 503–516 (2000)
3. Sapiro, G.: Color snakes. Comp. Vision and Image Understanding 68(2) (1997)
4. Blomgren, P., Chan, T.: Color TV: Total Variation methods for restoration of vector-valued images. IEEE Trans. on Image Processing 7(3), 304–309 (1998)
5. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in PDE-based computation of image motion. Int. Journ. of Computer Vision 45(3), 245–264 (2001)
6. Tschumperlé, D., Deriche, R.: Vector-valued image regularization with PDE's: A common framework for different applications. IEEE T-PAMI 27(4), 506–517 (2005)
7. Goldluecke, B., Strekalovskiy, E., Cremers, D.: The natural vectorial total variation which arises from geometric measure theory. SIAM J. Imaging Sci. 5, 537–563 (2012)
8. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE Trans. on Image Processing 7, 310–338 (1998)
9. Sochen, N., Bar, L.: The Beltrami-Mumford-Shah functional. In: Scale Space and Variational Methods in Computer Vision, pp. 183–193 (2012)
10. Grasmair, M., Lenzen, F.: Anisotropic total variation filtering. Applied Mathematics & Optimization 62, 323–339 (2010)
11. Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. Multiscale Modeling & Simulation 7(3), 1005–1028 (2008)
12. Wetzler, A., Kimmel, R.: Efficient Beltrami flow in patch-space. In: Scale Space and Variational Methods in Computer Vision, pp. 134–143 (2012)
13. Roussos, A., Maragos, P.: Tensor-based image diffusions derived from generalizations of the total variation and Beltrami functionals. In: Proc. Int. Conf. on Image Processing (2010)
14. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner, Stuttgart (1998)
15. Jähne, B.: Digital Image Processing. Springer (2002)

16. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE T-PAMI 12(7), 629–639 (1990)
17. Catté, F., Lions, P., Morel, J., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journ. Numer. Anal. 29(1), 182–193 (1992)
18. Bhatia, R.: Matrix Analysis. Springer (1997)
19. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. 4(4), 1168–1200 (2005)
20. Lefkimmiatis, S., Ward, J., Unser, M.: Hessian Schatten-norm regularization for linear inverse problems. IEEE Trans. Image Processing 22(5), 1873–1888 (2013)
21. Rokcafellar, R.T.: Convex Analysis. Princeton Univ. Press, Princeton (1970)
22. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Processing 18, 2419–2434 (2009)
23. Nesterov, Y.: A method for solving a convex programming problem with convergence rates $O\left(1/k^2\right)$. Soviet Math. Dokl 27, 372–376 (1983)

Adaptive Second-Order Total Variation: An Approach Aware of Slope Discontinuities

Frank Lenzen[1], Florian Becker[1], and Jan Lellmann[2]

[1] Heidelberg Collaboratory for Image Processing (HCI), Heidelberg, Germany
[2] DAMTP, University of Cambridge, UK

Abstract. Total variation (TV) regularization, originally introduced by Rudin, Osher and Fatemi in the context of image denoising, has become widely used in the field of inverse problems. Two major directions of modifications of the original approach were proposed later on. The first concerns *adaptive* variants of TV regularization, the second focuses on *higher-order* TV models. In the present paper, we combine the ideas of both directions by proposing *adaptive second-order* TV models, including one *anisotropic* model. Experiments demonstrate that introducing adaptivity results in an improvement of the reconstruction error.

Keywords: second-order total variation, adaptive, anisotropic, directional, TV, TGV, slope discontinuities.

1 Introduction

In 1992 Rudin, Osher and Fatemi [14] proposed to apply the total-variation (TV) semi-norm for regularization in a variational framework for image denoising. Their approach not only had a significant impact in the area of image restoration, but in the whole field of inverse problems. Since then, various modifications and improvements have been contributed by the community. Several publications have been devoted to the idea of adaptive TV regularization methods, where the regularization varies locally depending on the noise level or the image content [3,4,6,7]. Non-local TV models (e.g. [10]), which have proven as effective variants, can also be regarded as adaptive methods, since they use image information to locally determine the regularization weights. Another subclass of TV approaches are the anisotropic or directional methods, where the regularization not only depends on the location but also on the local orientation of the signal to be reconstructed [1,8,12,18]. TV regularization has the major benefit that it allows piecewise constant signals to be recovered. Recent works have shown that in certain cases it might be beneficial to assume even higher regularity of the signal, and thus introduced higher-order regularization schemes [2,9,11,13,15,17].

Contribution. We combine adaptive and second-order TV approaches into one regularization framework. Such a combination has not been proposed up to now. Our approach uses information on local image structures, in particular on edges and slope discontinuities obtained from structure tensors applied to the image

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 61–73, 2013.

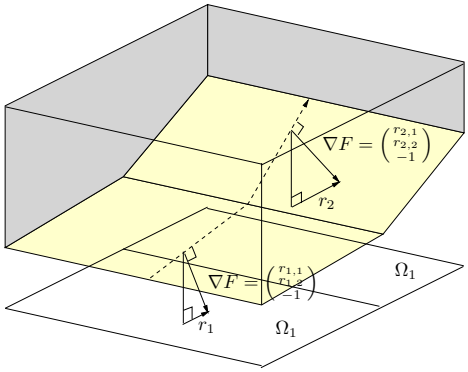


Fig. 1. Graph $\Gamma = (x, y, u(x, y))^\top$ (yellow) of a continuous and piecewise affine function u with a discontinuity in the gradient (interface between Ω_1 and Ω_2). The epigraph of u is the volume above Γ, represented as the super-level set of $F(x, y, z) = u(x, y) - z$. On the graph the gradient ∇F of F coincides with the surface normal of Γ.

and its epigraph. We demonstrate, that our approach can be applied to the standard second-order TV regularization as well as to regularization with total generalized variation (TGV) [2] and infimal convolution (IC) [17]. Moreover, we propose a new *anisotropic* second-order TV model and show its advantages over the isotropic models.

Paper Organization. In Sect. 2 we describe how the information on image structures required to steer adaptive regularization is retrieved. In Sect. 3 we consider adaptive second-order TV models. Experiments are provided in Sect. 4.

2 Detecting Discontinuities in Piecewise Affine Functions

In this section we provide an approach to extract information about the direction and location of edges and the location of slope discontinuities from a given input image. The first task is already addressed in literature. We rely on the standard structure tensor and just briefly recall the required definitions. However, we will see that this approach is not suitable for detecting slope discontinuities (sharp bends, kinks). For this second task, we propose a new approach.

2.1 Edge Detection

In the following, we represent an image as a function $u : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^2$. For detecting edges in u we follow the standard approach and use the classical structure tensor (cf. [5]) to identify regions with high gradient magnitude. To this end, let

$$S_u(x, y) := (\nabla u_\sigma(x, y) \nabla u_\sigma(x, y)^\top)_\rho. \tag{1}$$

be the standard structure tensor calculated on u_σ, which is obtained from u by convolution with a Gaussian kernel with variance σ^2. Furthermore, $(.)_\rho$ denotes

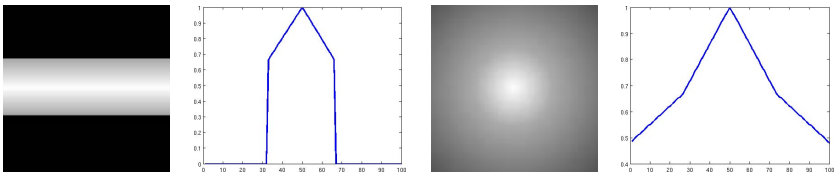


Fig. 2. Test images *roof* and *cone hat* for detecting slope discontinuities

a component-wise convolution of each entry with a Gaussian kernel with variance ρ^2. We denote by $\lambda_1^S(x,y), \lambda_2^S(x,y)$ the eigenvalues of $S_u(x,y)$ ordered with decreasing value, i.e. $\lambda_1^S(x,y) \geq \lambda_2^S(x,y)$. Moreover, we consider the eigenvector v^S to the eigenvalue λ_1^S. It is known that along edges in the image, λ_1^S takes large values, whereas λ_2^S is almost zero. Thus, $d^S(x,y) := \lambda_1^S(x,y) - \lambda_2^S(x,y)$ indicates the presence of edges. We define $E^S : \Omega \to [0,1]$ as $E^S(x,y) := \min\{c\, d^S(x,y), 1\}$ with some constant $c > 0$. In Sect. 3 we make use of the edge indicating function E^S together with the vector field v^S.

2.2 Slope Discontinuities

The standard structure tensor as considered so far is sufficient to identify discontinuities (edges) in u. We now focus on regions where u is continuous but has discontinuities in its first derivatives. In addition, we assume that u is piecewise affine. This assumption is in view of our ansatz in Sect. 3 to determine u as the solution of a second-order TV approach. For the sake of simplicity, let us consider a prototypical function model with only one discontinuity, which locally represents a part of a larger image: we assume that Ω can be divided into two segments $\Omega_i, i = 1,2$ such that u is affine in each segment, i.e. u can be represented as

$$u(x,y) = \begin{cases} r_1^\top \binom{x}{y} + b_1 & \text{if } (x,y) \in \Omega_1, \\ r_2^\top \binom{x}{y} + b_2 & \text{if } (x,y) \in \Omega_2, \end{cases} \tag{2}$$

for Ω_i open, such that $\Omega_1 \cap \Omega_2 = \emptyset$ and $\overline{\Omega_1} \cup \overline{\Omega_2} = \Omega$, and $r_i \in \mathbb{R}^2$, $b_i \in \mathbb{R}$ for $i = 1,2$. Fig. 1 illustrates such a prototypical function u.

The aim of this section is to derive a method to detect the case where $r_1 \neq r_2$. To this end, we consider the epigraph of u defined as the super-level set $\{(x,y,z) \mid F(x,y,z) \geq 0\}$ of $F(x,y,z) := u(x,y) - z$. In order to detect (surface) edges of the graph (i.e. locations, where the slope changes), we now apply the three-dimensional structure tensor to F, i.e. $\left((\nabla F)(\nabla F)^\top\right)_\rho$, where $\nabla F(x,y,z) = (\partial_x u^2 + \partial_y^2 u + 1)^{-\frac{1}{2}}(\partial_x u, \partial_y u, -1)^\top$. Note that ∇F is constant in z. Since we are only interested in edges of the graph $\Gamma := \{(x,y,z) \mid F(x,y,z) = 0\}$ (i.e. slope discontinuities), we restrict this structure tensor to Γ:

$$T_u(x,y) := \left((\nabla \tilde{F}(x,y))(\nabla \tilde{F}(x,y))^\top\right)_\rho, \tag{3}$$

where $\nabla \tilde{F}_u(x,y) := \nabla F(x,y,u(x,y)))$. We observe that $\nabla \tilde{F}_u(x,y)$ is the normal to the graph Γ at $(x,y,u(x,y))$.

Remark 1. The following two scenarios are of particular interest:

Within an Affine Region: For an affine function u, $T_u(x,y)$ has exactly one non-zero eigenvalue. This is due to the fact that in this case $\nabla \tilde{F}_u(x,y)$ is constant and convolution of $\nabla \tilde{F}_u \nabla \tilde{F}_u^\top$ does not change the rank.

Interface between Two Affine Regions of Different Slope: For such u, $T_u(x,y)$ sums up two different directions $(r_{1,1}, r_{1,2}, -1)$ and $(r_{2,1}, r_{2,2}, -1)$: rewriting the convolution of the matrix entries as a weighted integral,

$$\begin{aligned} T_u(x,y) &= (\nabla \tilde{F}_u \nabla \tilde{F}_u^\top)_\rho = \int_\Omega w(x) \nabla \tilde{F}_u \nabla \tilde{F}_u^\top \, dx \\ &= w_1 \begin{pmatrix} r_{1,1} \\ r_{1,2} \\ -1 \end{pmatrix} (r_{1,1}, r_{1,2}, -1) + w_2 \begin{pmatrix} r_{2,1} \\ r_{2,2} \\ -1 \end{pmatrix} (r_{2,1}, r_{2,2}, -1) \end{aligned} \tag{4}$$

with $w_i := \int_{\Omega_i} w(x) \, dx$, we observe that in (4) two rank-1 matrices are added up. Each matrix has one non-zero eigenvalue $w_i \cdot \|(r_{i,1}, r_{i,2}, -1)\|_2^2$ with corresponding eigenvector $v_i = (r_{i,1}, r_{i,2}, -1)$. Since the eigenvectors are linear dependent only if $r_1 = r_2$, $T_u(x,y)$ has rank 2 near the discontinuity, where $r_1 \neq r_2$.

In the following we denote by $\lambda_i^T(x,y), i = 1,2,3$ the eigenvalues of $T_u(x,y)$ in decreasing order. As an indicator for the existence of slope discontinuities we propose to use $\lambda_2^T(x)$. This is motivated by the fact that, similar to the standard structure tensor in $2D$, $T_u(x,y)$ reveals two eigenvalues significantly larger that 0 at edges of the graph, while in regions of constant slope the second eigenvalue becomes 0. Therefore the magnitude of the second eigenvalue can be used to distinguish between both cases. We propose $E^T : \Omega \to [0,1]$, $E^T(x) := \min(c\lambda_2^T(x), 1)$ with some constant $c > 0$ as an indicator for regions of slope discontinuities. In order to be less sensitive to edges, which are already covered by the standard structure tensor, we use an upwind scheme to compute the gradient in (3). In practice, it is advisable to use the pre-smoothed u_σ (cf. Sect. 2.1) instead of u to be robust against noise.

To demonstrate the benefits of using E^T to detect slope discontinuities, we compare our approach to one approach based on the standard structure tensor and one based on curvature, see Fig. 3. We observe that our approach detects slope discontinuities more reliably than the competitive methods.

3 Adaptive Second-Order Total Variation

In the following we discuss three state-of-the-art approaches for second-order total variation (TV) regularization. First, we focus on the straightforward approach of combining two TV semi-norms of first and second order [12,15]. We generalize this approach to allow for anisotropic regularization with locally adaptive strength. In addition, we consider two alternative approaches – infimal convolution (IC) [17] and total generalized variation (TGV) [2] – and propose a spatially adaptive choice of the regularization parameters.

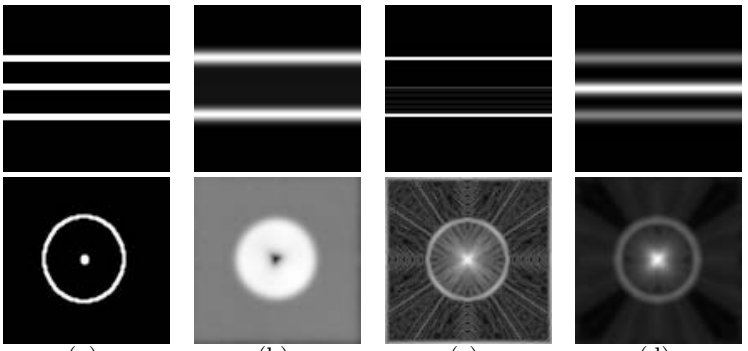


Fig. 3. Detecting slope discontinuities using the standard structure tensor (b), a curvature based approach (c), and the proposed method (d) in the test images depicted in Fig. 2 (black=0, white=1). In both cases the standard structure tensor fails to detect the slope discontinuities as shown in the ideal result (a) (middle line in the first image, ring and center point in the second image). Only the proposed approach detects the slope discontinuity in the first test image (top row). On the second test image (bottom row), the proposed approach provides a less noisy and more precise result than the curvature based approach.

3.1 Proposed Approach

Let $BV^2(\Omega)$ (Ω open, bounded, with Lipschitz boundary) be the space of functions with bounded first and second-order TV, i.e. $u \in BV^2(\Omega)$ iff $u \in L^1(\Omega)$ and

$$TV^l(u) := \sup \left\{ \int_\Omega u \operatorname{div}^l \varphi \, dx \mid \varphi \in C_c^\infty(\Omega, \mathbb{R}^{2l}), \forall x \in \Omega : \|\varphi(x)\|_2 \leq 1 \right\}, \quad (5)$$

is finite for $l = 1, 2$. Here, div^1 is the divergence operator and $\operatorname{div}^2 \varphi := \partial_{xx}\varphi_1 + \partial_{yx}\varphi_2 + \partial_{xy}\varphi_3 + \partial_{yy}\varphi_4$, where $\varphi = (\varphi_1, \varphi_2, \varphi_3, \varphi_4)^\top$. Note that for $u \in BV^2(\Omega)$ we have $\partial_x u, \partial_y u \in L^1(\Omega)$. For details on $BV^2(\Omega)$ we refer to [16, Chapter 9.8]. A standard denoising approach with first and second-order TV regularization consists in minimizing the functional

$$\mathcal{F}_{TV^2}(u) := \tfrac{1}{2}\|u - f\|_{L^2}^2 + \alpha\, TV(u) + \beta\, TV^2(u) \quad (6)$$

for given data $f \in L^2(\Omega)$ and regularization parameters $\alpha, \beta > 0$. We generalize this approach in two ways. Firstly, we allow α, β to vary depending on the location, i.e., $\alpha, \beta : \Omega \to \mathbb{R}_+$. Secondly, we allow anisotropic, i.e. directionally dependent regularization. To this end, we consider the optimization problem

$$\mathcal{F}(u) := \tfrac{1}{2}\|u - f\|_{L^2}^2 + \mathcal{R}_1(u) + \mathcal{R}_2(u) \quad (7)$$

with two regularization terms $\mathcal{R}_1(u)$ and $\mathcal{R}_2(u)$ defined as follows. For first-order TV, we use anisotropic TV regularization (cf. [7]) given as

$$\mathcal{R}_1(u) := \int_\Omega \left(\nabla u^\top(x) A(x) \nabla u(x)\right)^{\frac{1}{2}} dx, \quad (8)$$

for some matrix-valued mapping $A : \Omega \to \mathbb{R}^{2\times 2}_{sym}$, where $A(x)$ is symmetric and positive semi-definite at every x. Every such matrix $A(x)$ can be written as $A(x) = (v(x), v^\perp(x)) \begin{pmatrix} \alpha_1(x) & 0 \\ 0 & \alpha_2(x) \end{pmatrix} (v(x), v^\perp(x))^\top$ with some vector field $v(x)$, $\|v(x)\|_2 = 1$. We observe that (8) leads to an anisotropic regularization with strength $\alpha_1(x)$ in direction of $v(x)$ and $\alpha_2(x)$ in direction of $v^\perp(x)$.

For adaptive second-order TV regularization we propose a new approach, which we motivate by the smooth case $u \in C^2(\Omega)$: for arbitrary $\varphi \in C_c^\infty(\Omega, \mathbb{R}^4)$ we have

$$\int_\Omega (\mathrm{div}^2\, \varphi) u \; dx = \int_\Omega \langle \varphi, \nabla^2 u \rangle \; dx, \tag{9}$$

where $\nabla^2 u := (\partial_{xx} u, \partial_{xy} u, \partial_{yx} u, \partial_{yy} u)^\top$. For a given normalized vector field $v(x) = (v_1(x), v_2(x))^\top \in \mathbb{R}^2$, $\|v(x)\|_2 = 1$, we represent φ as $\varphi = t_1 w_1 + t_2 w_2 + s_1 w_3 + s_2 w_4$, where $t, s \in \mathbb{R}^2$ and $w_1 := (v_1, v_2, 0, 0)^\top$, $w_2 := (0, 0, v_1, v_2)^\top$, $w_3 := (v_1^\perp, v_2^\perp, 0, 0)$ and $w_4 := (0, 0, v_1^\perp, v_2^\perp)$. Note that $\{w_i\}_i$ form an orthonormal basis of $\mathbb{R}^4$. Then, standard calculus shows

$$\langle \varphi, \nabla^2 u \rangle = t^\top (Hu) v + s^\top (Hu)(v^\perp) \quad \text{for } Hu := \begin{pmatrix} \partial_{xx} u & \partial_{xy} u \\ \partial_{yx} u & \partial_{yy} u \end{pmatrix}. \tag{10}$$

Now we calculate $\beta_1 \|(Hu)v\|_2 + \beta_2 \|(Hu)v^\perp\|_2$ for some weighting constants $\beta_1, \beta_2 > 0$. To this end, we take in (10) the supremum over $t \in B_{\beta_1}(0)$ and $s \in B_{\beta_2}(0)$, where $B_r(0)$ denotes the ball centered at 0 with radius r, and derive

$$\sup_{t \in B_{\beta_1}(0), s \in B_{\beta_2}(0)} \varphi(\nabla^2 u) = \beta_1 \|(Hu)v\|_2 + \beta_2 \|(Hu)v^\perp\|_2. \tag{11}$$

Thus, we obtain in (11) the absolute values of the second order derivative of u in direction of v weighted by β_1 and in perpendicular direction weighted by β_2. The above considerations motivate the following definition for arbitrary $u \in L^1(\Omega)$:

$$\mathcal{R}_2(u) := \sup\Big\{ \int_\Omega (\mathrm{div}^2\, \varphi) u \; dx \mid \varphi \in \mathcal{C} \Big\}, \text{ with} \tag{12}$$

$$\mathcal{C} := \{ C_C^\infty(\Omega; \mathbb{R}^4), \forall x \in \Omega : \langle \varphi(x), w_1(x) \rangle^2 + \langle \varphi(x), w_2(x) \rangle^2 \le (\beta_1(x))^2, \tag{13}$$
$$\langle \varphi(x), w_3(x) \rangle^2 + \langle \varphi(x), w_4(x) \rangle^2 \le (\beta_2(x))^2 \},$$

Existence Theory

We now show the existence of a unique minimizer of (7), where $\mathcal{R}_1(u)$ and $\mathcal{R}_2(u)$ are given by (8) and (12), respectively.

Proposition 1. *Assume that for every $x \in \Omega$ the eigenvalues $\lambda_i(x)$ of $A(x)$ are uniformly bounded by $0 < c_1 \le \lambda_i(x) \le c_2 < \infty$. Moreover, assume that $\|v(x)\|_2 = 1$ and that $\beta_i(x), i = 1, 2$ are bounded by $0 < c_3 \le \beta_i(x) \le c_4 < \infty$. Then functional (7) attains a unique minimizer in $L^2(\Omega) \cap BV^2(\Omega)$.*

The proof of Prop. 1 utilizes the following two lemmas:

Lemma 1. *Under the assumptions of Prop. 1 we have*

$$c_1\,\mathrm{TV}(u) \le \mathcal{R}_1(u) \le c_2 TV(u), \qquad c_3\,\mathrm{TV}^2(u) \le \mathcal{R}_2(u) \le \sqrt{2}c_4 TV^2(u). \quad (14)$$

Proof. The first claim follows, since $c_1\|v\|_2 \le \sqrt{v^\top A v} \le c_2\|v\|_2$ for any $v \in \mathbb{R}^2$. To show the inequalities for $\mathcal{R}_2$, we note that $cTV^2(u) = \sup\{\int_\Omega (\mathrm{div}^2\,\varphi)u\;dx \mid \varphi \in \mathcal{C}(c)\}$ with $\mathcal{C}(c) := \{\varphi \in C_C^\infty(\Omega, \mathbb{R}^4) \mid \|\varphi(x)\|_2 \le c\}$. Since $\{w_i\}_i$ is an orthonormal basis of $\mathbb{R}^4$ and the set $\mathcal{C}$ in (12) includes $\mathcal{C}(c_3)$, the first inequality follows. Moreover, $\mathcal{C}$ is included in $\mathcal{C}(\sqrt{2}c_4)$, providing the second inequality. □

Lemma 2 (Weakly-∗-semi-continuity). *Let $u^k \in BV^2(\Omega)$ be weakly-∗-converging to u^*, i.e. $\|u^k - u^*\|_{L^1} \to 0$, $\|\partial_{x_i}u^k - \partial_{x_i}u^*\|_{L^1} \to 0$, $i = 1, 2$, and $\sup_k TV^2(u^k) < \infty$. Then, again under the assumptions of Prop. 1, we have*

$$\mathcal{R}_1(u^*) \le \liminf_{k\to+\infty} \mathcal{R}_1(u^k) \quad \text{and} \quad \mathcal{R}_2(u^*) \le \liminf_{k\to+\infty} \mathcal{R}_2(u^k). \quad (15)$$

Proof. **Semi-continuity of $\mathcal{R}_1(u^k)$:** note that $\mathcal{R}_1(u^k) \le c_2\|\nabla u^k\|_{L^1} < \infty$. Since $\|\nabla u^k - \nabla u^*\|_{L^1} \to 0$, there exists a subsequence ∇u^{k_l} converging pointwise almost everywhere to ∇u^*. From the continuity of the mapping $v \mapsto \left(v^\top A v\right)^{\frac{1}{2}}$ it follows that $\left((\nabla u^{k_l})^\top A \nabla u^{k_l}\right)^{\frac{1}{2}}(x) \to \left((\nabla u^*)^\top A \nabla u^*\right)^{\frac{1}{2}}(x)$ almost everywhere. Since any converging subsequence of ∇u^k converges to ∇u^* (Lebesgue thm.), we find $\liminf_{k\to+\infty}\left((\nabla u^k)^\top A \nabla u^k\right)^{\frac{1}{2}}(x) = \left((\nabla u^*)^\top A \nabla u^*\right)^{\frac{1}{2}}(x)$ almost everywhere. The claim then follows from Fatou's Lemma.

Semi-continuity of $\mathcal{R}_2(u^k)$: For $\varphi \in \mathcal{C}$ we have

$$\begin{aligned}
\int_\Omega (\mathrm{div}^2\,\varphi)u^*\;dx &= -\int_\Omega (\partial_x\varphi_1 + \partial_y\varphi_2)\partial_x u^* + (\partial_x\varphi_3 + \partial_y\varphi_4)\partial_y u^*\;dx && (16)\\
&= -\lim_{k\to+\infty}\int_\Omega (\partial_x\varphi_1 + \partial_y\varphi_2)\partial_x u^k + (\partial_x\varphi_3 + \partial_y\varphi_4)\partial_y u^k\;dx && (17)\\
&= \lim_{k\to+\infty}\int_\Omega (\mathrm{div}^2\,\varphi)u^k\;dx \le \liminf_{k\to+\infty}\mathcal{R}_2(u^k). && (18)
\end{aligned}$$

Thus

$$\mathcal{R}_2(u^*) = \sup\Big\{\int_\Omega (\mathrm{div}^2\,\varphi)u^*\;dx \mid \varphi \in \mathcal{C}\Big\} \le \liminf_{k\to+\infty}\mathcal{R}_2(u^k). \quad (19)$$

□

Proof (of Prop. 1).
Since $\mathcal{F}(u)$ is bounded from below, we have $F_{inf} := \inf_{u\in BV^2(\Omega)}\mathcal{F}(u) > -\infty$. We consider a minimizing sequence $\{u^k\}_k$, $\mathcal{F}(u^k) \to F_{inf}$. Due to Lemma 1, $\mathcal{F}(u)$ is finite on $BV^2(\Omega)$, thus $\sup_k \mathcal{F}(u^k) \le C < +\infty$ for some $C > 0$. We show that $\{u^k\}_k$ is bounded in $L^2(\Omega) \cap BV^2(\Omega)$ due to coercivity of $\mathcal{F}$: from $C \ge \mathcal{F}(u^k) \ge \frac{1}{2}\|u^k - f\|_{L^2}^2$ it follows that $\{u^k\}_k$ is bounded in $\|\cdot\|_{L^2}$ and, since

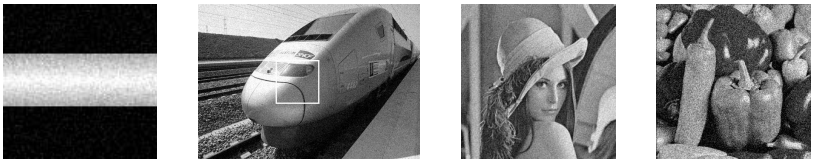


Fig. 4. Test images used in the comparison in Table 1

Ω is bounded, also in $\|\cdot\|_{L^1}$; $C \geq \mathcal{F}(u^k) \geq \mathcal{R}_l(u^k)$ and Lemma 1 provide that the minimizing sequence is bounded in $TV^l(\cdot)$, $l = 1, 2$. From boundedness follows by Theorem 9.83 in [16] that a weakly-$*$-converging subsequence in $L^2(\Omega) \cap BV^2(\Omega)$ with some limit u^* exists. We denote the subsequence also by $\{u^k\}_k$. We have $\frac{1}{2}\|u^k - f\|_{L^2}^2 \to \frac{1}{2}\|u^* - f\|_{L^2}^2$, and due to Lemma 2, $\mathcal{R}_l(u^*) \leq \liminf_{k\to+\infty} \mathcal{R}_l(u^k)$, $l = 1, 2$. Thus $\mathcal{F}(u^*) \leq \liminf_{k\to+\infty} \mathcal{F}(u^k) = F_{inf}$, i.e. u^* is a minimizer of $\mathcal{F}(u)$. Uniqueness follows from the strict convexity of $\mathcal{F}(u)$. □

Choice of Regularization Parameters
It remains to choose appropriate regularization parameters $\alpha_i(x), \beta_i(x)$, $i = 1, 2$, and directions $v(x)$. For the vector field $v(x)$ we choose $v^S(x)$ as defined in Sect. 2.1. Recall that $v^S(x)$ provides a smoothed version of the image gradient, which at edges coincides with the edge normals. To avoid a loss of contrast at edges and over-smoothing at slope discontinuities, we reduce α_i and β_i at edges and slope discontinuities using the indicator function $E(x) := \max(E^S(x), E^T(x))$ based on structure tensors S_f, and T_f applied to data f. We propose

$$\begin{aligned} \alpha_1(x) &:= E(x)\underline{\alpha} + (1 - E(x))\overline{\alpha}, & \alpha_2(x) &:= \overline{\alpha}, \\ \beta_1(x) &:= E(x)\underline{\beta} + (1 - E(x))\overline{\beta}, & \beta_2(x) &:= \overline{\beta}, \end{aligned} \tag{20}$$

with four free parameters $\underline{\alpha}, \overline{\alpha}, \underline{\beta}, \overline{\beta} > 0$ to be chosen appropriately. We propose a weak smoothing at edges and slope discontinuities with small $\underline{\alpha}, \underline{\beta}$. These parameters can be chosen fairly independent from the image content or noise level. Similar to other second-order TV approaches, it remains to choose two appropriate values for $\overline{\alpha}, \overline{\beta}$ depending mainly on the noise level of the image.

3.2 Remarks on Alternative Approaches

Regarding second-order approaches based on total generalized variation (TGV) [2] and infimal convolution (IC) [17], which both require two regularization parameters α, β, we observe that both approaches can be extended to be spatially adaptive by locally varying these parameters. We propose to choose

$$\alpha(x) := E(x)\underline{\alpha} + (1 - E(x))\overline{\alpha}, \qquad \beta(x) := E(x)\underline{\beta} + (1 - E(x))\overline{\beta}, \tag{21}$$

with suitable $\underline{\alpha}, \overline{\alpha}, \underline{\beta}, \overline{\beta}$ and $E : \Omega \to [0, 1]$ as defined in Sect. 3.1.

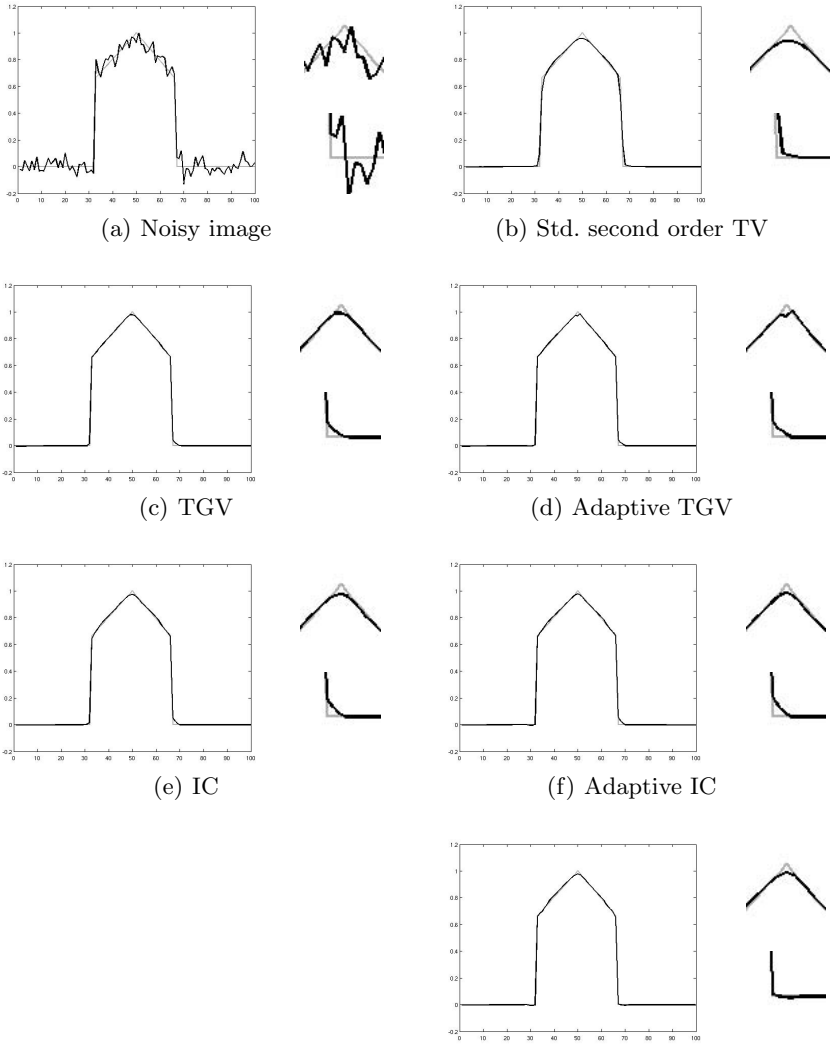
Table 1. Mean squared errors (MSE) to the noise-free image for the different methods. For each method, the approximate optimal parameters were retrieve by grid search. Independent of the model, introducing adaptivity always improves the error. The results of the proposed *anisotropic* method show the lowest reconstruction error.

Example	Roof	Train (part)	Lena	Peppers
2nd order TV with std. struct. tensor	4.6946e-4	2.4249e-4	1.0703e-4	1.8858e-4
TGV	0.8857e-4	2.3644e-4	0.9362e-4	1.3883e-4
Adaptive TGV	0.8703e-4	2.3364e-4	0.8985e-4	1.3258e-4
IC	1.0405e-4	2.3968e-4	0.9519e-4	1.3822e-4
Adaptive IC	0.9861e-4	2.3693e-4	0.9205e-4	1.3589e-4
Proposed method	**0.5703e-4**	**2.2560e-4**	**0.8749e-4**	**1.2997e-4**

4 Experiments

In this section we perform a quantitative comparison of the total generalized variation (TGV) approach, infimal convolution (IC), their adaptive counterparts as proposed in Sect. 3.2, and the proposed anisotropic second-order TV model (Sect. 3.1)[1]. For TGV and IC we use the original codes, which were kindly provided by the authors of [2,17]. In addition, we consider an anisotropic second-order TV, where the adaptivity is determined only by the standard structure tensor, i.e. $E(x) = E^S(x)$. As test images we use the image *roof*, cf. Fig. 2, left, a part of the *train* image from [2] and the *Lena* and *peppers* image, adding 5% zero mean Gaussian noise, cf. Fig. 4. For each image, the approximate optimal parameters for each method ($\underline{\alpha}$,β,$\overline{\alpha}$,$\overline{\beta}$) were determined via a grid search minimizing the mean squared error (MSE) to the noise-free image. While other error norms are also applicable, we have chosen the MSE as it is the most commonly used. Table 1 shows the errors for each method. We observe that by introducing *adaptivity*, we are able to decrease the error compared to the non-adaptive methods. The proposed method achieves the smallest error across all instances, showing the advantage of introducing *anisotropic* regularization. Moreover, it becomes clear that using solely the standard structure tensor to steer the anisotropy does not suffice, justifying our approach of also taking slope discontinuities into account. Figs. 5 and 6 depict the results of the methods on the *roof* and *train* images. For the latter, we observe that the results still contain some amount of the original noise. It seems that minimizing the MSE by grid search favors such residual noise rather than to strongly smooth the results. Since human users generally prefer a stronger smoothing, we provide in Fig. 7 results with manually adapted parameters for TGV and the proposed methods (due to space constraints, we omit IC here). The increased smoothing removes some image structures, as can be seen in the difference images. The proposed method preserves edges better than the competitive approaches.

[1] Computational speed: 17 sec for a MATLAB implementation on an 256x256 image using an Intel i7-2600K CPU 3.40GHz processor.



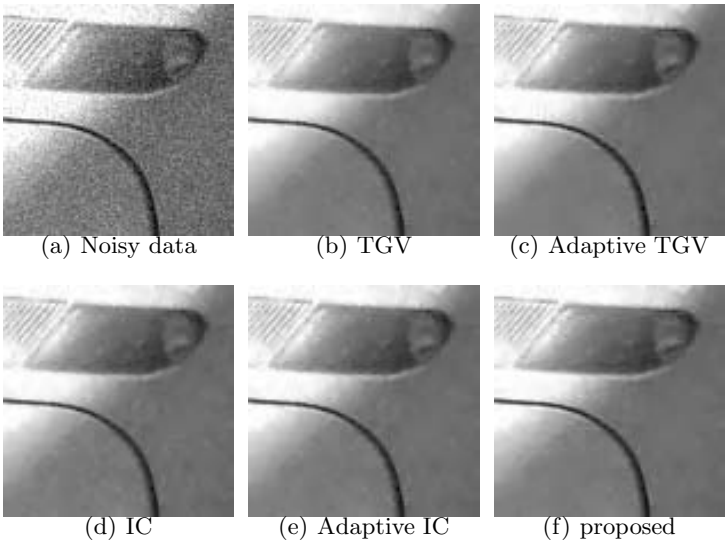
(a) Noisy image (b) Std. second order TV

(c) TGV (d) Adaptive TGV

(e) IC (f) Adaptive IC

(g) Proposed

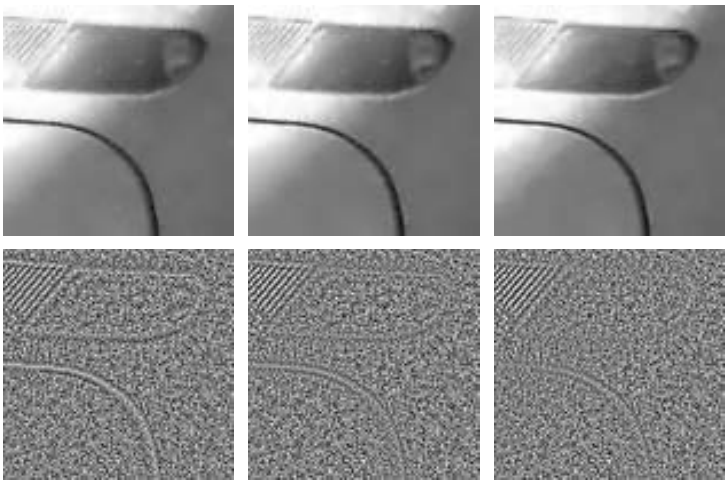
Fig. 5. Cross-section of the results (black lines) of TGV, IC, their adaptive variants and the proposed method on the *roof* image and detailed views of the peak and the left step. The noise-free data is shown in gray. We remark that standard second-order TV (b), cf. (6), significantly flattens the peak. All considered approaches avoid such a flattening to varying degrees. The TGV variants provide the sharpest reconstruction of the peak. The proposed approach provides a sharp reconstruction of both kinks.



(a) Noisy data (b) TGV (c) Adaptive TGV

(d) IC (e) Adaptive IC (f) proposed

Fig. 6. Results of the tested methods on the *train* image. For each method the parameters were selected by a grid search minimizing the mean squared error (MSE). As a consequence, all methods preserve some noise. Visually, the results are very similar.



(a) TGV (b) Adaptive TGV (c) proposed

Fig. 7. Denoising results using manually chosen parameters (top row) and difference image to noisy data (bottom row). TGV shows a strong smoothing effect, with the drawback, that also edges become smoother. Adaptive methods preserve edge structures better, as can be seen from the weaker edges in the difference images. In textured regions, all methods partly remove the texture. The proposed method shows the smallest amount of structures in the difference image.

5 Conclusion and Future Work

We proposed a way to modify state-of-the-art second-order TV models by introducing spatial adaptivity. Moreover, we introduced a new anisotropic second-order TV model. Experiments show that the modifications lead to an improved reconstruction performance. Since all considered methods exhibit over-smoothing in textured regions, future work will focus on how adaptive approaches can be improved by including texture information. New insight into this problem could also possibly close the conceptual gap to non-local regularization approaches.

Acknowledgements. We thank Tanja Teuber and Kristian Bredies for kindly providing their codes. The work of J.L. was supported by Award No. KUK-I1-007-43, made by King Abdullah University of Science and Technology (KAUST), EPSRC first grant EP/J009539/1, and EPSRC/Isaac Newton Trust Small Grant.

References

1. Berkels, B., Burger, M., Droske, M., Nemitz, O., Rumpf, M.: Cartoon extraction based on anisotropic image classification. In: Vision, Modeling, and Visualization Proceedings, pp. 293–300 (2006)
2. Bredies, K., Kunisch, K., Pock, T.: Total Generalized Variation. SIAM J. Imaging Sciences 3(3), 492–526 (2010)
3. Chen, Q., Montesinos, P., Sun, Q.S., Heng, P.A., Xia, D.S.: Adaptive total variation denoising based on difference curvature. Image Vision Comput. 28(3), 298–306 (2010)
4. Dong, Y., Hintermüller, M.: Multi-scale total variation with automated regularization parameter selection for color image restoration. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 271–281. Springer, Heidelberg (2009)
5. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: Proc. ISPRS Conf. on Fast Processing of Photogrammetric Data, pp. 281–305 (1987)
6. Frick, K., Marnitz, P., Munk, A.: Statistical multiresolution estimation for variational imaging: With an application in Poisson-biophotonics. J. Math. Imaging Vis, 1–18 (2012)
7. Grasmair, M.: Locally adaptive total variation regularization. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 331–342. Springer, Heidelberg (2009)
8. Grasmair, M., Lenzen, F.: Anisotropic Total Variation Filtering. Appl. Math. Optim. 62(3), 323–339 (2010)
9. Hu, Y., Jacob, M.: Higher degree total variation (HDTV) regularization for image recovery. IEEE Trans. Image Processing 21, 2559–2571 (2012)
10. Kindermann, S., Osher, S., Jones, P.W.: Deblurring and denoising of images by nonlocal functionals. Multiscale Model. Simul. 4(4), 1091–1115 (2005) (electronic)
11. Lefkimmiatis, S., Bourquard, A., Unser, M.: Hessian-based norm regularization for image restoration with biomedical applications. IEEE Transactions on Image Processing 21(3), 983–995 (2012)

12. Lenzen, F., Becker, F., Lellmann, J., Petra, S., Schnörr, C.: A class of quasi-variational inequalities for adaptive image denoising and decomposition. Comput. Optim. Appl. (2012) (online first)
13. Lysaker, M., Tai, X.-C.: Iterative image restoration combining total variation minimization and a second-order functional. Int. J. Comp. Vis. 66, 5–18 (2006)
14. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D 60(1-4), 259–268 (1992)
15. Scherzer, O.: Denoising with higher order derivatives of bounded variation and an application to parameter estimation. Computing 60, 1–27 (1998)
16. Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational Methods in Imaging. Springer (2009)
17. Setzer, S., Steidl, G., Teuber, T.: Infimal convolution regularizations with discrete l1-type functionals. Comm. Math. Sci. 9, 797–872 (2011)
18. Steidl, G., Teuber, T.: Anisotropic smoothing using double orientations. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 477–489. Springer, Heidelberg (2009)

Variational Methods for Motion Deblurring with Still Background

Eileen Laue and Dirk A. Lorenz

Institute for Analysis and Algebra, TU Braunschweig, 38092 Braunschweig, Germany
{e.laue,d.lorenz}@tu-braunschweig.de

Abstract. Motion deblurring problems are considered, however, as additional difficulty we consider that the motion occurs in front of a still background. First we propose a model for the formation of this kind of partly blurred images which involve four unknown quantities: The object, the background, the blur kernel and a mask that encodes the shape of the object. Then we propose variational methods to solve the deblurring problem. We show that the method performs well if three of the sought-after quantities are known. Finally we show that the method even works for real world examples as soon as the user makes a crude selection of the blurred region in the image.

1 Introduction

Deblurring is one of the most fundamental basic problems in image processing since blurring occurs naturally in many imaging systems. Blur may occur due to various reasons, e.g. wrong focus of the imaging system, failures of the imaging system or due to motion of the camera or the object (cf. [1, Chapter 5]). Here we are going to focus on the following case which has not received much attention: The image consists of blurred objects in front of a sharp background and the blur of the object is due to motion during exposure time.

(a) (b)

Fig. 1. Examples of motion blur in front of still background. (a) An artificially generated image. (b) A real-world example (Fort Washington Way@ Flickr, http://www.flickr.com/photos/27745117@N00/2666006951/).

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 74–85, 2013.

There is a huge literature about deblurring, see, e.g., [2] and the references therein. Typically, blur is modeled as a convolution of the image with a blur kernel h and hence, deblurring is also known as deconvolution. The instance in which the blur kernel h is not known, but which is still modeled as a convolution is known as blind deconvolution and we refer to [3] for a fairly recent introduction. For the model of motion deblurring in front of a still background we are only aware of two other works: The work [4] proposes a variational method and is based on two images from a motion blurred video. It utilizes optical flow techniques and assumes only linear movement of the object. The work [5] uses just a single image but estimates the motion blur kernel from parts of the image where the blurred object boundary is in front of a uniform background. In this work we tackle the more difficult problem where we only assume that *one* image is available. Moreover, we do not make special assumptions on the blurring process (e.g. it is not limited to a linear motion) and propose a variational method to approximate the blur kernel, the object and the background from just one single image.

2 Modeling

If a rigid object moves in front of a still background during exposure time, the object occupies different parts of the background at different times. At the points where the object occludes the background only for a fraction of the exposure time, the background "shines through" the object. We model this kind of transparency by a binary mask which moves along with the object. The whole image model is as follows: We consider a domain $\Omega \subset \mathbf{R}^2$ and a background image $u_B : \Omega \to \mathbf{R}$. The object image u_0 is also modeled as a function on Ω, however, its contours are modeled by a binary mask $\alpha : \Omega \to \{0, 1\}$ where $\alpha(x) = 1$ means, that x is an object pixel and $\alpha(x) = 0$ means that the pixel does not belong to the object. Moreover, the motion blur is given by $\gamma : [0, 1] \to \Omega$ (where we normalized the exposure time to 1), i.e., the object u_0 at time $t \in [0, 1]$ is $u_0(x - \gamma(t))$. The observed image u obtained from the object u_0 (with mask α), blurred by movement along γ in front of the still background u_B is

$$u(x) = \int_0^1 \alpha(x - \gamma(t))u_0(x - \gamma(t)) + (1 - \alpha(x - \gamma(t)))u_B(x)dt$$

In fact, we will model the movement γ by a convolution kernel h, rendering the image formation model as

$$u = (\alpha\, u_0) * h + (1 - \alpha * h)\, u_B. \tag{1}$$

Our goal is, to extract from a given image u the object image u_0 along with its mask α, the background image u_B and the blur kernel h. This problem is highly under-determined as there are several trivial solutions which are not of interest: for example we could set $u_0 = u$, $\alpha = 1$, $h = \delta$ and u_B arbitrary (i.e. we take the image u as object, and no movement) or similarly $u_B = u$, $\alpha = 0$ and h and

u_0 arbitrary (i.e. we take the image u as background). To overcome this underdetermination and also the ill-posedness we employ variational regularization in the following.

Note that the forward model is non-linear in the tuple of sought-after quantities (α, h, u_0, u_B), but linear in each unknown individually.

The variational approach we propose consists of a quadratic discrepancy term

$$\tfrac{1}{2}\|(\alpha\, u_0) * h - (\alpha * h)\, u_B + u_B - u\|^2 \tag{2}$$

which will be augmented by regularization terms for the sought-after variables. In the next section we treat each sought after quantity individually and motivate the variational models which we propose. The minimization of the respective functional can be done by standard methods (by the primal-dual method from [6] for the object u_0 and the mask α, by projected gradient descent [7] for the blur kernel h and directly for the background u_B). In Section 4 we tackle the problem of simultaneous reconstruction of all quantities by carefully choosing initializations and alternating minimization.

3 Separate Reconstruction

We start to tackle the problem of reconstructing α, h, u_0 and u_B by investigating how one can reconstruct a single one of these components when the others are assumed to be known. We illustrate the methods with an artificially generated image (cf. Figure 1 (a)). The background consists of a corner of a soccer field of size 305×600 pixels (Figure 2 (b)), the object is a football of size 75×75 pixels (Figure 2 (c)), the contour of the corresponding mask in depicted Figure 2 (a). The blur kernel represents a severe motion blur of approximately 60 pixels length, see Figure 2. As a matter of fact, it is an easy task to generate a crude initial guess for the mask α of the object by hand from the given image u (Figure 2 (a)) by just marking the blurred region. We are going to use such an initial guess in the following, which is depicted in Figure 2 (a).

3.1 Reconstruction of the Object

The object is assumed to be contained in the region specified by the mask α. In that region we impose regularity by a total variation penalty [8], i.e. we penalize u_0 with $\mu_0 \int_\Omega \alpha |\nabla u_0|$. Moreover, we explicitly enforce bound constraints on u_0 to avoid over and undershoots. This leads to the minimization problem

$$\min_{0 \le u_0 \le 1} \tfrac{1}{2}\|(\alpha\, u_0) * h - (\alpha * h)\, u_B + u_B - u\|^2 + \mu_0 \int_\Omega \alpha |\nabla u_0| \tag{3}$$

Since u_0 enters the first norm linearly, this is almost a standard TV-denoising problem [9] (apart from the bound constraints and the weighted TV-seminorm). It can be tackled, e.g., by the primal-dual approach described in [6] as follows:

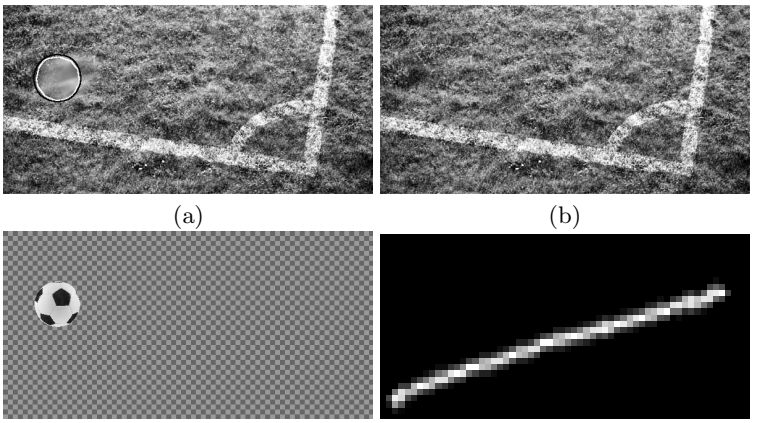


Fig. 2. Data for the artificial example. (a) Image u with blurred object, contour of the object (black) and contour of the user-generated crude guess of the mask (white). (b) Background u_B. (c) Object u_0. (d) Blur kernel h.

We define a linear operator A by $Au_0 = (\alpha\, u_0) * h$ and set $Ku_0 = \left[\nabla u_0,\ \ Au_0\right]^T$. With

$$F(v,w) = \mu_0 \int_\Omega \alpha|v| + \frac{1}{2}\int_\Omega |w - (\alpha * h)\, u_B + u_B - u|^2$$

$$G(u_0) = \begin{cases} 0 & 0 \le u_0 \le 1 \\ \infty & \text{else} \end{cases}$$

the problem (3) reads as $\min_{u_0} F(Ku_0) + G(u_0)$. To employ the primal-dual method we denote $F(v,w) = F_v(v) + F_w(w)$ with $F_v(v) = \mu_0 \int_\Omega \alpha|v|$ and $F_w(w) = \frac{1}{2}\int_\Omega |w - (\alpha * h)\, u_B + u_B - u|^2$ and calculate the proximal mappings for G and the conjugates ${F_v}^*(v^*) = 0$ (if $|v^*| \le \mu_0\alpha$), $= \infty$ (else) and ${F_w}^*(w^*) = \int_\Omega \frac{1}{2}(w^*)^2 + w^*((\alpha * h)\, u_B + u_B - u)$ as follows:

$$\mathrm{prox}_{\sigma G}(u_0) = (\mathrm{Id} + \sigma\partial G)^{-1}(u_0) = \min(\max(u_0, 0), 1)$$

$$\mathrm{prox}_{\tau {F_v}^*}(v) = (\mathrm{Id} + \tau\partial {F_v}^*)^{-1}(v) = \min(|v|, \alpha\mu_0)\frac{v}{|v|}$$

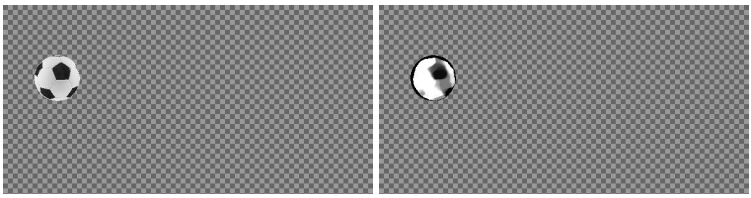
$$\mathrm{prox}_{\tau {F_w}^*}(w) = (\mathrm{Id} + \tau\partial {F_w}^*)^{-1}(w) = \frac{w - \tau((\alpha * h)\, u_B - u_B + u)}{1+\tau}$$

The primal-dual method with primal extra-gradient then reads as

$$\begin{aligned} v^{n+1} &= \mathrm{prox}_{\tau {F_v}^*}(v^n + \tau\nabla\overline{u_0}^n) \\ w^{n+1} &= \mathrm{prox}_{\tau {F_w}^*}(w^n + \tau A\overline{u_0}^n) \\ u_0^{n+1} &= \mathrm{prox}_{\sigma G}(u_0^n - \sigma(-\nabla\cdot v^{n+1} + A^* w^{n+1})) \\ \overline{u_0}^{n+1} &= 2u_0^{n+1} - u_0^n \end{aligned}$$

which converges under the condition that $\tau\sigma\|K\|^2 \le 1$.

If all quantities, except u_0, were known, one could set $\mu_0 = 0$ and minimize explicitly using the Fourier transform, see Figure 3 (a). However, if, e.g., the mask α is only known approximately, e.g. the user generated mask α from Figure 2 (a), one needs the regularization term to obtain meaningful results. In this case we also assumed the background image to be unknown and extracted it from the given image u by the approximate mask, i.e. we used $u_B = u\,(1-\alpha)$ (while the blur kernel h was known), see Figure 3 for the result.



(a) (b)

Fig. 3. Reconstruction of the object. (a) Object reconstructed from exactly known u_B, h and α by solving (3) with $\mu_0 = 0$. (b) Object reconstructed from a user-generated mask α and estimated background image $u_B = u\,(1-\alpha)$ with $\mu_0 = 10^{-2}$.

3.2 Reconstruction of the Background

The reconstruction of the background is probably the easiest part if we do not aim to reconstruct parts which are occluded by the moving object at all times (a problem which could, in principle, be tackled by inpainting methods which we do not address here). The quadratic discrepancy (2) can be minimized with respect to u_B explicitly and easily by

$$u_B = \frac{u - (\alpha\, u_o) * h}{1 - \alpha * h}$$

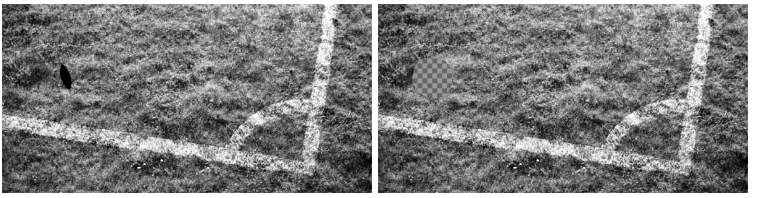
if the denominator is not zero. To ensure this, we add a quadratic regularization term $\frac{\lambda_B}{2}\|u_B\|_2^2$ with a small parameter $\lambda_B > 0$ and obtain

$$u_B = \frac{u - (\alpha\, u_0) * h}{1 - \alpha * h + \lambda_B}. \tag{4}$$

where positivity of the denominator is ensured since $\alpha \in \{0, 1\}$ and $h \geq 0$. The results can be seen in Figure 4 (a). The results do not depend crucially on the initial guess of the mask, as can be seen in Figure 4 (b).

3.3 Reconstruction of the Mask

The solution for the mask α is clearly of the type of a segmentation problem. To approximate the binary mask $\alpha : \Omega \to \{0, 1\}$ we propose to relax the problem



(a) (b)

Fig. 4. Reconstruction of the background. (a) Reconstructed background from exact u_0, h and α (note that the reconstruction is also accurate at the positions where the background has been mostly occluded). (b) Reconstructed background from exactly known u_0 and h but a user-generated mask (see Figure 2 (a)) with $\lambda_B = 10^{-8}$.

to a phase-field function $\phi : \Omega \to [0,1]$. In the spirit of [10] we propose the variational approach for the calculation of ϕ (given the other quantities) as

$$\min_{0\le\phi\le 1} \frac{1}{2}\int_\Omega |(\phi\, u_0) * h - (\phi * h)\, u_B + u_B - u|^2 + \lambda_\phi \int_\Omega |\nabla\phi| + \mu_\phi \int_\Omega \phi(1-\phi). \quad (5)$$

The regularization term $\lambda_\phi \int_\Omega |\nabla\phi|$ is a regularity constraint for the boundary of the mask (and in the case of a characteristic function ϕ it is the perimeter of the support). The second term $\mu_\phi \int_\Omega \phi(1-\phi)$ is a term which forces the values of ϕ towards the bounds 0 and 1. Note that this term is not convex but together with the bounds $0 \le \phi \le 1$ the existence of a minimizer is still ensured by standard arguments. The model bears similarities with phase field models which involve double-well potentials $\int \phi^2(1-\phi)^2$ (cf. [10]), however, in this form the minimizers are guaranteed to respect the bounds 0 and 1 and tend to be binary faster than for the double-well potential.

Although the problem is not convex, we can still tackle it by the primal-dual approach already described in Section 3.1: We define a linear operator A by $A\phi = (\phi\, u_0) * h - (\phi * h)\, u_B$ and set $K\phi = \begin{bmatrix}\nabla\phi, & A\phi\end{bmatrix}^T$. With

$$F(v,w) = \lambda_\phi \int_\Omega |v| + \frac{1}{2}\int_\Omega |w + u_B - u|^2, \qquad G(\phi) = \begin{cases} \mu_\phi \int_\Omega \phi(1-\phi) & 0 \le \phi \le 1 \\ \infty & \text{else} \end{cases}$$

the problem (5) reads as

$$\min_\phi F(K\phi) + G(\phi).$$

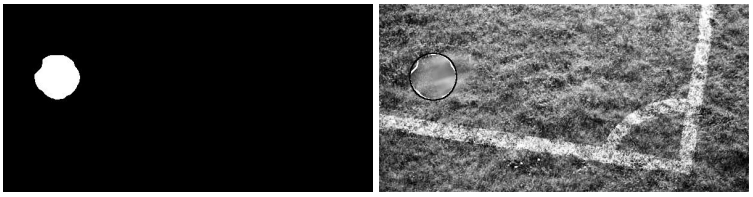
To employ the primal-dual method we denote $F(v,w) = F_v(v) + F_w(w)$ with $F_v(v) = \lambda_\phi \int_\Omega |v|$ and $F_w(w) = \frac{1}{2}\int_\Omega |w + u_B - u|^2$ and calculate the proximal mappings for G and the conjugates ${F_v}^*$ and ${F_w}^*$ (cf. Section 3.1) as follows:

$$\begin{aligned} \mathrm{prox}_{\sigma G}(\phi) &= (\mathrm{Id} + \sigma\partial G)^{-1}(\phi) = \max(\min(\tfrac{\phi - \mu_\phi\sigma}{1 - 2\mu_\phi\sigma}, 1), 0) \\ \mathrm{prox}_{\tau {F_v}^*}(v) &= (\mathrm{Id} + \tau\partial {F_v}^*)^{-1}(v) = \min(|v|, \lambda_\phi)\frac{v}{|v|} \\ \mathrm{prox}_{\tau {F_w}^*}(w) &= (\mathrm{Id} + \tau\partial {F_w}^*)^{-1}(w) = \frac{w + \tau(u_B - u)}{1 + \tau} \end{aligned}$$

The primal-dual method with primal extra-gradient then reads as

$$\begin{aligned} v^{n+1} &= \operatorname{prox}_{\tau F_{v^*}}(v^n + \tau \nabla \bar{\phi}^n) \\ w^{n+1} &= \operatorname{prox}_{\tau F_{w^*}}(w^n + \tau A \bar{\phi}^n) \\ \phi^{n+1} &= \operatorname{prox}_{\sigma G}(\phi^n - \sigma(-\nabla \cdot v^{n+1} + A^* w^{n+1})) \\ \bar{\phi}^{n+1} &= 2\phi^{n+1} - \phi^n \end{aligned}.$$

The convergence can not be ensured by the existing theory (due to non-convexity of the problem), however, for small σ (especially smaller than $(2\mu_\phi)^{-1}$) convergence is still observed in practice. This may be due to the fact that $\operatorname{prox}_{\sigma G}$ is Lipschitz continuous with constant $(1 - 2\mu_\phi\sigma)$ which is only slightly larger than one (for small σ or small μ_ϕ). Put differently: The non-convexity of the penalty may be so mild that it does not matter practically. The result of the minimization can be seen in Figure 5. Note that we did not use any prior knowledge on the mask or the location of the object as we initialized the mask with zero. However, the result closely resembles the original contour (and the obtained phase field function was in fact a binary function).



(a) (b)

Fig. 5. (a) Reconstruction of the mask by solving (5) with $\lambda_\phi = 0.02$ and $\mu = 0.01$. The mask has been initiated as $\phi^0 \equiv 0$. (b) Overlay of the reconstructed contour (white) and the original contour (black) over the given image.

3.4 Reconstruction of the Kernel

Our main assumption for the reconstruction of the blur kernel h is sparsity of h which suggests to use an ℓ^1 penalty. However, since a pure sparsity constraint may lead to overly sparse kernels, one may think of an elastic-net penalty [11], i.e. a penalty of the form $\lambda_h \|h\|_1 + \mu_h \|h\|_2^2$. But there are further assumptions on the convolution kernel which suggest to use a different approach: First the convolution kernel is assumed to be non-negative and second, we assume that it sums to 1 (i.e. the convolution with h does not change to total intensity). The first assumption is enforced directly by a bound constraint $h \geq 0$ and the second can be enforced by a constraint $\int h = 1$. Note that the last constraint does also enforce sparsity of the kernel, since it is a natural generalization of an ℓ^1-penalty $\|h\|_1 \leq 1$ to non-negative kernels. The combination of the non-negativity with the normalization of the one-norm has several advances: First, we can pass from a "one-norm-ball"-constraint to the direct normalization $\|h\|_1 = 1$ since,

in combination with the non-negativity, the resulting constraint set is convex. Second, the normalization of the one-norm of the kernel partly resolves the "scaling ambiguity" of bilinear problems like this one of "blind deconvolution type". Similar assumptions have been made previously however, in contrast to e.g. [12,13], where these constraints are imposed implicitly in the algorithm, we enforce the constraints directly in the minimization problem and the algorithm we propose is guaranteed to converge to a global minimizer of the constraint optimization problem.

To summarize, we approximate the kernel by solving

$$\min_{h\geq 0} \tfrac{1}{2}\|(\alpha\, u_0) * h - (\alpha * h)\, u_B + u_B - u\|^2 + \tfrac{\mu_h}{2}\|h\|_2^2 \quad \text{s.t.} \quad \int h = 1. \tag{6}$$

In fact, this problem can be reformulated as a standard quadratic program with linear constraints: With the operator

$$Ah = \begin{bmatrix} (\alpha\, u_0) * h - (\alpha * h)\, u_B \\ \sqrt{\mu_h} h \end{bmatrix}$$

the problem (6) reads as

$$\min_{h} \tfrac{1}{2}\|Ah - \begin{bmatrix} u - u_B \\ 0 \end{bmatrix}\|_2^2 \quad \text{s.t.} \quad \int h = 1,\ h \geq 0. \tag{7}$$

The constraints on h form (after discretization) the standard simplex on which one can efficiently project (cf. [14]). Hence, we solve for h by a projected gradient method with Barzilai-Borwein step-sizes (cf. [7]). For known data u_0, u_B and α, almost no regularization is needed, i.e. μ_h can be very small, see Figure 6 (b). In Figure 6 (c) we show the result if we do not assume exact knowledge of the other quantities but use the user-generated mask α (same as in Figure 3 (c)) and $u_B = u\,(1-\alpha)$ as estimate for the background image.

If there is an initial guess $\tilde{h}$ for the kernel available (which is reasonable for simple examples of motion blur), we could also use the penalty $\mu_h/2\|h-\tilde{h}\|_2^2$ which changes the data fit term in (7) into $\tfrac{1}{2}\|Ah - \begin{bmatrix} u - u_B \\ \sqrt{\mu_h}\tilde{h} \end{bmatrix}\|_2^2$.

4 Joint Reconstruction

In conclusion of the previous findings, we propose to minimize the following cost functional to solve the motion deblurring problem in front of a still background:

$$\begin{aligned} &\tfrac{1}{2}\|(\phi\, u_0) * h - (\phi * h)\, u_B + u_B - u\|^2 \\ &+ \mu_0 \int_\Omega \phi|\nabla u_0| + \frac{\lambda_B}{2}\|u_B\|_2^2 + \lambda_\phi \int_\Omega |\nabla\phi| + \mu_\phi \int_\Omega \phi(1-\phi) + \frac{\mu_h}{2}\|h\|_2^2 \\ &\text{s.t. } 0 \leq u_0 \leq 1,\ 0 \leq \phi \leq 1,\ h \geq 0,\ \int h = 1. \end{aligned} \tag{8}$$

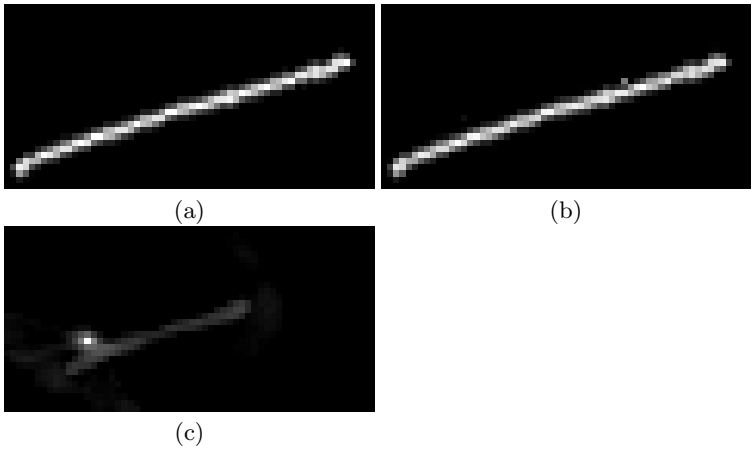


Fig. 6. Reconstruction of the kernel by solving (7). The kernel has been initiated as $h^0 = \delta$. (a) Original kernel. (b) Reconstructed kernel with exactly known data with $\lambda_h = 10^{-8}$and $\mu_h = 10^{-10}$. (c) Reconstructed kernel with user generated mask α and estimated background $u_B = u\,(1 - \alpha)$ with $\lambda_h = 5 \cdot 10^{-6}$and $\mu_h = 5 \cdot 10^{-6}$.

To reconstruct all quantities from just the given image u we propose alternating minimization with respect to the four involved variables u_0, u_B, h and α by the methods described in the previous sections. Due to numerous local minimizers which are not meaningful (cf. Section 2) careful initial guesses for at least three quantities are needed.

A widely used initial guess for the blur kernel h in the context of motion deblurring is the delta peak (i.e. no blurring is assumed) which is usually not very far from the true kernel. For the background image u_B we may take the given image u, which is in fact correct on a large part of the image domain. If we would now choose to initialize the object also by the given image, we would not get a meaningful estimate for the mask of the object, since any mask would lead to a global minimizer of the cost functional (8). Hence, we propose to use a user-generated initial guess for the mask α, i.e. the user extracts an approximate mask from the given image manually by selecting the blurred regions. A task which easily accomplished for images as the ones considered here. With this user generated mask α, we can improve the initial guess of the background u_B by using $u \cdot (1 - \alpha)$ (and similarly we could use $u \cdot \alpha$ as an initial guess for the object).

Once approximations for u_B, h and α are available we enter the following loop:

1. Solve for a new object u_0, as described in Section 3.1.
2. Solve for a new background u_B, as described in Section 3.2.
3. Solve for a new mask α, as described in Section 3.3.
4. Solve for a new blur kernel h, as described in Section 3.4, and iterate.

The outcome of just one and a half iteration of this loop (i.e. stopped after step 2 has been executed for the second time) performed on a real world example in shown in Figure 7. The size of the image is 239×365 pixel and the used parameters are $\mu_\phi = 0.1;$, $\lambda_\phi = 1$, $\lambda_B = 5 \cdot 10^{-4}$, $\mu_h = 10^3$, and $\mu_0 = 10^{-2}$. The method identifies a meaningful blur kernel and the sharpening of the object is satisfactory. On the downside, the reconstructed mask did not adapt properly to the object boundaries.

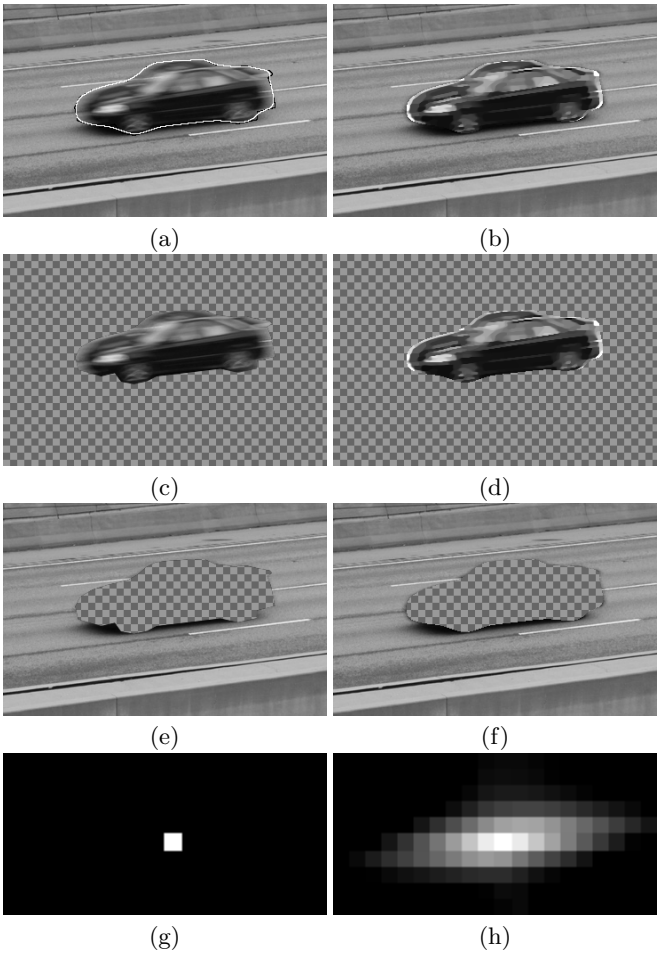


Fig. 7. Deblurring of a real world example. (a) Input image u. (b) Overlayed background and deblurred object. (c) Initial guess for the object. (d) Deblurred object u_0. (e) Initial guess for the background. (f) Reconstructed background u_B. (g) Initial guess for the blur kernel. (h) Reconstructed blur kernel h.

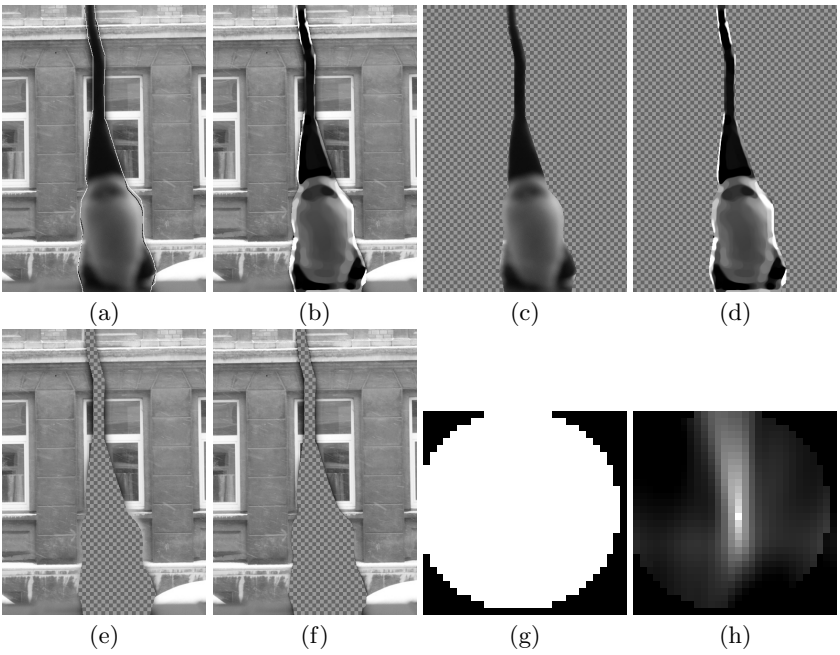


Fig. 8. Deblurring of a real world example. (a) Input image u. (b) Overlayed background and deblurred object. (c) Deblurred object u_0. (d) Reconstructed background u_B. (e) Reconstructed mask α. (f) Reconstructed blur kernel h.

Finally we note that the problem deblurring of a blurred object (by out of focus blur) in front a sharp background can be handled by exactly the same approach. A real world example of a small felt Santa Claus figurine in front of a house is shown in Figure 8. Again, a crude estimate of the mask was generated by hand and then four iterations of the loop has been performed. The image size is 600×427 pixels and the used parameters are $\mu_\phi = 5 \cdot 10^{-2}$, $\lambda_\phi = 10^{-1}$, $\lambda_B = 5 \cdot 10^{-8}$, $\mu_h = 10^5$, and $\mu_0 = 10^{-2}$. The blur kernel was initiated with a fairly out-of-focus kernel, however, the reconstructed kernel does not have a specific "out-of-focus" shape. The reconstructed object is again acceptable and again the reconstructed mask only differs slightly from the initial guess. Note, that the background image is been partly cleared from the occlusion of the blurred object (e.g. on the middle right boundary of the object). It should be noted that this example is fairly difficult due to the low contrast within the object which makes deblurring harder.

5 Conclusion

A variational method for deblurring of objects in front of a still background has been proposed. The proposed model for image formation was shown to be

accurate and a proof of concept for deblurring has been given. For the minimization, an algorithm for bound constrained and weighted TV-deblurring has been developed. Moreover, the primal-dual method for a slightly non-convex problem has been used and it was observed that it converged nicely (which gives rise to further studies of this subject). The projected gradient method with projection onto the standard simplex for the identification of the blur kernel worked particularly well (especially in the real world examples, rapid convergence has been observed). The overall results on real world examples are promising. In further work, the minimization algorithm could be tuned directly to the full objective functional, by directly incorporating the multilinear structure of the problem.

References

1. Chan, T.F., Shen, J.: Image Processing and Analysis - Variational, PDE, Wavelet, and Stochastic Methods. SIAM, Philadelphia (2005)
2. Hansen, P.C., Nagy, J.G., O'Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. Society for Industrial and Applied Mathematics, Philadelphia (2006)
3. Justen, L.A., Ramlau, R.: A non-iterative regularization approach to blind deconvolution. Inverse Problems 22, 771–800 (2006)
4. Bar, L., Berkels, B., Rumpf, M., Sapiro, G.: A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In: IEEE 11th International Conference on Computer Vision, pp. 1–8 (2007)
5. Jia, J.: Single image motion deblurring using transparency. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
6. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40, 120–145 (2011)
7. Figueiredo, M.A.T., Nowak, R.D., Wright, S.J.: Gradient projection for sparse reconstruction: Applications to compressed sensing and other inverse problems. IEEE Journal of Selected Topics in Signal Processing 4, 586–597 (2007)
8. Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. Numerische Mathematik 76, 167–188 (1997)
9. Rudin, L.I., Osher, S.J., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
10. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of denoising and segmentation models. SIAM Journal for Applied Mathematics 66, 1632–1648 (2006)
11. Jin, B., Lorenz, D.A., Schiffler, S.: Elastic-net regularization: error estimates and active set methods. Inverse Problems 25(11), 115022 (26pp) (2009)
12. Chan, T.F., Wong, C.: Total variation blind deconvolution. IEEE Transactions on Image Processing 7, 370–375 (1998)
13. You, Y.L., Kaveh, M.: A regularization approach to joint blur identification and image restoration. IEEE Trans. Im. Proc. 5(3), 416–428 (1996)
14. Chen, Y., Ye, X.: Projection onto a simplex (2011), `http://arxiv.org/abs/1101.6081`

Blind Deblurring Using a Simplified Sharpness Index

Arthur Leclaire and Lionel Moisan

Université Paris Descartes, France
MAP5, CNRS UMR 8145
{arthur.leclaire,lionel.moisan}@parisdescartes.fr

Abstract. It was shown recently that the phase of the Fourier Transform of an image could lead to interesting no-reference image quality measures. The Global Phase Coherence, and its recent Gaussian variant called Sharpness Index, rate the sharpness of an image in contrast not only with blur, but also noise, ringing, etc. In this work, we introduce a new variant of these indices, that can be computed with one Fourier Transform only, hence four times quicker than the Sharpness Index. We use this new index S to build an image restoration algorithm that, in a stochastic framework, selects a radial-unimodal deconvolution kernel for which the S-value of the restored image is optimal. Experiments are discussed, and comparison is made with a radial oracle deconvolution filter and the recent blind deconvolution algorithm of Levin et al.

Keywords: global phase coherence, sharpness, blind deconvolution, no-reference image quality assessment, oracle deconvolution filter.

1 Introduction

No-reference image quality assessment consists in designing algorithms to evaluate the quality of an image (in particular in relation with its level of blur and noise) without requiring either an ideal version of this image (full-reference) or features extracted from this ideal image (reduced-reference). Finding good image quality (and sharpness) metrics has several applications, like, e.g., parameter selection, image restoration [16], benchmarking, or depth estimation [3].

A way to address the notion of image quality is to think in terms of precision of its geometric elements (contours, alignments, etc.). Since the pioneering work of Oppenheim and Lim [14], it is well known that the geometry of an image is mainly encoded in the phase of its Fourier Transform. And yet, the phase information itself is still very difficult to understand. A first definition of local phase coherence was given in [13], [10] and used for edge detection. Later, it was used to design a local sharpness measure in [15] and [9]. In 2008, the authors of [1] defined a notion of Global Phase Coherence (GPC), which rates the sharpness of an image depending on how the regularity of the image is destroyed as its phase information is lost. Very recently in [2], a variant of GPC called Sharpness Index (SI) was introduced. It has the advantage of being described by an explicit

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 86–97, 2013.

closed-form formula, without needing computationally expensive Monte-Carlo Simulations like GPC.

In the present paper, we show that the SI metric can be further simplified to yield a new metric S that can be computed with only one Discrete Fourier Transform (versus four for the SI metric), while being an excellent approximation of the latter (Section 2). The behavior of this new metric is analyzed, in particular in comparison with the Q metric proposed by Zhu and Milanfar [16] (Section 3). Then, a blind deblurring algorithm is built in Section 4, that looks for the linear filter that maximizes the S-value of the restored image while imposing the Fourier Transform of the convolution kernel to be radial, unimodal and smooth. The results of this algorithm are discussed, and compared with the corresponding linear oracle and with the blind deconvolution algorithm recently proposed by Levin et al. [11].

2 Global Phase Coherence and Derived Sharpness Metrics

Let us first introduce some useful notations. In all the following, we consider gray-level images $u : \Omega \to \mathbb{R}$ defined on a discrete $M \times N$ rectangular domain

$$\Omega = \mathbb{Z}^2 \cap \left(\left[-\frac{M}{2}, \frac{M}{2} \right) \times \left[-\frac{N}{2}, \frac{N}{2} \right) \right).$$

The discrete Fourier transform (DFT) of u is the complex function $\hat{u}$ defined by

$$\forall \boldsymbol{\xi} \in \mathbb{Z}^2, \quad \hat{u}(\boldsymbol{\xi}) = \sum_{\mathbf{x} \in \Omega} u(\mathbf{x}) e^{-i\langle \boldsymbol{\xi}, \mathbf{x} \rangle} , \tag{1}$$

where $\langle \boldsymbol{\xi}, \mathbf{x} \rangle = 2\pi \left(\frac{x_1 \xi_1}{M} + \frac{x_2 \xi_2}{N} \right)$ with $\boldsymbol{\xi} = (\xi_1, \xi_2)$ and $\mathbf{x} = (x_1, x_2)$. The function $|\hat{u}|$ will be called the modulus of u. A phase function for u is any function $\varphi : \mathbb{Z}^2 \to \mathbb{R}$ such that for all $\boldsymbol{\xi} \in \mathbb{Z}^2$, one has $\hat{u}(\boldsymbol{\xi}) = |\hat{u}(\boldsymbol{\xi})| e^{i\varphi(\boldsymbol{\xi})}$.

The Ω-periodization of u is the image $\dot{u} : \mathbb{Z}^2 \to \mathbb{R}$ that extends u to $\mathbb{Z}^2$ by $\dot{u}(\mathbf{x}) = u(\mathbf{x}')$, where $\mathbf{x}'$ is the unique element of Ω such that $\mathbf{x}' - \mathbf{x} \in M\mathbb{Z} \times N\mathbb{Z}$. The gradient of $\dot{u}$ is defined by

$$\forall (x, y) \in \mathbb{Z}^2, \quad \nabla \dot{u}(x, y) = \begin{pmatrix} \partial_x \dot{u}(x, y) \\ \partial_y \dot{u}(x, y) \end{pmatrix} = \begin{pmatrix} \dot{u}(x+1, y) - \dot{u}(x, y) \\ \dot{u}(x, y+1) - \dot{u}(x, y) \end{pmatrix} , \tag{2}$$

and the (periodic and anisotropic) Total Variation of u is

$$\mathrm{TV}(u) = \|\partial_x \dot{u}\|_1 + \|\partial_y \dot{u}\|_1 = \sum_{\mathbf{x} \in \Omega} |\partial_x \dot{u}(\mathbf{x})| + |\partial_y \dot{u}(\mathbf{x})| . \tag{3}$$

The autocorrelation of $\nabla \dot{u}$ is the function $\Gamma : \Omega \to \mathbb{R}^{2\times 2}$ defined by

$$\Gamma(\mathbf{z}) = \begin{pmatrix} \Gamma_{xx}(\mathbf{z}) & \Gamma_{xy}(\mathbf{z}) \\ \Gamma_{xy}(\mathbf{z}) & \Gamma_{yy}(\mathbf{z}) \end{pmatrix} = \sum_{\mathbf{y} \in \Omega} \left(\nabla \dot{u}(\mathbf{y}) \right) \left(\nabla \dot{u}(\mathbf{y} + \mathbf{z}) \right)^T . \tag{4}$$

2.1 Global Phase Coherence

As mentioned earlier, the phase of an image u encodes a great part of the geometry of u: if one reproduces the famous experiment of [14] consisting in imposing the phase of an image u to another image v, one can see on the result that several edges from u have appeared and all the geometric content of v has disappeared. Indeed, phase coefficients need strong alignment constraints in order to produce sharp edges and clean flat regions in an image.

The Global Phase Coherence metric introduced in [1] quantifies how the loss of this phase coherence affects the image regularity, measured by its Total Variation (3). More precisely, the phase of an image u is randomized to produce the Random Phase Noise image U_ψ defined in Fourier Domain by

$$\forall \boldsymbol{\xi} \in \Omega, \quad \widehat{U_\psi}(\boldsymbol{\xi}) = |\hat{u}(\boldsymbol{\xi})| e^{i\psi(\boldsymbol{\xi})} , \tag{5}$$

where $\psi : \Omega \to \mathbb{R}$ is a uniform random phase (the coefficients of ψ are independent and uniformly distributed in $(0, 2\pi)$, modulo the relation $\psi(-\boldsymbol{\xi}) = -\psi(\boldsymbol{\xi})$ ensuring that U_ψ is real-valued, see [8]), which leads to

Definition 1 (Blanchet, Moisan, Rougé, 2008 [1]). *The Global Phase Coherence (GPC) of u is the number*

$$\mathrm{GPC}(u) = -\log_{10} \mathbb{P}(\mathrm{TV}(U_\psi) \leq \mathrm{TV}(u)) . \tag{6}$$

For an image u with sharp edges and clean uniform zones, the Total Variation is expected to be low amongst the ones of its phase randomizations. Therefore, for such an image, the probability of the event $\{\mathrm{TV}(U_\psi) \leq \mathrm{TV}(u)\}$ will be very small, and the value of $\mathrm{GPC}(u)$ will be large. That is why this phase coherence index (and the variants that follow) is expected to behave like an image quality measure. Note that without the logarithm in (6), the values of $\mathrm{GPC}(u)$ would often cause a numerical underflow (a value like, e.g., 10^{-1000} cannot be represented in most computer environments).

The main issue with (6) is that no closed-form formula has been found so far to compute $\mathrm{GPC}(u)$, so that a computationally expensive Monte-Carlo simulation (coupled with a Gaussian approximation of the random variable $\mathrm{TV}(U_\psi)$) is proposed in [1], which limits the potential application of the GPC metric.

2.2 Sharpness Index

Hopefully, a closed-form variant of GPC was recently found. In [2], the periodic convolution of u with a conveniently normalized Gaussian white noise W is considered instead of U_ψ, and the first two moments of $\mathrm{TV}(u * W)$ are explicitly computed in function of u. Note that $u * W$ is nothing but the natural Gaussian approximation of U_Ψ, and these two random images only differ by a convolution with the texton of a white noise, which is close to a Dirac distribution [6] (in Fourier Domain, $\widehat{u * W}$ and $\widehat{U_\Psi}$ differ by a multiplicative Rayleigh Noise). Even if the exact law of $\mathrm{TV}(u * W)$ seems difficult to compute, it is expected to be approximately Gaussian, which leads to

Definition 2 (Blanchet, Moisan, 2012 [2]). *The Sharpness Index of u is*

$$\mathrm{SI}(u) = -\log_{10} \Phi\left(\frac{\mu - \mathrm{TV}(u)}{\sigma}\right) \tag{7}$$

$$\text{where} \quad \Phi(t) = \frac{1}{\sqrt{2\pi}} \int_t^{+\infty} \exp\left(-\frac{x^2}{2}\right) dx \;, \quad \mu = (\alpha_x + \alpha_y)\sqrt{\frac{2}{\pi}}\sqrt{MN} \;, \tag{8}$$

$$\sigma^2 = \frac{2}{\pi} \sum_{\mathbf{z} \in \Omega} \alpha_x^2 \cdot \omega\left(\frac{\Gamma_{xx}(\mathbf{z})}{\alpha_x^2}\right) + 2\alpha_x\alpha_y \cdot \omega\left(\frac{\Gamma_{xy}(\mathbf{z})}{\alpha_x\alpha_y}\right) + \alpha_y^2 \cdot \omega\left(\frac{\Gamma_{yy}(\mathbf{z})}{\alpha_y^2}\right) \;, \tag{9}$$

$\alpha_x = \|\partial_x \dot{u}\|_2 = \left(\sum_{\mathbf{x} \in \Omega} |\partial_x \dot{u}(\mathbf{x})|^2\right)^{\frac{1}{2}}$, $\alpha_y = \|\partial_y \dot{u}\|_2$, Γ is the autocorrelation of $\nabla \dot{u}$ given in (4)*, and ω is the function defined by*

$$\forall t \in [-1, 1], \quad \omega(t) = t \cdot \mathrm{Arcsin}(t) + \sqrt{1 - t^2} - 1 \;. \tag{10}$$

In practice, the numerical computation of $\mathrm{SI}(u)$ requires the computation of $TV(u)$, α_x and α_y (linear time), plus the three different components of Γ that can be computed with four Fast Fourier Transforms (a direct FFT of u, and 3 inverse FFTs for the cross correlations of $\partial_x \dot{u}$ and $\partial_y \dot{u}$). The overall cost is hence dominated by these four $M \times N$ FFT computations, which represents a complexity of $\mathcal{O}(MN \log(MN))$ for well-suited image dimensions. Note that the function Φ is available in most mathematical libraries through the complementary error function (often written `erfc`), but when t is greater than say, 20, the following (almost exact) approximation is systematically used to avoid numerical underflow:

$$-\log_{10} \Phi(t) \simeq \frac{t^2 + \log(2\pi t^2)}{2\log(10)} \;. \tag{11}$$

2.3 A Simplified Version of SI

We now introduce a new index S which is analytically close to SI and faster to compute. For that, let us observe that $\omega(t)$ is equivalent to $t^2/2$ when $t \to 0$. Therefore, this approximation can be used to replace $\sigma^2 = \mathbb{V}\mathrm{ar}(\mathrm{TV}(u * W))$ by

$$\sigma_a^2 = \frac{1}{\pi} \sum_{\mathbf{z} \in \Omega} \alpha_x^2 \cdot \left(\frac{\Gamma_{xx}(\mathbf{z})}{\alpha_x^2}\right)^2 + 2\alpha_x\alpha_y \cdot \left(\frac{\Gamma_{xy}(\mathbf{z})}{\alpha_x\alpha_y}\right)^2 + \alpha_y^2 \cdot \left(\frac{\Gamma_{yy}(\mathbf{z})}{\alpha_y^2}\right)^2 \;,$$

which after simplification yields the following

Definition 3. *The S-metric of an image u is*

$$S(u) = -\log_{10} \Phi\left(\frac{\mu - \mathrm{TV}(u)}{\sigma_a}\right) \;, \tag{12}$$

$$\text{where} \qquad \sigma_a^2 = \frac{1}{\pi}\left(\frac{\|\Gamma_{xx}\|_2^2}{\alpha_x^2} + 2 \cdot \frac{\|\Gamma_{xy}\|_2^2}{\alpha_x\alpha_y} + \frac{\|\Gamma_{yy}\|_2^2}{\alpha_y^2}\right) \;, \tag{13}$$

and α_x, α_y, μ, and Γ are as in Definition 2.

Whereas SI needed all the coefficients of the gradient auto-correlation matrix, the S metric only depends on the overall energy of the three components Γ_{xx}, Γ_{xy} and Γ_{yy}. Thanks to Parseval's formula, they can be computed in Fourier domain, recalling that

$$\widehat{\Gamma_{xx}}(\boldsymbol{\xi}) = |\widehat{\partial_x u}(\boldsymbol{\xi})|^2 = 2\sin^2\left(\frac{\pi\xi_1}{M}\right)|\hat{u}(\boldsymbol{\xi})|^2 \;, \quad \widehat{\Gamma_{yy}}(\boldsymbol{\xi}) = 2\sin^2\left(\frac{\pi\xi_2}{N}\right)|\hat{u}(\boldsymbol{\xi})|^2 \;,$$

$$\text{and} \quad |\widehat{\Gamma_{xy}}(\boldsymbol{\xi})| = |\widehat{\partial_x u}(\boldsymbol{\xi})||\widehat{\partial_y u}(\boldsymbol{\xi})| = 2\left|\sin\left(\frac{\pi\xi_1}{M}\right)\sin\left(\frac{\pi\xi_2}{N}\right)\right||\hat{u}(\boldsymbol{\xi})|^2 \;.$$

The computation of S only involves the l^1 and l^2 norms of the gradient, and the FFT of u. Thus, the overall dominant cost is only one FFT (compared to 4 for SI), while the approximation of SI by S is very good as stated by

Proposition 1. *We have* $0 \leq \dfrac{\sigma^2 - \sigma_a^2}{\sigma_a^2} \leq \pi - 3 \approx 0.142$.

Proof. With the expressions of σ^2 and σ_a^2, one can write

$$\begin{aligned}\sigma^2 - \sigma_a^2 =& \frac{2}{\pi}\sum_{\mathbf{x}\in\Omega}\alpha_x^2\left[\omega\left(\frac{\Gamma_{xx}(\mathbf{x})}{\alpha_x^2}\right) - \frac{1}{2}\left(\frac{\Gamma_{xx}(\mathbf{x})}{\alpha_x^2}\right)^2\right] \\ &+2\alpha_x\alpha_y\left[\omega\left(\frac{\Gamma_{xy}(\mathbf{x})}{\alpha_x\alpha_y}\right) - \frac{1}{2}\left(\frac{\Gamma_{xy}(\mathbf{x})}{\alpha_x\alpha_y}\right)^2\right] + \alpha_y^2\left[\omega\left(\frac{\Gamma_{yy}(\mathbf{x})}{\alpha_y^2}\right) - \frac{1}{2}\left(\frac{\Gamma_{yy}(\mathbf{x})}{\alpha_y^2}\right)^2\right] \;.\end{aligned}$$

Besides, Taylor Formula applied to ω yields

$$\forall t \in [-1,1], \quad 0 \leq \omega(t) - \frac{1}{2}t^2 \leq ct^4 \leq ct^2 \;, \tag{14}$$

with $c = \omega(1) - 1 = \frac{\pi-3}{2}$, and thus

$$0 \leq \sigma^2 - \sigma_a^2 \leq \frac{2c}{\pi}\sum_{\mathbf{x}\in\Omega}\alpha_x^2\left(\frac{\Gamma_{xx}(\mathbf{x})}{\alpha_x^2}\right)^2 + 2\alpha_x\alpha_y\left(\frac{\Gamma_{xy}(\mathbf{x})}{\alpha_x\alpha_y}\right)^2 + \alpha_y^2\left(\frac{\Gamma_{yy}(\mathbf{x})}{\alpha_y^2}\right)^2 = 2c\sigma_a^2 \;.$$

3 Validation of S as a Quality Measure

As in [1], we shall systematically apply two simple image transforms to an image u before computing $S(u)$ with (12), in order to avoid periodization and quantization biases. First, as the S metric is defined (like GPC and SI) through a periodic setting, the periodic component of u (see [12]) is first extracted to avoid discontinuities across the image frame border. Then, a simple dequantization procedure (a $(1/2, 1/2)$ sub-pixel translation with Fourier interpolation, see [5]) is applied to ensure that the quantization of the original image (generally with 256 gray levels) does not artificially decreases its Total Variation.

On Fig. 1, we give a first empirical evidence that S behaves as an image quality measure. Indeed, this blur-noise diagram shows that the value of S decreases as

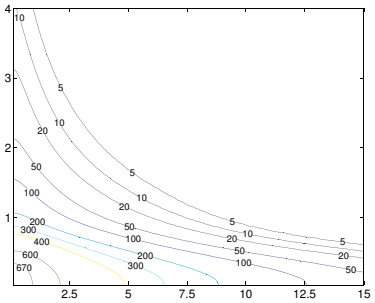


Fig. 1. Some level lines of the function $(r, \beta) \mapsto S(g_r * u + \beta W)$ where g_r is the 2-D Gaussian convolution kernel with standard deviation r, and W is a white noise image with unit variance in each pixel. The absolute values of S and the exact shape of the level lines depend on the image considered (here, Barbara), but the overall shape remains similar.

the level of blur or noise increases, with a correspondence between noise and blur which is similar to GPC and SI (see [1,2]).

The S metric not only decreases with respect to blur and noise, but unlike the Q metric of Zhu and Milanfar [16], it also decreases when ringing artifacts (that may result from excessive deblurring) appear, as shown on Fig. 2. This suggests that S could be used in a parametric or even non-parametric blind deblurring algorithm, as will be done in Section 4. Note however, that the Q metric performs slightly better than S for parameter selection in the SKR denoising method presented in [16]. In fact, the reason is logical considering the origin of S: the noise left by SKR in uniform zones is structured and the coherence of its phase makes S prefer less denoised images than Q.

4 Application to Blind Deblurring

Removing blur from a single image is a difficult task. If the blur is linear and spatially uniform, it can be modeled as a convolution. Several algorithms (see, e.g., the recent efficient scheme for $TV - L^2$ deblurring proposed in [4]), have been proposed to invert the effect of this convolution when the blur kernel is known. Addressing the problem of blind deconvolution, i.e. when the kernel is not known, is even more difficult, and several solutions have been proposed in the last decades. In the present paper, we shall consider in particular (for comparison purposes) the very recent work of Levin et al. [11].

Here, rather than trying to reverse the effect of a convolution, we shall try to improve the image directly by convolving the blurry image u with a unit-mass (that is, average-preserving) filter k that maximizes $F_u(k) = S(k * u)$. Indeed, Fig. 2 shows that, in a parametric Wiener deconvolution, the S metric is able to select the blur parameter. In [3], Calderero and Moreno made a similar observation for the SI metric in a context of reverse diffusion. In this section, we will show that S can be used for non-parametric blind deblurring.

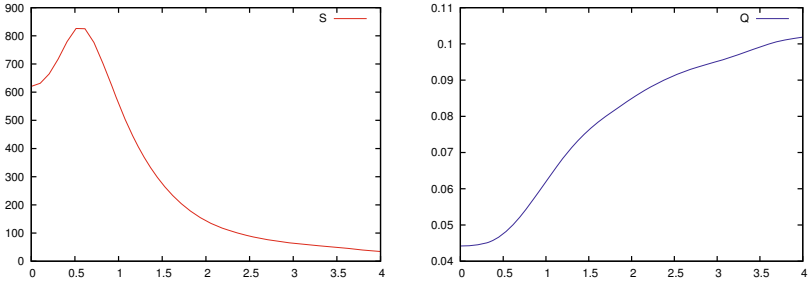


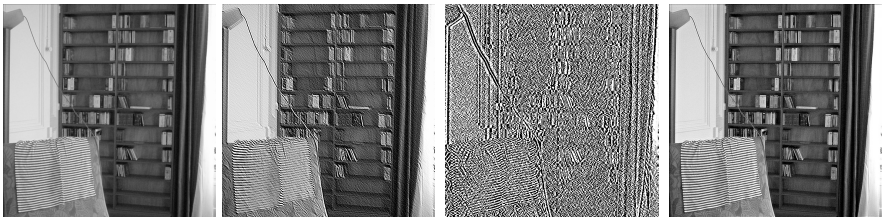
Fig. 2. These diagrams plot the proposed metric S (left) and the metric Q (right) of Zhu and Milanfar [16] obtained when applying (in Fourier domain) a H^1-regularized deconvolution filter to a natural image (Yale). The abscissa (ρ) corresponds to the standard deviation of the supposedly Gaussian convolution kernel. Over the results obtained for the whole range of ρ, one can see that contrary to Q, S attains a maximum for $\rho = 0.55$, which roughly corresponds to the value beyond which ringing artifacts begin to appear. This suggests that S is able to discriminate ringing, whereas Q does not.

4.1 Kernels with Compact Support

Algorithm 1 below can be used to optimize F_u on particular sets of kernels k, as soon as the number of coefficients that define k remains small enough. For example, one can optimize F_u on the set of symmetric 5×5 kernels (12 coefficients), or on the set of separable symmetric 21×21 kernels (20 coefficients, or 10 if the same one-dimensional kernel is used for each coordinate). In general, the results obtained with Algorithm 1 are good, but some images lead to interesting failure cases, in particular when regions with highly structured textures or dominant orientations are present (see Fig. 3). Since the functional F_u is not concave, it does not necessarily have a unique local maximum, and a reason could be that Algorithm 1 does not manage to converge to the actual global maximum of the objective function F_u. However, experiments suggests that the failure is more likely to be due to an inadequate set of kernels (in particular the set of separable kernels). To avoid such degenerated cases, and get rid of the small-kernel-support constraint, we consider in Section 4.2 other sets of kernels for which constraints are considered in Fourier domain.

Algorithm 1

- Begin with $k = \delta_0$ (discrete Dirac kernel)
- Repeat n times
  - ▷ Define k' from a random perturbation of k
  - ▷ If $S(k' * u) > S(k * u)$ then $k \leftarrow k'$
- Renormalize k to a unit-mass kernel
- Return k and $k * u$



original symmetric symmetric separable radial unimodal

Fig. 3. Blind deblurring of the original *Room* image (left). Algorithm 1 is applied for two different sets of kernels: 5×5 symmetric kernels (second column) and 21×21 symmetric separable kernels (third column). The right image is the result obtained with the method proposed in Section 4.2 (Algorithm 2, $\mu = 0$). We observe that Algorithm 1 fails in both cases (probably in reason of the large striped texture), while the radial-unimodal constraint imposed in Algorithm 2 yields a nice-looking result.

4.2 Optimization of a Radial-Unimodal Kernel

Instead of imposing that the convolution kernel has a fixed compact support, we here consider the set of kernels that are radial in Fourier domain, with a unimodal profile, which is a plausible assumption for a deconvolution kernel. More precisely, we assume that the DFT of the restoration kernel k_r is given by

$$\forall \boldsymbol{\xi} \in \Omega, \quad \widehat{k_r}(\xi_1, \xi_2) = L_r \left(\sqrt{2(d-1) \left(\left(\frac{\xi_1}{M} \right)^2 + \left(\frac{\xi_2}{N} \right)^2 \right)} \right) , \tag{15}$$

where $L_r : [0, d-1] \to \mathbb{R}$ is the piecewise affine interpolate on $[0, d-1]$ of the finite sequence $r(0) = 1, r(1), r(2), \ldots, r(d-2), r(d-1) = 0$. This sequence is supposed to be unimodal, which means that there exists a value i_m (mode index) that satisfies

$$\forall i < i_m, \quad r(i+1) \geq r(i) , \quad \text{and} \quad \forall i \geq i_m, \quad r(i+1) \leq r(i) .$$

One possible perturbation strategy for this set of kernels consists in the addition of a uniform random value to a randomly chosen coefficient of r, followed by a projection on the set U of unimodal sequences (this projection can be computed in $O(n^2)$ operations with the Pool Adjacent Violators algorithm [7]). We observed that with this strategy, moving the mode position was difficult, so we relaxed the unimodality constraint and incorporated in the objective function the l^2-distance $d(r, U)$ between r and the set U of unimodal sequences. We also found useful to add the possibility to increase the regularity of the radial profile r by incorporating a term depending on

$$\|r\|_{H^1}^2 = \sum_{i=0}^{d-2} (r(i+1) - r(i))^2 . \tag{16}$$

Finally, the objective function (to be maximized) is

$$\mathcal{F}_u(r) = S(k_r * u) - \lambda \, d(r, U) - \mu \, \|r\|_{H^1} , \tag{17}$$

where λ and μ are weighting parameters.

In order to maximize $\mathcal{F}_u$, we used Algorithm 2 below. We observed that $n = 10000$ was sufficient to ensure convergence on r (the relative changes after 10000 iterations were less than 10^{-3}); moreover, we checked that several realizations of this stochastic technique led to the same local maximum. The other parameters were set to $d = 20$, initial $i_m = 5$, $a = 0.1$, $\lambda = 10000$, and μ varying from 0 to 100. Let us comment the choice of i_m. As soon as several local maxima are present, the result of an optimization technique may depend on the initialization, and the natural solution would be to apply the algorithm for all possible values of the initial mode index. But in a wide majority of cases, we observed that the result of this algorithm was not depending on this initial value. In a few cases, two different local maxima could be found, but the higher value of the objective function $\mathcal{F}_u$ was always obtained for $i_m \in [d/4, 3d/4]$. This is why it seems empirically sufficient to run the algorithm only once with the initial value of i_m in that range.

Algorithm 2

- Initialize r with the piecewise-linear profile such that $r(0) = 1, r(i_m) = 2$, and $r(d-1) = 0$.
- Repeat n times
  - ▷ Pick a random index $i \in [1, d-2]$
  - ▷ Draw a random value $\varepsilon \in [-a/2, a/2]$ with uniform distribution
  - ▷ Set $r' \leftarrow r$, then $r'(i) \leftarrow r(i) + \varepsilon$
  - ▷ If $\mathcal{F}_u(r') > \mathcal{F}_u(r)$ then $r \leftarrow r'$
- Return r, k_r and $k_r * u$

4.3 Results

We first used Algorithm 2 with $\mu = 0$ on *Room* image, and checked that the failure of Algorithm 1 was avoided (Fig. 3, right).

Then, to produce the results shown in Fig. 4, we took two classical images (*Capitol* and *Parrots*) and corrupted them with a Gaussian blur kernel (width 1 pixel) and an additive white Gaussian noise (variance 1). We then applied several deblurring algorithms (detailed below) and evaluated their performances by computing their respective PSNR values with respect to the original clean image. Notice, however, that for blind deblurring tasks the PSNR value is not very reliable (in particular because even the original clean image is necessarily, in some sense, blurry and noisy), and visual inspection is often preferable to compare the algorithms.

First, we used Algorithm 2 with $\mu = 0$ and $\mu = 10$. We observed that the results were stable, and that the restoration resulted in a significant sharpness increase. However, for $\mu = 0$ some low-frequency noise is still visible on uniform zones. Increasing the value of μ to $\mu = 10$ reduces the residual noise (because it reduces the amplitude of L_r, that is, the amplification of noisy Fourier coefficients), but also attenuates some details in textured zones.

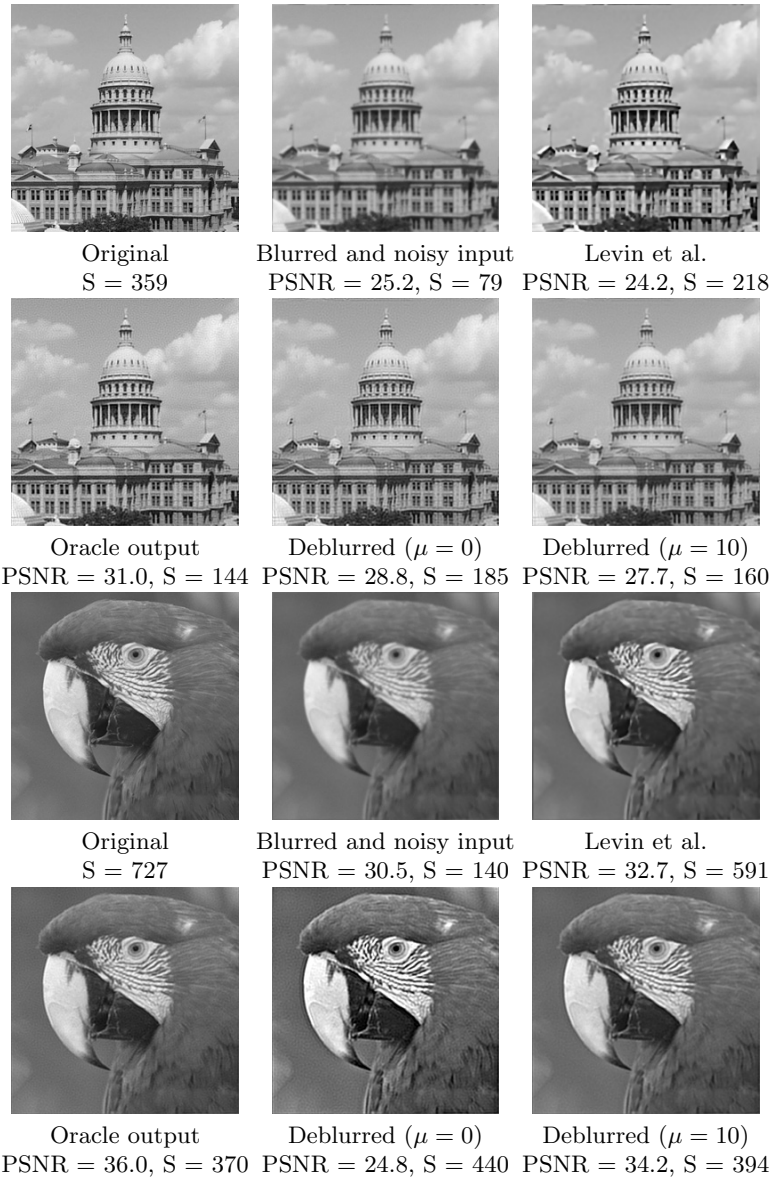


Fig. 4. Blind deblurring of a degraded version of *Capitol* and *Parrots* images (Gaussian blur of width 1 pixel plus Gaussian noise with variance 1). We present in each case the original image, the blurred and noisy input, the result of Levin et al. algorithm [11], the oracle output (best possible result obtained by a radial unimodal convolution filter) and the results of Algorithm 2 with $\mu = 0$ and $\mu = 10$. The PSNR values are computed in each case with respect to the original image.

We then compared these results with the state-of-the-art blind deconvolution algorithm of Levin et al. [11]. One can see in Fig. 4 that Algorithm 2 is more precise on the fine details of the image, but it also keeps much more noise than this method. In fact, the result of [11] is really "clean" and has a small Total Variation, which explains incidentally why its S value is significantly larger compared to Algorithm 2. Notice also that the method [11] is more general, and has been shown to perform particularly well in the case of a motion blur, while the radial constraint of Algorithm 2 cannot handle such motion blurs.

Another interesting experiment consists in computing the optimal kernel k_o that maximizes the expected distance between the reconstructed image and the clean image u_0, knowing the parameters (kernel κ and noise level β) of the degradation process (this is an oracle since neither these parameters nor the clean image are supposed to be known). One has

$$k_o = \underset{k}{\mathrm{Arg\,min}}\ \mathbb{E}\|u_0 - k * (\kappa * u_0 + \beta W)\|^2 \tag{18}$$

and if all kernels k were considered, the solution would be given by

$$\forall \boldsymbol{\xi} \in \Omega, \quad \widehat{k_o}(\boldsymbol{\xi}) = \frac{\overline{\hat{\kappa}(\boldsymbol{\xi})}\ |\hat{u_0}(\boldsymbol{\xi})|^2}{|\hat{\kappa}(\boldsymbol{\xi})|^2 |\hat{u_0}(\boldsymbol{\xi})|^2 + \sigma^2 MN}\ . \tag{19}$$

Now, since we only consider kernels whose DFT is built from a radial profile (linearly interpolated on d points), one can show that the optimal radial profile is the minimum of a quadratic function and thus can be obtained by solving a small linear system. In Fig. 5 we can see that the profile of the oracle radial kernel is unimodal, with a mode at a position which is close to the one estimated by Algorithm 2. The restored images obtained with this oracle filter are also displayed in Fig. 4: they are a little more precise on the details, but present a significant amount of structured noise; indeed, such a noise is not very costly for a l^2 risk function. This is somehow reassuring: this shows that the structured noise also appearing with Algorithm 2 (in particular with $\mu = 0$) is truly a limit of techniques based on linear filtering.

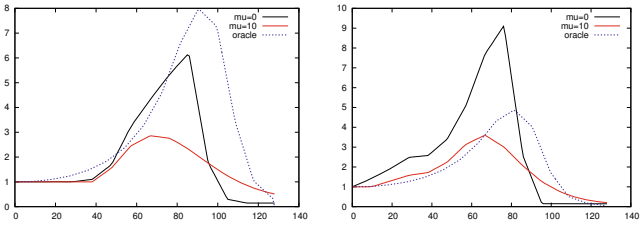


Fig. 5. Different radial profiles (oracle, and Algorithm 2 with $\mu = 0$ and $\mu = 10$) obtained on images *Capitol* (left) and *Parrots* (right)

5 Conclusion

We introduced a new variant of the GPC and SI image quality metrics, that can be computed four times faster than SI. This new index S provides a sharpness

measure that can be used in a stochastic optimization framework to achieve blind deblurring through linear convolution with a radial unimodal kernel. Though suffering from the limits of linear filtering, the obtained results are convincing, and visually similar to the best possible ones (oracle) obtained by such an approach. The extension to motion blur kernels could be an interesting generalization, as well as the use of more sophisticated (non-linear) restoration techniques to make the best usage of the selection performances of the S metric.

Acknowledgments. This work has been supported by the French National Research Agency under grant ANR-09-BLAN-0029-01.

References

1. Blanchet, G., Moisan, L., Rougé, B.: Measuring the Global Phase Coherence of an Image. In: Proceedings of ICIP 2008, pp. 1176–1179 (2008)
2. Blanchet, G., Moisan, L.: An Explicit Sharpness Index Related to Global Phase Coherence. In: Proceedings of ICASSP 2012, pp. 1065–1068 (2012)
3. Calderero, F., Moreno, P.: Evaluation of Sharpness Measures and Proposal of a Stop Criterion for Reverse Diffusion in the Context of Image Deblurring. In: Proceedings of VISAPP 2013 (2013)
4. 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), 161–179 (2011)
5. Desolneux, A., Moisan, L., Morel, J.-M.: Dequantizing image orientation. IEEE Transactions on Image Processing 11(10), 1129–1140 (2002)
6. Desolneux, A., Moisan, L., Ronsin, S.: A Compact Representation of Random Phase and Gaussian Textures. In: Proceedings of ICASSP 2012, pp. 1381–1384 (2012)
7. Frisen, M.: Unimodal Regression. Journal of the Royal Statistical Society. Series D The Statistician 35(4), 479–485 (1986)
8. Galerne, B., Gousseau, Y., Morel, J.-M.: Random Phase Textures: Theory and Synthesis. IEEE Transactions on Image Processing 20(1), 257–267 (2011)
9. Hassen, R., Wang, Z., Salama, M.: No-Reference Image Sharpness Assessment Based on Local Phase Coherence Measurement. In: Proc. of ICASSP 2010, pp. 2434–2437 (2010)
10. Kovesi, P.: Phase Congruency: a Low-level Image Invariant. Psychological Research 64, 136–148 (2000)
11. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient Marginal Likelihood Optimization in Blind Deconvolution. In: Proceedings of CVPR 2011 (2011)
12. Moisan, L.: Periodic Plus Smooth Image Decomposition. Journal of Mathematical Imaging and Vision 39(2), 120–145 (2011)
13. Morrone, M.C., Burr, D.C.: Feature Detection in Human Vision: a Phase-Dependent Energy Model. Proc. R. Soc. Lond. B235, 221–245 (1988)
14. Oppenheim, A.V., Lim, J.S.: The Importance of Phase in Signals. Proceedings of the IEEE 69, 529–541 (1981)
15. Wang, Z., Simoncelli, E.P.: Local Phase Coherence and the Perception of Blur. Adv. Neural Information Processing Systems (NIPS 2003) 16, 786–792 (2004)
16. Zhu, X., Milanfar, P.: Automatic Parameter Selection for Denoising Algorithms Using a No-Reference Measure of Image Content. IEEE Transactions on Image Processing 19(12), 3116–3132 (2010)

A Cascadic Alternating Krylov Subspace Image Restoration Method

Serena Morigi[1], Lothar Reichel[2], and Fiorella Sgallari[1]

[1] Department of Mathematics-CIRAM,
University of Bologna, Bologna, Italy
{serena.morigi,fiorella.sgallari}@unibo.it
[2] Department of Mathematical Sciences,
Kent State University, Kent, OH 44242, USA
reichel@math.kent.edu

Abstract. This paper describes a cascadic image restoration method which at each level applies a two-way alternating denoising and deblurring procedure. Denoising is carried out with a wavelet transform, which also provides an estimate of the noise-level. The latter is used to determine a suitable regularization parameter for the Krylov subspace iterative deblurring method. The cascadic multilevel method proceed from coarse to fine image resolution, using suitable restriction and prolongation operators. The choice of the latter is critical for the performance of the multilevel method. We introduce a special deblurring prolongation procedure based on TV regularization. Computed examples demonstrate the effectiveness of the method proposed for determining image restorations of high quality.

1 Introduction

Image restoration is a classical and important research area in image processing. Let the function f^δ represent the available noise- and blur-contaminated two-dimensional image, and let the function $\hat{u}$ represent the associated (unknown) blur- and noise-free image that we would like to recover. We assume the functions f^δ and $\hat{u}$ to be related by the degradation model

$$f^\delta(x) = \int_\Omega h(x,y)\hat{u}(y)dy + \eta^\delta(x), \qquad x \in \Omega, \tag{1}$$

where Ω is a square or rectangle on which the image is defined, η^δ represents additive noise (error) in the data f^δ, and h is the point-spread function (PSF). The integral may represent a space-invariant or space-variant blurring operator. We would like to recover $\hat{u}$ given the observed image f^δ and the PSF h.

It is well known that the solution of (1) is an ill-posed inverse problem and therefore computationally challenging. Many algorithms are available for determining an approximate solution of (1), including recently proposed multilevel and alternating methods; see, e.g., [1, 2, 7, 9–11, 13]. To be able to determine

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 98–109, 2013.

accurate restorations, the methods apply regularization, i.e., they replace the original problem by a nearby one that is less sensitive to perturbations.

The multilevel methods proposed in [9–11] proceed from coarser to finer image resolution levels and are based on regularization by truncated iteration on each level. Prolongation of a coarse-level approximation of $\hat{u}$ to a finer level is carried out with the aid of nonlinear edge-preserving and noise-reducing operators. Restrictions are computed by a local weighted least-squares method that is designed to preserve structures, such as edges, in the image. For many image restoration problems, the multilevel methods demand fewer matrix-vector product evaluations on the finest level than the corresponding one-level truncated iterative methods and often determine restorations of higher quality. The number of iterations on each level is based on a computed estimate of the amount of noise-contamination on each level.

The attractions of alternating iterative image restoration schemes, such as the ones described in [1, 7, 13], include that deblurring and denoising can be carried out independently, which simplifies the design and implementation of these schemes, and that they often yield restorations of high quality. Huang et al. in [7] describe a two-way alternating iterative method in which regularization is achieved by a Total Variation (TV)-norm operator.

This paper proposes a new multilevel alternating method for solving image restoration problems (1). The method applies an alternating method on each level of a cascadic multilevel method, going from coarser to finer image resolution. Denoising is achieved by wavelet transformation, and yields estimates of the amount of noise on each level. These estimates determine the regularization parameter for Tikhonov regularization, which is for deblurring. The prolongation from coarser to finer resolution introduces slight blurring in the image. Therefore, to further improve the quality of the restored image, we combine prolongation with TV regularization.

This paper is organized as follows. Sect. 2 describes the new image restoration method, in Sect. 3 we discuss the details of the denoising, deblurring, and prolongation steps. Sect. 4 presents a few computed examples, and concluding remarks can be found in Sect. 5.

2 A Cascading-Alternating Image Restoration Method

Consider a discretization of (1) and let the gray-scale image in the left-hand side of (1) be represented by an array of $n \times n$ pixels. Ordering the pixels column-wise defines a vector in $\mathbb{R}^{n^2}$, which we also denote by f^δ. The integral operator in (1) is represented by the matrix $H \in \mathbb{R}^{n^2 \times n^2}$, which typically is large and severely ill-conditioned. Let

$$W_1 \subset W_2 \subset \cdots \subset W_m$$

be a sequence of nested subspaces of $\mathbb{R}^{n^2}$ with W_j of dimension $\dim(W_j) = N(j)$ and $N(1) < N(2) < \ldots < N(m) = n^2$. We refer to the subspaces W_j as levels, with W_1 being the coarsest and $W_m = \mathbb{R}^{n^2}$ the finest level. The restriction operator $R_j : \mathbb{R}^{n^2} \to W_j$ is such that

$$H_j = R_j H R_j^T \qquad f_j^\delta = R_j f^\delta, \qquad 1 \le j < m, \tag{2}$$

where the R_j are determined by repeated local weighted least-squares approximation; see [9–12] for more details.

Going from level 1 to m, we apply on each level an alternating procedure for denoising and deblurring. To simplify the notation, we refer to the representations of H_j and f_j^δ on level j also by H and f^δ, respectively. The meaning of these and other matrices and vectors is clear from the context. Thus, on level j the initial iterate is $u^{(0)} := f^\delta \in \mathbb{R}^{N(j)}$ and the alternating method carries out the iterations, for $i = 1, 2, 3, \ldots$,

$$w^{(i)} = S_w(u^{(i-1)}) := \underset{w \in \mathbb{R}^{N(j)}}{\operatorname{argmin}} \{\|w - u^{(i-1)}\|^2 + \sum_k \lambda_k \phi(\langle w, \psi_k \rangle)\}, \tag{3}$$

$$u^{(i)} = S_h(w^{(i)}) := \underset{u \in \mathcal{K}_\ell}{\operatorname{argmin}} \{\|Hu - f^\delta\|^2 + \alpha \|u - w^{(i)}\|^2\}, \tag{4}$$

where the regularization parameter $\alpha > 0$ is determined by the discrepancy principle using an estimate of the noise in the image on level j. Thus, α depends on the level j; see below for details. The function ϕ in (3) is a penalty function, the λ_k denote weights, and $\{\psi_k\}$ is an orthonormal wavelet basis. A common choice of penalty function is $\phi(x) = |x|^p$ for some $1 \le p \le 2$. We use this penalty function with $p = 1$. Minimization in (4) is on every level carried out over an ℓ-dimensional Krylov subspace $\mathcal{K}_\ell$ determined by ℓ steps of Golub-Kahan bidiagonalization applied to H with initial vector f^δ; see Subsection 3.2 for the definition of $\mathcal{K}_\ell$ and further details.

The prolongation operators are nonlinear edge-preserving and noise-reducing, see Sect. 3, while the restriction operators are determined by weighted local least-squares approximation following [11]. The purpose of the weights is to avoid smearing of edges. Specifically, the prolongation method, inspired by the work [8] for super-resolution image processing, maps the image $u^{(i)} \in \mathbb{R}^{N(j)}$ from level j to an image $u^{(0)} \in \mathbb{R}^{N(j+1)}$ on level $j + 1$,

$$u^{(0)} = S_{tv}(u^{(i)}) := \underset{u \in \mathbb{R}^{N(j+1)}}{\operatorname{argmin}} \{\|u\|_{TV} + \beta \|u^{(i)} - R(G * u)\|^2\}, \tag{5}$$

where $\|\cdot\|_{TV}$ is a vector semi-norm of TV-type, $\beta > 0$ is an empirically determined fixed parameter [8], and R is the restriction operator used in the cascadic procedure. The kernel is assumed to be a convolution, and $*$ denotes convolution. In the computed examples we use the Gaussian kernel

$$G(x, y) := \frac{1}{4\pi\gamma} e^{-(x^2+y^2)/4\gamma}, \tag{6}$$

where γ is tuned based on the fact that the higher-resolution image has four times as many pixels as the lower resolution image. The image $u^{(0)}$ obtained from (5) in this manner is applied in (3), i.e., $u^{(0)}$ is the first iterate of the alternating method on level $j + 1$.

3 Denoising, Deblurring, and Prolongation Methods

This section describes the denoising, deblurring, and prolongation methods that are used in the cascadic alternating method.

3.1 Denoising

Denoising methods seek to remove the noise in an image without removing the signal. Thresholding in the wavelet domain for denoising has been pioneered by Donoho [4]. Nonlinear soft thresholding in the wavelet transform domain consists of three steps: 1) linear forward wavelet transformation, 2) nonlinear shrinkage denoising based on thresholding of the wavelet coefficients, and 3) linear inverse wavelet transformation.

In the denoising step (3) of the cascadic alternating method, the first term in brackets can be written as

$$\|w - u^{(i-1)}\|^2 = \sum_k (\langle w, \psi_k \rangle - \langle u^{(i-1)}, \psi_k \rangle)^2$$

by using the unitary invariance property of the 2-norm. Therefore (3) can be expressed as

$$w^{(i)} = \underset{w \in \mathbb{R}^{N(j)}}{\operatorname{argmin}} \left\{ \sum_k \left((\langle w, \psi_k \rangle - \langle u^{(i-1)}, \psi_k \rangle)^2 + \lambda_k |\langle w, \psi_k \rangle| \right) \right\}. \tag{7}$$

The solution of (7) is obtained by soft thresholding [4]:

$$\langle w^{(i)}, \psi_k \rangle = \begin{cases} \langle u^{(i-1)}, \psi_k \rangle - \lambda_k/2, & \text{if} \langle u^{(i-1)}, \psi_k \rangle \geq \lambda_k/2 \\ \langle u^{(i-1)}, \psi_k \rangle + \lambda_k/2, & \text{if} \langle u^{(i-1)}, \psi_k \rangle \leq -\lambda_k/2 \\ 0, & \text{otherwise.} \end{cases}$$

The threshold parameter λ_k is determined by the BayesShrink soft thresholding technique as described in [3]. Our cascadic alternating method applies this denoising technique as a first step on each level of the alternating method. This yields an estimate of the amount of noise in the currently available contaminated image. It is important that a fairly accurate estimate of the noise is available in the subsequent deblurring step of the alternating method to be able to determine a suitable value of the regularization parameter. Following [3], we use on each level j the robust median estimator for the noise. Thus, the variance of the noise σ^2 is estimated by

$$\widehat{\sigma}_j = MAD_j / C, \tag{8}$$

where MAD_j denotes the median absolute value of appropriately normalized fine-scale wavelet coefficients, and following [3], we let $C = 0.6745$. An estimate of the norm of the noise in f^δ, required in the deblurring step, now is obtained from (8),

$$\delta^2 = \widehat{\sigma}_j^2 N(j).$$

We use this formula to estimate the amount of noise on all levels, including the finest one.

3.2 Deblurring

In step (4) on level j of the alternating method, we solve a sequence of discrete image deblurring problems by the iterative Krylov subspace method proposed in [1]. The solution method is based on partial Golub-Kahan bidiagonalization of the blurring matrix H_j with initial vector f_j^δ given by the restriction (2). Similarly as above, we denote H_j and f_j^δ by H and f^δ, respectively. Application of ℓ steps of Golub-Kahan bidiagonalization to H yields the matrices $U_{\ell+1} \in \mathbb{R}^{N(j)\times(\ell+1)}$ and $V_\ell \in \mathbb{R}^{N(j)\times\ell}$ with orthonormal columns, and a lower bidiagonal matrix $\bar{C}_\ell \in \mathbb{R}^{(\ell+1)\times\ell}$ with positive diagonal and subdiagonal entries such that

$$HV_\ell = U_{\ell+1}\bar{C}_\ell, \qquad H^*U_\ell = V_\ell C_\ell^*, \qquad U_{\ell+1}e_1 = f^\delta/\|f^\delta\|, \tag{9}$$

where $U_\ell \in \mathbb{R}^{N(j)\times\ell}$ is made up of the ℓ first columns of $U_{\ell+1}$, $C_\ell \in \mathbb{R}^{\ell\times\ell}$ consists of the first ℓ rows of $\bar{C}_\ell$, the superscript * denotes transposition, and $e_1 = [1, 0, \ldots, 0]^*$ is the first axis vector. The columns of V_ℓ span the Krylov subspace

$$\mathcal{K}_\ell := \mathcal{K}_\ell(H^*H, H^*f^\delta) := \mathrm{span}\{H^*f^\delta, (H^*H)H^*f^\delta, \ldots, (H^*H)^{\ell-1}H^*f^\delta\}.$$

We assume ℓ to be small enough, so that the decompositions (9) with the stated properties exist. Substituting $u = V_\ell y$ into (4) yields the reduced minimization problem

$$\begin{aligned}\min_{y\in\mathbb{R}^\ell} \left\{ \|\bar{C}_\ell y - e_1\|f^\delta\|\|^2 + \alpha\|y - V_\ell^* w^{(i)}\|^2 \right\} \\ = \min_{y\in\mathbb{R}^\ell} \left\| \begin{bmatrix} \bar{C}_\ell \\ \sqrt{\alpha}\, I_\ell \end{bmatrix} y - \begin{bmatrix} e_1\|f^\delta\| \\ \sqrt{\alpha}\, V_\ell^* w^{(i)} \end{bmatrix} \right\|^2,\end{aligned} \tag{10}$$

where, for $i > 1$, $w^{(i)}$ is obtained from the previous alternating step (3), and $w^{(0)} := u^{(0)}$. The minimization problem (10) has a unique solution $y_\ell = y_{\ell,\alpha}$ for any $\alpha > 0$, and the corresponding solution of (4) is given by

$$u^{(i)} = u_\ell = V_\ell y_\ell. \tag{11}$$

Let δ be an available bound for the Euclidean norm of the error in f^δ. A vector u is said to satisfy the discrepancy principle if $\|Hu - f^\delta\| \le \eta\delta$ for some chosen value of the parameter η. Typically, η is chosen to be close to unity if an accurate estimate of the norm of the noise δ is available. We let the regularization parameter α be as large as possible so that the solution (11) of (4) satisfies the discrepancy principle, i.e., so that

$$\|Hu_\ell - f^\delta\| = \eta\delta. \tag{12}$$

It follows from (9) and (11) that

$$\|Hu_\ell - f^\delta\| = \|\bar{C}_\ell y_\ell - e_1\|f^\delta\|\,\|. \tag{13}$$

Therefore, a value of α such that the computed solution u_ℓ satisfies (12) can be determined by only considering the reduced problem in the right-hand side of (13).

The determination of such a value of α typically requires the solution of a sequence of small least-squares problems (10), each problem corresponding to a different value of α. We may solve these problems, e.g., by using the singular value decomposition of the matrix $\bar{C}_\ell$, or more cheaply by applying a scheme described by Eldén [5]. Zero-finders for determining a value of α such that (12) holds are discussed in [1].

The computations on each level are terminated as soon as two successive approximate solutions $w^{(i)}$ and $w^{(i-1)}$ are sufficiently close; see (15) below. The convergence of a Krylov subspace-based alternating one-level method is established in [1] by an adaption of the convergence proof in [7]. The computations with the alternating multilevel method of the present paper on levels $1, 2, \ldots, m-1$, i.e., on all levels but the finest one, may be considered preprocessing for a one-level Krylov subspace-based alternating method. The purpose of the preprocessing is to determine an accurate initial iterate for alternation on the finest level. Since the convergence result does not depend on the use of a particular initial iterate, the convergence proof in [1] applies to multilevel methods. In fact, convergence on each level can be established by considering the computations on the previous levels a preprocessing step designed to determine an accurate initial approximate solution of the solution on the next level.

The convergence proofs in [1, 7] do not address the quality of the restored images in the sense that on each level the stopping rule (15) may be satisfied by many images of varying quality. In fact, the quality of the computed restoration depends on the quality of the initial iterate on the finest level. An accurate initial iterate may help determine an accurate restoration. This is illustrated in [10, 11], and is one of the benefits of multilevel methods. The design of the prolongation method therefore is important. It is also important that no high-frequency errors, such as spurious edges, are introduced during the computations on the first $m-1$ levels, because such errors may be difficult to remove on the finest level.

3.3 Prolongation

The cascadic alternating method requires prolongation operators to be applied to map the computed approximate solution from level j to the next finer level $j+1$ for all j. Both linear and nonlinear prolongation operators can be used; see [9] and reference therein.

The prolongation step is a super-resolution process and suffers from similar difficulties as the latter due to the ill-conditioning of the problem. In fact, high-resolution and low-resolution images are typically related through a convolution operator and a down-sampling operator. Several methods have been proposed in the literature for super-resolution. Many of them are based on least-squares approximation, the use of Fourier series, and other L_2-norm approximation methods; see Marquina and Osher [8], who propose a variational method that uses the TV-norm as regularizing functional for deblurring and oversampling images.

We can solve the Euler-Lagrange equation associated to the variational problem (5) by means of the gradient-descent method formulated as the time evolution equation

$$\frac{\partial u}{\partial t} = \nabla \cdot \frac{\nabla u}{|\nabla u|} + \beta G * (\mathcal{S}(u^{(i)}) - \mathcal{S}(R(G * u))),$$

where $\mathcal{S}$ represents an up-sampling operator implemented as a bilinear interpolation, R is the restriction operator, and G is defined in (6). We considered homogeneous Neumann boundary conditions and initialize with $u^0 = S(u^{(i)})$.

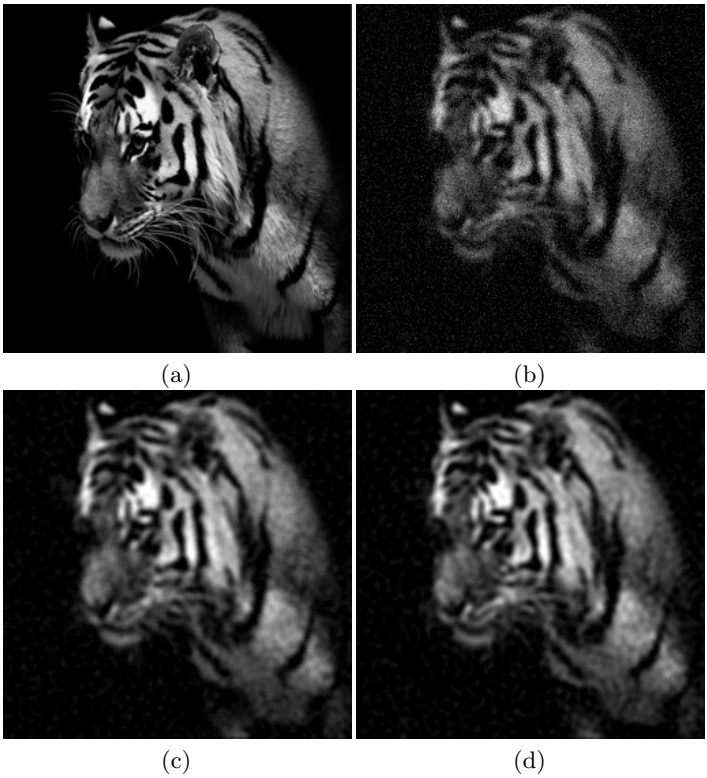


Fig. 1. Example 4.1: Restoration of corrupted version of the `tiger` image: (a) Unperturbed image; (b) the corrupted image produced by Gaussian blur, determined by the parameters `band` = 5 and `sigma` = 3, and by 20% noise, SNR=9.05; (c) restored images with 1−level alternating method (SNR=10.79), (d) 2−level cascadic-alternating method (SNR=11.89).

4 Numerical Experiments

This section illustrates the performance of the cascadic alternating method defined by (3)-(4) and (5). Given a representation of the blur- and noise-free image $\hat{u} \in \mathbb{R}^{n^2}$, we determine a blur- and noise-contaminated image $f^\delta \in \mathbb{R}^{n^2}$ from

$$f^\delta = H\hat{u} + e.$$

The "noise-vector" $e \in \mathbb{R}^{n^2}$ has normally distributed entries with mean zero, and is scaled to yield a desired noise-level

$$\delta = \frac{\|e\|}{\|\hat{u}\|}. \tag{14}$$

Our task is to compute an accurate approximation of $\hat{u}$, given f^δ and H by the cascadic-alternating iterative method. We terminate the alternating iterations and accept $w^{(i)}$ as the computed approximation of $\hat{u}$ as soon as the relative difference between consecutive iterates $w^{(1)}, w^{(2)}, w^{(3)}, \ldots$ is sufficiently small; specifically, we accept $w^{(i)}$ when for the first time

$$\|w^{(i)} - w^{(i-1)}\|/\|w^{(i)}\| < 1 \cdot 10^{-4}. \tag{15}$$

The displayed restored images provide a qualitative measure of the performance of the alternating methods. The signal-to-noise ratio

$$\mathrm{SNR}(w^{(i)}, \hat{u}) = 20 \log_{10} \frac{\|\hat{u}\|}{\|w^{(i)} - \hat{u}\|} \,\mathrm{dB}$$

is a quantitative measure of the quality of $w^{(i)}$. A high SNR-value indicates that the restoration is accurate.

Example 4.1. We consider the restoration of `tiger` images that have been corrupted by white Gaussian noise and Gaussian blur. Each image is represented by 256×256 pixels, i.e., $n = 256$. The block-Toeplitz-Toepliz-block matrix H represents a Gaussian blurring operator and is generated with the MATLAB function `blur.m` from Regularization Tools [6]. This function has two parameters, `band` and `sigma`. The former specifies the half-bandwidth of the Toeplitz blocks and the latter the variance of the Gaussian point spread function. The larger `sigma`, the more blurring. Enlarging `band` increases the storage requirement, the arithmetic work necessary for the evaluation of matrix-vector products with H, and to some extent the blurring.

Tables 1 and 2 report results achieved with the cascadic-alternating method of this paper and compare them to results obtained with a corresponding one-level alternating method for several noise-levels δ. The first column of Table 1 shows the cascadic level and the second column displays the noise-level (14). The third column, labeled SNR_i, reports the SNR-values for the available contaminated image f^δ, i.e., the value $\mathrm{SNR}(f^\delta, \hat{u})$. Columns four and five display the SNR-values of the restored images determined by two levels of the cascadic-alternating method after the alternating procedure at a given cascadic level (SNR_{alt}) and after prolongation (SNR_{prol}) from the first to the second level. The number of alternating iterations is reported in brackets (its). SNR_{alt} in the sixth column refers to a basic one-level alternating method applied to the given contaminated image on the finest level only. The number of iterations required (its) is also shown. Thus, the SNR-values increase with each level of the alternating method. Moreover, the initial image for the second level has a larger SNR-value than the available contaminated image f^δ. The parameter η in (12) is set to 0.4 on the

Table 1. Example 4.1: Results for restorations of `tiger` images that have been corrupted by Gaussian blur, determined by `band` = 5 and `sigma` = 3, and by noise corresponding to noise-level δ.

level	δ	SNR_i	SNR_{alt}(its)	SNR_{prol}	SNR_{alt}(its)
1	0.10	10.82	12.15(2)	12.08	
2			13.18(1)		
					12.85 (4)
1	0.20	9.05	11.33(1)	11.60	
2			11.89(1)		
					10.79 (3)
1	0.30	7.00	10.31(1)	10.75	
2			11.31(1)		
					10.72 (3)

first level and to 0.98 on the finer level. The number of bidiagonalization steps is set to $\ell = 10$.

Tables 1 and 2 show the restorations obtained by cascadic-alternating multilevel method to be of higher quality, as measured by the SNR-values, than restorations computed by one-level alternating methods. This is in agreement with visual perception. The SNR_{prol}-values, which displays the SNR-value of the restored image after prolongation, show how the prolongation method improves the restorations. The observed blurred and noisy image represented by f^δ is shown on the right-hand side of Fig. 1(a) and the restoration $w^{(3)}$ is depicted in Fig. 1(d). □

Example 4.2. We consider the restoration of blur- and noise-contaminated `butterfly` images. They are represented by 512×512 pixels, i.e., $n = 512$. The exact image is shown in Fig. 2(a). The observed image is corrupted by white Gaussian noise and Gaussian blur, characterized by the parameter values of `band` and `sigma`.

Table 3 is analogous to Tables 1 and 2, and reports SNR-values for restored `butterfly` images determined by the proposed cascadic alternating method and by a corresponding one-level alternating method. We observe that the computational effort required by the cascadic alternating method is smaller than for the one-level alternating method, due to the fact that the cascadic alternating method only requires one iteration on each level, while the one-level alternating method demands 4 iterations on the finest level. Since the computational cost of each cascadic alternating iteration grows with the image dimension, only the cost for the iteration on the finest level is significant. The parameter η in (12) is set to 0.5, 0.9, and 0.95, from the coarsest to finest level. The number of bidiagonalization steps ℓ is increased with the level number according to $\ell = 5, 10, 20$.

Our experimental results show that the quality of restored images obtained with a three-level cascadic alternating method is competitive with a corresponding one-level alternating method with regard to image quality as well as with regard to computational effort, since all iterations with the one-level method are carried out on the finest level.

Table 2. Example 4.1: Results for restorations of `tiger` images that have been corrupted by Gaussian blur, determined by `band` = 3 and `sigma` = 3, and by noise corresponding to noise-level δ.

level	δ	SNR_i	SNR_{alt}(its)	SNR_{prol}	SNR_{alt}(its)
1	0.15	12.12	12.85(2)	13.75	
2			14.88(1)		
					14.50 (4)
1	0.30	8.02	10.72(1)	12.07	
2			12.88(1)		
					12.70 (4)
1	0.45	4.97	8.91(2)	10.58	
2			11.18(1)		
					10.75 (5)

Table 3. Example 4.2: Results for restorations of `butterfly` images that have been corrupted by Gaussian blur, determined by variable `band` and `sigma`, and by noise corresponding to noise-level $\delta = 20\%$.

level	`band`	`sigma`	SNR_i	SNR_{alt}(its)	SNR_{prol}	SNR_{alt}(its)
1	3	3	11.57	11.05(1)	12.10	
2				13.08(1)	13.55	
3				15.35(1)		
						15.03 (4)
1	5	3	9.43	10.55(1)	10.64	
2				12.26(1)	12.38	
3				13.33(1)		
						13.02 (4)
1	7	3	8.09	9.11(1)	9.58	
2				10.82(1)	10.89	
3				11.74(1)		
						11.43 (4)
1	5	5	9.26	10.33(1)	10.53	
2				12.05(1)	12.06	
3				13.03(1)		
						12.65 (4)

The contaminated blurred and noisy image represented by f^δ is shown in Fig. 2(b) and the restorations obtained by the cascadic alternating and by the one-level alternating methods are depicted in Fig. 2(c) and 2(d), respectively.

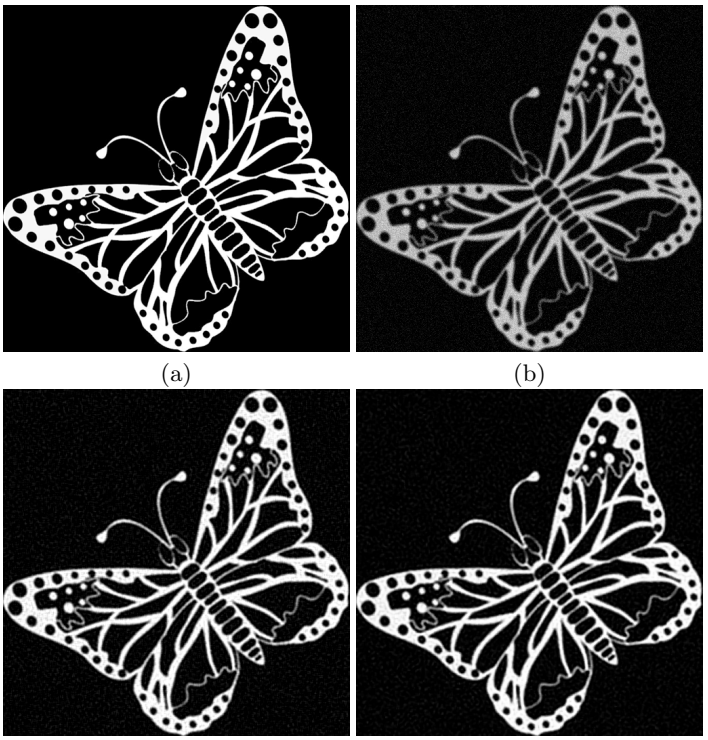


Fig. 2. Example 4.2: Restoration of corrupted version of the `butterfly` image: (a) unperturbed image ; (b) the corrupted image produced by Gaussian blur, determined by the parameters `band` = 5 and `sigma` = 5, and by 20% noise, SNR =9.26; (c) restored image by the 1-level alternating method with 4 iterations; (d) restored image determined by 3-levels of cascadic alternating.

5 Conclusion and Further Developments

This paper describes a new cascadic alternating method for image deblurring and denoising, in which we alternate between deblurring, carried out by a Krylov subspace iterative method based on partial Golub-Kahan bidiagonalization of the blurring matrix, and denoising by wavelet thresholding. The method combines the performance of a cascadic method with the well-known accuracy obtained by an alternating method at each level. Further numerical results and comparisons with state-of-the-art methods will be reported and convergence properties and accuracy aspects will be discussed in forthcoming work.

Acknowledgment. Research by LR was supported in part by NSF grant DMS-1115385. This work was partially supported by GNCS-INDAM 2012 project, $ex60\%$ project by University of Bologna "Funds for selected research topics".

References

1. Abad, J.O., Morigi, S., Reichel, L., Sgallari, F.: Alternating Krylov subspace image restoration methods. J. Comput. Applied Math. 236, 2049–2062 (2012)
2. Chan, T.F., Chen, K.: An optimization-based multilevel algorithm for total variation image denoising. Multiscale Model. Simul. 5, 615–645 (2006)
3. Chang, S.G., Yu, B., Vetterli, M.: Adaptive wavelet thresholding for image denoising and compression. IEEE Trans. Image Proc. 9, 1532–1546 (2000)
4. Donoho, D.L.: De-noising by soft-thresholding. IEEE Trans. Inf. Theory 41, 613–627 (1995)
5. Eldén, L.: Algorithms for the regularization of ill-conditioned least squares problems. BIT 17, 134–145 (1977)
6. Hansen, P.C.: Regularization tools version 4.0 for Matlab 7.3. Numer. Algorithms 46, 189–194 (2007)
7. Huang, Y., Ng, M.K., Wen, Y.-W.: A fast total variation minimization method for image restoration. Multiscale Model. Simul. 7, 774–795 (2008)
8. Marquina, A., Osher, S.: Image super-resolution by TV-regularization and Bregman iteration. J. Sci. Comput. 37, 367–382 (2008)
9. Morigi, S., Reichel, L., Sgallari, F., Shyshkov, A.: Cascadic multiresolution methods for image deblurring. SIAM J. Imaging Sci. 1, 51–74 (2008)
10. Morigi, S., Reichel, L., Sgallari, F.: Noise-reducing cascadic multilevel methods for linear discrete ill-posed problems. Numer. Algorithms 53, 1–22 (2010)
11. Morigi, S., Reichel, L., Sgallari, F.: Cascadic multilevel methods for fast nonsymmetric blur- and noise-removal. Appl. Numer. Math. 60, 378–396 (2010)
12. Morigi, S., Reichel, L., Sgallari, F.: An edge-preserving multilevel method for deblurring, denoising, and segmentation. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 427–439. Springer, Heidelberg (2009)
13. Wen, Y.-W., Ng, M.K., Ching, W.-K.: Iterative algorithms based on decoupling of deblurring and denoising for image restoration. SIAM J. Sci. Comput. 30, 2655–2674 (2008)

B-SMART: Bregman-Based First-Order Algorithms for Non-negative Compressed Sensing Problems

Stefania Petra*, Christoph Schnörr, Florian Becker, and Frank Lenzen

IPA & HCI, Heidelberg University,
Speyerer Str. 6, 69115 Heidelberg, Germany
{petra,schnoerr,becker}@math.uni-heidelberg.de,
frank.lenzen@iwr.uni-heidelberg.de
http://ipa.iwr.uni-heidelberg.de,
http://hci.iwr.uni-heidelberg.de

Abstract. We introduce and study Bregman functions as objectives for non-negative sparse compressed sensing problems together with a related first-order iterative scheme employing non-quadratic proximal terms. This scheme yields closed-form multiplicative updates and handles constraints implicitly. Its analysis does not rely on global Lipschitz continuity in contrast to established state-of-the-art gradient-based methods, hence it is attractive for dealing with very large systems. Convergence and a $O(k^{-1})$ rate are proved. We also introduce an iterative two-step extension of the update scheme that accelerates convergence. Comparative numerical experiments for non-negativity and box constraints provide evidence for a $O(k^{-2})$ rate and reveal competitive and also superior performance.

Keywords: multiplicative algebraic reconstruction, compressed sensing, underdetermined systems of nonnegative linear equations, convergence rates, limited angle tomography.

1 Introduction

Overview. Since the advent of Compressed Sensing [8,12] it is well-known that the sparsest solution of an underdetermined system of equations can be found via ℓ_1-minimization under adequate conditions. In many interesting applications the vector x^* to be recovered is nonnegative or even binary. Recent results [24,17,13,21] show that under appropriate conditions, a sparse nonnegative (or binary) solution is also the *unique* solution of

$$Ax = b, \quad x \in X, \tag{1}$$

with $X = \mathbb{R}^n_+$ or $X = [0,1]^n$, and thus recovery reduces to a simpler feasibility problem. As a consequence, this may lead to alternatives superior to ℓ_1-minimization since *any* objective function subject to the constraints (1) can

* Gratefully acknowledges support by the State Ministry of Baden-Württemberg for Sciences, Research and the Arts.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 110–124, 2013.

recover the sparse solution. On the other hand, (1) becomes infeasible when noise is present, and we have to allow for a distance of Ax^* to b.

In this paper we suggest and study the approach

$$x^* = \text{argmin}_{x \in X} f(x), \qquad f(x) := B_\phi(Ax, b), \tag{2}$$

with B_ϕ an appropriate Bregman distance induced by ϕ. In the case of the Euclidean distance $B_\phi(x, y) = \frac{1}{2}\|x - y\|_2^2$ it is shown in [22] that recovery of nonnegative sparse solutions via nonnegative least-squares is stable and outperforms ℓ_1-regularization when combined with thresholding. Other choices for B_ϕ can be more adequate, however, if the noise is non-Gaussian, like e.g. Poisson noise in tomographic applications, or when the data b and the sensor matrix A are nonnegative.

For particular sparse (nonnegative) images and tomographic projection matrices A the *Simultaneous Multiplicative Algebraic Reconstruction Technique (SMART)* recently proved to be quite efficient by returning meaningful solutions after few iterations [1]. It applies only to the specific but important case of systems with nonnegative b and A. SMART has been invented and re-invented several times in the field of medical imaging. Convergence was proved in [7]. For consistent projection equations (1), it returns the feasible point in $\{Ax = b, x \geq 0\}$ that minimizes the cross-entropy distance $KL(x, x^0)$ to the initial vector x^0. When all entries of x^0 are all equal SMART converges to the maximizer of the Shannon entropy.

In a nutshell, past studies showed that SMART:

1. is adequate for ill-conditioned problems and huge problem sizes,
2. converges provably,
3. performs at each iteration only a single multiplication with A and $A^\top$, and
4. returns meaningful solution after few iterations.

Contribution and Organization. Motivated by the specific case of SMART (section 2.1), we introduce in section 2.2 an iterative scheme for the general case (2) based on a linearized objective and a related Bregman-based proximal term, that enables closed-form multiplicative updates and handles the constraints implicitly. We prove convergence and the convergence rate $O(k^{-1})$ in section 2.3.

Our approach may be understood as a blend of (i) optimal gradient-based schemes based on a linearized objective and upper bound surrogates through quadratic proximation, and (ii) fully nonlinear Bregman-based proximal iterations studied in [14]. While each step of the latter scheme is as costly as the original objective, the former schemes depend on the Lipschitz constant of the gradient of the objective that can be very large in large-scale nonnegative problems like 3D algebraic tomography. Our approach and the analyis do not require global Lipschitz continuity.

In section 2.4 we specifically consider Bregman distances induced by the Shannon entropy and by the Fermi-Dirac entropy and the corresponding multiplicative updates, to deal with nonnegativity or box constraints. Connections of the

resulting objectives to nonnegative least-squares and ℓ_1-regression, that substantiate our approach more formally, are outlined in section 2.5.

While proving a $O(k^{-2})$ convergence rate is beyond the scope of the present conference contribution, we suggest two algorithmic extensions called F(AST)-SMART in section 3, aking to a Bregman-based versions of established first-order optimal schemes [20,5]. Competitive numerical experiments discussed in section 4 illustrate the discussion above and support our claims.

Notation. We set $[n] = \{1, \ldots, n\}$ for $n \in \mathbb{N}$. $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and $\| \cdot \| = \| \cdot \|_2 = \langle \cdot, \cdot \rangle^{1/2}$ the corresponding norm. $\mathbb{1} = (1, \ldots, 1)^\top$, that is $\|x\|_1 = \langle \mathbb{1}, x \rangle$ for $x \in \mathbb{R}^n_+$. Vectors are enumerated with superscripts x^i, and vector and matrix components with subscripts x_i, A_{ij}, while matrix rows and columns we denote by $A_{i,\bullet}$ and $A_{\bullet,j}$ respectively. Vector inequalities $x \geq y$ and $\log x, \exp x$ etc., are understood component-wise. By x_+ we denote $\mathbb{1}^\top x$. $\Delta_n = \{x \geq 0 \colon \|x\|_1 = 1\} \subset \mathbb{R}^n_+$ denotes the probability simplex. $KL(x, y)$ denotes the Kullback-Leibler distance of two nonnegative vectors, see Appendix.

2 B-SMART

2.1 Motivation: The SMART Iteration

It is well known [7] that the *Simultaneous Multiplicative Algebraic Reconstruction Technique (SMART)* minimizes $f(x) = KL(Ax, b)$ over the positive orthant, provided that $A \geq 0$, $(A_{\bullet,j})_+ > 0, j \in [n]$ and $b > 0$. This corresponds to (2) with φ being the negative entropy (21). For a positive iterate $x^k \in \mathbb{R}^n_{++}$ the SMART iteration reads

$$x_j^{k+1} = x_j^k \prod_{i=1}^m \left(\frac{b_i}{\langle A_{i,\bullet}, x^k \rangle} \right)^{t_k A_{ij}}, \quad j \in [n]. \tag{3}$$

Here t_k is a relaxation parameter, with $t_k \leq \min_j \{(A_{\bullet,j})_+\}$. We observe that algorithm (3) employs at each step the minimization of the linearized objective f plus a "prox"-like term of the form

$$x^{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n_+} f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{t_k} KL(x, x^k), \tag{4}$$

with arbitrary starting vector $x^0 > 0$. This implies

$$\log(x^{k+1}) = \log x^k - t_k A^\top (\log Ax^k - \log b), \tag{5}$$

since for every $x^k > 0$, $x^{k+1} > 0$ holds as well. This is exactly the SMART iteration with relaxation parameter t_k.

Remark 1. We note that the above algorithm (4) is closely related to the gradient descent method

$$x^{k+1} = \operatorname{argmin}_x f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2t_k} \|x - x^k\|^2, \tag{6}$$

better known as $x^{k+1} = x^k - t_k \nabla f(x^k)$. For convex LC^1 functions there exist precise bounds for the value of t_k depending on the Lipschitz constant of the gradient of f. Moreover, convergence rates are well understood and optimal gradient methods have been established [18,5,19,23]. Our objective function f however is only locally Lipschitz-continuous, due to differentiability, and non-differentiable on the boundary of $\mathbb{R}^n_+$, where sparse solutions occur.

2.2 A Nonlinear Projected Gradient Method

In this section we derive convergence rates for the iteration (4) by considering a general minimization scheme for problems of the form (2).

Let $\varphi : X \to \mathbb{R}$ and $\phi : Y \to \mathbb{R}$ be convex and continuously differentiable on $\text{int}(X)$ and $\text{int}(Y)$ respectively, with $A(X) \subset Y$. Further define the distance-like functions $B_\varphi : X \times \text{int}(X) \to \mathbb{R}$ and $B_\phi : Y \times \text{int}(Y) \to \mathbb{R}$ by

$$B_\varphi(x, y) = \varphi(x) - \varphi(y) - \langle x - y, \nabla\varphi(y)\rangle \tag{7}$$

and

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle x - y, \nabla\phi(y)\rangle. \tag{8}$$

We assume $A(X) \subset Y$ and $b \in \text{int}(Y)$, and define $f : X \to \mathbb{R}$ by

$$f(x) = B_\phi(Ax, b). \tag{9}$$

Choosing an appropriate constant $c > 0$, we apply with $\nabla_x B_\phi(Ax, b) = A^\top (\nabla\phi(Ax) - \nabla\phi(b))$ the iteration

$$x^{k+1} = \text{argmin}_{x\in X} f(x^k) + \langle \nabla f(x^k), x - x^k\rangle + \frac{c}{t_k} B_\varphi(x, x^k) \tag{10}$$

$$= \text{argmin}_{x\in X} f(x^k) + \langle \nabla\phi(Ax^k) - \nabla\phi(b), Ax - Ax^k\rangle + \frac{c}{t_k} B_\varphi(x, x^k). \tag{11}$$

We will see that under an appropriate assumption the r.h.s. of (10) is an upper bound of f.

2.3 Convergence and Convergence Rates

Iteration (10) is exactly the nonlinear projected gradient method from [4], except for the fact that due to the particular form of the objective function, only relaxed conditions of f are required. In fact, we can replace the Lipschitz-condition in [4] by Assumption A, part (b), below.

Assumption A:

(a) X is a closed and convex set with nonempty interior;
(b) We have $B_\phi(Ax, Ay) \le cB_\varphi(x, y)$ for all $x, y \in X$;
(c) The set of optimal solutions $X^* := \text{argmin}_{x\in X} f(x)$ is nonempty.

The following results will turn out to be useful in the sequel.

Lemma 1 ([10, Lem 3.1]). *Let $S \subset \mathbb{R}^n$ be an open set with closure $\overline{S}$, and let $\psi : \overline{S} \to R$ be continuously differentiable on S. Then for any three points $a, b \in S$ and $c \in \overline{S}$ the following identity holds*

$$B_\psi(c,a) + B_\psi(a,b) - B_\psi(c,b) = \langle \nabla\psi(b) - \nabla\psi(a), c - a\rangle.$$

Theorem 1 ([3, Thm. 3.12]). *Suppose φ is closed proper convex and differentiable on $\operatorname{int}(\operatorname{dom}\varphi)$, X is closed convex with $X \cap \operatorname{int}(\operatorname{dom}\varphi) \neq \emptyset$, and $y \in \operatorname{int}(\operatorname{dom}\varphi)$. If φ is Legendre, then the Bregman projection $\overline{x}$ of y is unique and contained in $\operatorname{int}(\operatorname{dom}\varphi)$,*

$$\operatorname*{argmin}_{x \in X \cap \operatorname{dom}\varphi} B_\varphi(x,y) = \{\overline{x}\}, \qquad \overline{x} \in \operatorname{int}(\operatorname{dom}\varphi). \tag{12}$$

Remark 2. It is easy to see that assertion (12) also holds for the case

$$\operatorname*{argmin}_{x \in X \cap \operatorname{dom}\varphi} \big\{B_\varphi(x,y) + \langle l, x\rangle\big\} = \{z\}, \qquad z \in \operatorname{int}(\operatorname{dom}\varphi), \tag{13}$$

with $l \in \mathbb{R}^n$ arbitrary and $\|l\| \leq \infty$.

Our main result is stated next.

Theorem 2. *Under* **Assumption A** *above, for the sequence $\{x_k\}_{k\leq\kappa}$ generated by (10) with starting point $x^0 \in \operatorname{int}(X)$ and $t_k = t \leq 1$, one has:*

(a) Iteration (10) is well defined.
(b) For every κ,

$$\min_{0\leq k\leq\kappa} f(x^k) - \min_X f(x) \leq \frac{cB_\varphi(x^*, x^0)}{t\kappa}\ . \tag{14}$$

(c) The sequence $\{f(x_k)\}_{k\leq\kappa}$ is decreasing. In particular, the method converges.

Proof. Statement (a) follows by Remark 2.
(b) Let x^* be the optimal solution. The optimality conditions for (10) imply

$$\langle x - x^{k+1}, t_k\nabla f(x^k) + c\left(\nabla\varphi(x^{k+1}) - \nabla\varphi(x^k)\right)\rangle \geq 0, \quad x \in X.$$

In particular, for $x = x^*$ we get

$$\langle x^* - x^{k+1}, c\left(\nabla\varphi(x^k) - \nabla\varphi(x^{k+1})\right) - t_k\nabla f(x^k)\rangle \leq 0, \quad x \in X. \tag{15}$$

Since f is convex

$$\begin{aligned} 0 \leq t_k(f(x^k) - f(x^*)) \ &\leq\ t_k\langle x^k - x^*, \nabla f(x^k)\rangle && (16)\\ &= \langle x^* - x^{k+1}, c\left(\nabla\varphi(x^k) - \nabla\varphi(x^{k+1})\right) - t_k\nabla f(x^k)\rangle && (17)\\ &\quad + c\langle x^* - x^{k+1}, \nabla\varphi(x^{k+1}) - \nabla\varphi(x^k)\rangle + \langle x^k - x^{k+1}, t_k\nabla f(x^k)\rangle && (18)\\ &:= s_1 + cs_2 + s_3. && (19)\end{aligned}$$

By equation (15) $s_1 \leq 0$ holds, and by Lemma 1 we have

$$s_2 := \langle x^* - x^{k+1}, \nabla\varphi(x^{k+1}) - \nabla\varphi(x^k)\rangle = B_\varphi(x^*, x^k) - B_\varphi(x^*, x^{k+1}) - B_\varphi(x^{k+1}, x^k).$$

Furthermore

$$\begin{aligned} s_3 = \langle x^k - x^{k+1}, t_k \nabla f(x^k)\rangle &= t_k \langle \nabla\phi(Ax^k) - \nabla\phi(b), Ax^k - Ax^{k+1}\rangle \\ &\overset{\text{Lem. 1}}{=} t_k (B_\phi(Ax^{k+1}, Ax^k) + B_\phi(Ax^k, b) - B_\phi(Ax^{k+1}, b)). \end{aligned}$$

Summarizing

$$\begin{aligned} t_k(f(x^k) - f(x^*)) &\leq cB_\varphi(x^*, x^k) - cB_\varphi(x^*, x^{k+1}) \\ &+ \underbrace{t_k B_\phi(Ax^{k+1}, Ax^k) - cB_\varphi(x^{k+1}, x^k)}_{\leq 0, \text{Ass.}(b)} + \underbrace{t_k B_\phi(Ax^k, b)}_{= t_k f(x^k)} - t_k B_\phi(Ax^{k+1}, b)) \ , \end{aligned}$$

gives

$$t_k(f(x^{k+1}) - f(x^*)) \leq cB_\varphi(x^*, x^k) - cB_\varphi(x^*, x^{k+1}).$$

Summing over k yields

$$\min_{0 \leq k \leq \kappa} f(x^{k+1}) - f(x^*) \leq \frac{cB_\varphi(x^*, x^0) - cB_\varphi(x^*, x^{\kappa+1})}{t(\kappa+1)} \leq \frac{cB_\varphi(x^*, x^0)}{t(\kappa+1)}.$$

(c) By Lemma 1 we have

$$\begin{aligned} \langle \nabla\phi(Ax^k) - \nabla\phi(b), Ax - Ax^k\rangle &= B_\phi(Ax, b) - B_\phi(Ax, Ax^k) - B_\phi(Ax^k, b) \\ &= f(x) - f(x^k) - B_\phi(Ax, Ax^k) \ . \end{aligned}$$

Thus

$$x^{k+1} = \text{argmin}_{x \in X} f(x) + \underbrace{\frac{c}{t_k} B_\varphi(x, x^k) - B_\phi(Ax, Ax^k)}_{:= f_k(x)}, \tag{20}$$

where $f_k(x) \geq 0$ due to Assumption A., part (b) and $f_k(x^k) = 0$. Consequently, algorithm (10) minimizes an upper bound on f, in analogy to the classical gradient method. Now,

$$f(x^{k+1}) + f_k(x^{k+1}) \leq f(x^k) + f_k(x^k) = f(x^k)$$

follows and

$$f(x^k) - f(x^{k+1}) \geq f_k(x^{k+1}) \geq 0 \ .$$

Hence, the sequence $\{f(x^k)\}_k$ is decreasing and bounded from below by 0. Statement (c) then follows by standard arguments. □

2.4 Application: Multiplicative Updates

It is well-known that multiplicative updates as e.g. employed by the exponential gradient method [4,15], typically lead to faster convergence if the solution x^* of the optimization problem is sparse. As discussed in Section 2.1 the choice

$$\varphi_1(x) = \langle x, \log x \rangle, \quad x \in \mathbb{R}^n_+ \tag{21}$$

and $\phi_1(x) = \langle x, \log(x) \rangle, x \in \mathbb{R}^m_+$, leads to the update rule (3) of SMART, since $B_{\varphi_1}(x, y) = KL(x, y)$ and $f(x) = B_{\phi_1}(Ax, b) = KL(Ax, y)$. For this particular choice we obtain $c_1 = 1$, for matrices A with columns that sum up to one, compare Appendix, Prop. 2.

To include an upper bound on feasible points x, often known in applications (e.g. $x \in [0, 1]^n$), we additionally consider the generalization of the Fermi-Dirac entropy

$$\varphi_2(x) = \langle x - l, \log(x - l) \rangle + \langle u - x, \log(u - x) \rangle, \quad x \in X = [l, u],\ l < u. \tag{22}$$

A simple computation shows $B_{\varphi_2}(x, y) = KL(x - l, y - l) + KL(u - x, u - y)$. With B_{φ_2} and $f(x) = B_{\phi_2}(x, y) = KL(Ax, b)$, we obtain again $c_2 = 1$, compare Appendix, Prop. 3. This choice leads to the following algorithm that we call **bounded-SMART**

$$\frac{(x^{k+1} - l)_j}{(u - x^{k+1})_j} = \frac{(x^k - l)_j}{(u - x^k)_j} \prod_{i=1}^{m} \left(\frac{b_i}{\langle A_{i,\bullet}, x^k \rangle} \right)^{t_k A_{ij}}. \tag{23}$$

Proposition 1 below provides convergence rates for the multiplicative updates (3) and (23). The proof of the following preparatory Lemma is given in the Appendix.

Lemma 2. *For a minimizer $x^* \in X^*$ and some arbitrary starting point $x^0 \in \operatorname{int} X$ with $x^0_{\min} := \min_{i \in [n]} x^0_i$, we have*

$$B_{\varphi_i}(x^*, x^0) \ \leq \ \begin{cases} R(\log R - \log x^0_{\min} - 1) + \|x^0\|_1, & i = 1,\ X = \mathbb{R}^n_+ \\ \log n, & i = 1,\ X = \Delta_n \\ 2R(R - \log x^0_{\min}), & i = 2,\ X = [l, u] \end{cases} \tag{24}$$

for some sufficiently large $R > 0$ such that $\|x^*\|_1 \leq R$. In the case of $\varphi = \varphi_2$ and $X = [l, u]$, we have $R = \|u - l\|_1$.

Proposition 1. *Algorithms* (3) *and* (23) *converge for $t_k = 1$ with rate*

$$\min_{0 \leq k \leq \kappa} f(x^k) - f(x^*) \leq \frac{c(R)}{\kappa},$$

with $c(R)$ given by (24)*, for any $x^0 \in \operatorname{int} X$ and all $\kappa \geq 0$.*

Proof. Propositions 2 and 3 in the Appendix establish $c = 1$ in both cases, in the context of Assumption A, part (b). Parameter t_k can be fixed to 1. Together with parts (a) and (c), Theorem 2 then yields the assertion. □

2.5 B-SMART: An Alternative to Nonnegative Least Squares and ℓ_1–Regression

We briefly relate our approach to the more established objective functions

$$\min_{x\geq 0} \|Ax-b\|^2 \qquad \text{and} \qquad \min_{x\geq 0} \|Ax-b\|_1. \tag{25}$$

The nonnegative least-squares approach on the l.h.s. corresponds to the special case $\varphi_3(x) = \phi_3(x) = \frac{1}{2}\|x\|^2$, cf. (7), (8). Iteration (10) reads (up to a constant)

$$x^{k+1} = \operatorname{argmin}_{x\in X} \langle Ax^k - b, A(x-x^k)\rangle + c\|x-x^k\|^2, \tag{26}$$

with $c = \|A\|_2^2 = \lambda_{\max}(A^\top A)$ for Assumption A, part (b), to hold. While directly tackling the normal equations corresponding to the l.h.s. of (25) is known to be ill-conditioned, the surrogate (right-most term) in (26) provides only a poor approximation of the objective. The large weight c entails only small steps, in addition to the need to take non-smooth projections onto X into account. By contrast, $c = 1$ suffices for both cases (21) and (22), and the feasible set X is taken implicitly into account by closed-form iterative updates.

Adopting a probabilistic viewpoint, **non-negative least-squares** may be critized because the residuals $(Ax-b)_i^2$ do *not* follow a Gaussian distribution. Rather than rectifying this for specific applications (e.g. by a Poisson model in connection with tomography), our approach is additionally motivated by recent results of compressed sensing for non-negative sensing matrices, corresponding to sparse expander graphs with constant column sums (cf., e.g., [6]): $\mathbb{1}^\top A = d\mathbb{1}$, $d > 0$. As a consequence, we have $\|d^{-1}Ax\|_1 = \|x\|_1 = \|b\|_1$ for consistent systems $Ax = b$, that is $x, b \in \Delta_n$, up to a common scale factor. This suggests to adopt the distance $KL(Ax, b)$ in the inconsistent case (noisy measurements b), that is more natural for comparing points in the simplex Δ_n. Applying Jensen's inequality, we then get

$$\begin{aligned} KL(Ax,b) &\leq \log\Big(\sum_i \frac{(Ax)_i^2}{b_i}\Big) = \log\Big(\sum_i \frac{(Ax)_i^2}{b_i} + \underbrace{\|b\|_1 - 2\|Ax\|_1 + 1}_{=0}\Big) \\ &= \log\Big(\sum_i \frac{(Ax-b)_i^2}{b_i} + 1\Big) = \log\Big(1 + \big\langle Ax-b, \operatorname{Diag}(b)^{-1}(Ax-b)\big\rangle\Big). \end{aligned} \tag{27}$$

This relates in view of non-negative least-squares our objective to (the logarithm of) a *scaled* squared Euclidean objective, which is known as the χ^2-distance that provides a first-order expansion of the KL-distance at b [11].

The ℓ_1**-regression objective** in (25), suggested e.g. by [9], may be considered as total variation distance $d_{\mathrm{TV}}(Ax, b) = \sum_i |(Ax-b)_i|$, again from the viewpoint of discrete probability distributions. Our objective upper-bounds this distance, $\frac{1}{2}KL \geq d_{\mathrm{TV}}^2$, as shown in [16], hence minimizes the total variation as well. On the other hand, unlike the *residuals* $(Ax-b)_i$ are known to be sparse (cf. [9]) (rather than x), considering the KL distance seems more appropriate.

Summing up, there are good reasons to consider and study (2) as objective for a range of non-negative compressed sensing scenarios.

3 F-SMART: Towards an Optimal Nonlinear Proj. Gradient Method

Above we showed that SMART and its bounded version converge with rate $O(k^{-1})$. We believe that it should be possible to design an "optimal" entropic gradient method in the sense of [20] with rate $O(k^{-2})$. An elaboration is beyond the scope of the present conference contribution. We therefore confine ourselves to specifying below the algorithm and to providing empirical evidence supporting our conjecture in Section 4.

Similar to Alg. 1 in [23], we suggest the following iteration called **F(ast)-SMART 1**,

$$y^k = (1-\theta_k)x^k + \theta_k z^k \tag{28a}$$

$$z^{k+1} = \operatorname{argmin}_{x\in X}\langle \nabla f(y^k), x - y^k\rangle + c\,\theta_k B_\varphi(x, z^k) \tag{28b}$$

$$x^{k+1} = (1-\theta_k)x^k + \theta_k z^{k+1}, \tag{28c}$$

where $x^0 = z^0 \in \operatorname{int}(\operatorname{dom}\varphi)$ and $\theta_k \in (0,1]$ satisfies

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2} \leq \frac{1}{\theta_k^2}. \tag{29}$$

Additionally, similar to FISTA [5], we suggest the following scheme called **F(ast)-SMART 2**,

$$x^k = \operatorname{argmin}_{x\in X} B_\varphi(x, y^{k-1}) + \langle \nabla f(y^{k-1}), x - y^{k-1}\rangle \tag{30a}$$

$$v^k = \Pi_X\Big(x^{k-1} + \frac{1}{\theta_k}(x^k - x^{k-1})\Big) \tag{30b}$$

$$y^k = (1-\theta_{k+1})x^k + \theta_{k+1}v^k, \tag{30c}$$

where $x^0 = y^0 = v^0 \in \operatorname{int}(\operatorname{dom}\varphi)$ and θ_k satisfies again (29).

Numerical evidence for convergence and the rate of both F-SMART variants is provided in the next section.

4 Experiments and Discussion

In this section we illustrate the performance of BSMART (10) compared to FISTA [5]. BSMART includes the *SMART* scheme for $\varphi_1(x)$ (21) and *b(ounded)-SMART* for $\varphi_2(x)$ (22) as special cases. In the following SMART (3), FSMART1 (28) and FSMART2 (30) will minimize $f(x) = KL(Ax, b)$ over $X = R_+^n$, while b(ounded)-SMART, b(ounded)-FSMART1 and b(ounded)-FSMART2 minimizes $f(x) = KL(Ax, b)$ over $X = [0,1]^n$. Cf. the discussion of Eq. (27), FISTA will be applied to $f(x) = 0.5\|Ax - b\|^2$ and $f(x) = 0.5\langle Ax - b, \operatorname{Diag}(b)^{-1}(Ax - b)\rangle$ subject to both $X = \mathbb{R}_+^n$ and $X = [0,1]^n$. Matrix A will be scaled so that the every column sums up to one.

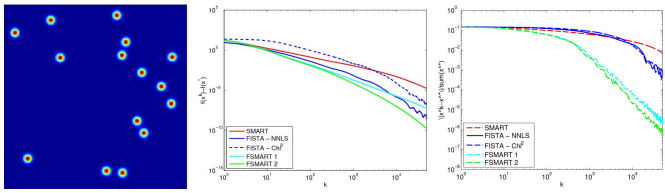


Fig. 1. The first test image consists of 15 particles at random positions (left). Comparison of function value errors $f(x^k) - f(x^*)$ for all algorithms (middle). While BSMART is competitive, the relative error decays faster for FSMART1 and FSMART1 (right).

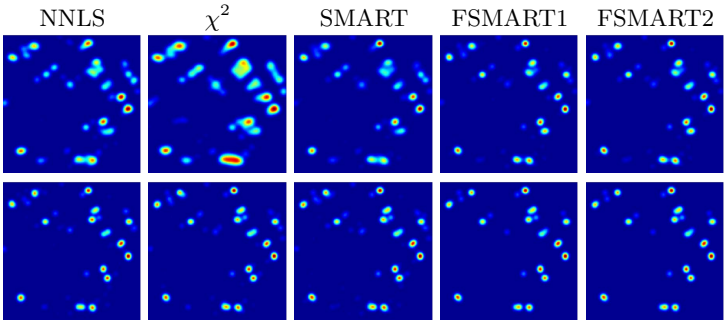


Fig. 2. Reconstructions of 15 particles at random positions at iteration 100 (top row) and at the final iteration (bottom row). The reconstruction is accurate after a (significantly) smaller number of iterations in the case of the KL objective that copes better with an ill-conditioned matrix A.

Test Case 1: Here we consider an infeasible ill-conditioned problem inspired by a real-world application [1]. The original sparse image I, see Fig. 1 left, consists of 15 Gaussian blobs (particles) at random positions in a square. The measurement vector $b \in \mathbb{R}^{200}$ is computed by integrating the particle image exactly along 50×4 lines arranged in 4 fan beams (angles $45^o, 15^o, -15^o, -45^o$). Image I is discretized in 66×66 Gaussian basis functions positioned on a regular grid. The matrix entries A_{ij} equal the line integral of every basis function along every line, thus $A \in \mathbb{R}^{200\times 4356}$ and $A \geq 0$. After scaling $\mathbb{1}^\top A = \mathbb{1}$, and $L_{\chi^2} = 1004.8$, $L_{\mathrm{NNLS}} \approx 53.6$. We underline that *no* nonnegative solution exists which satisfies the constraints $Ax = b$. Additionally we added uniform (non-Gaussian) noise to b. The parameters for FISTA, FSMART1 and FSMART2 are chosen as $\theta_k = 1$, $\theta_{k+1} = 0.5(\sqrt{\theta_{k+1}^4 + 4\theta_{k+1}^2} - \theta_{k+1}^2)$ and satisfy (29), according to [23]. The function value at iteration k of all algorithms is depicted in Fig. 1. The function value for FSMART2 is lower than for FISTA, which is explained by the high values of L_{χ^2} and L_{NNLS}. The decay of $f(x^k) - f(x^*)$ for both FSMART1 and FSMART2 suggests a $O(k^{-2})$ rate, consistently with FISTA, see Fig. 1, middle. The solutions x^* for the three problems considered, $\min_{x\in\mathbb{R}^n_+} KL(Ax, b)$, $\min_{x\in\mathbb{R}^n_+} 0.5\langle Ax-b, \mathrm{Diag}(b)^{-1}(Ax-b)\rangle$ and $\min_{x\in\mathbb{R}^n_+} 0.5\|Ax-b\|^2$, are not known,

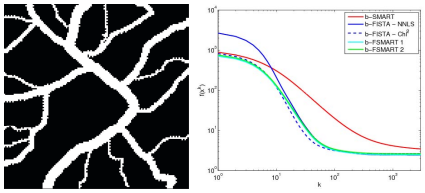


Fig. 3. Original 256×256 binary test image from [2] (left). Comparison of function values for all algorithms and $X = [0, 1]^n$ (right). Again FSMART1 and FSMART2 exhibit an $O(k^{-2})$ rate.

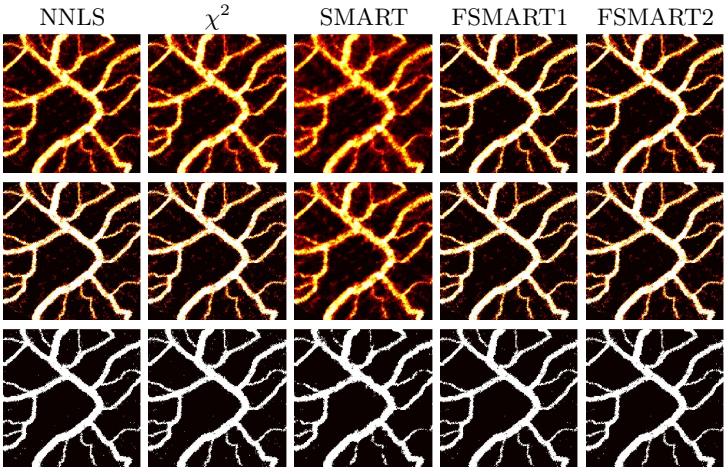


Fig. 4. Reconstructions after 50 iterations (top row) and after 100 iterations (middle row). By replacing at iteration 100 all values above a globally determined threshold with one and the others with zero, we obtain similar results for all algorithms with slightly better and faster reconstructions for FSMART and FISTA.

but we computed an accurate solution via an interior point solver for the KKT conditions. Iteration 100 and the final one are described in Fig. 2. *The reconstructions produced by SMART, FSMART1 and FSMART2 are of better quality even if only few iterations are performed.*

These preliminary computational results indicate that BSMART is sometimes even faster then the proven predicted theoretical rate and FSMART is a promising extension with a high potential for designing fast algorithms for nonnegative data.

Test Case 2: The second 256×256 test image [2] is a vascular system containing larger and smaller vessels, see Fig. 3 (left). We consider 20 projecting directions, although the uncorrupted image binary image is determined by 18 projections

and is unique in $[0,1]^n$. Here $A \in \mathbb{R}^{7240\times 65536}$, $L_{\chi^2} = 5.0948$, $L_{\mathrm{NNLS}} = 12.2308$ and $X = [0,1]^n$. Thus A is better conditioned than the previous one. To vector b we add again 5% uniform (nongaussian) noise. This results in an infeasible problem. Due to the low Lipschitz constant L_{χ^2} we expect a similar behavior of FISTA and FSMART, which is exactly what happens, see Fig. 3 (right) for the decrease of the function values.

Adding the additional information that the image entries are in $[0,1]^n$ leads to a fairly good reconstruction in Fig. 4 within the first iterations. This can be improved by thresholding.

5 Conclusion and Further Work

This paper advocates Bregman functions as objectives for constrained nonnegative compressed sensing problems, together with a corresponding non-quadratic proximation scheme that only requires first-order gradient evaluations of the objective. The attractive properties of this approach concerning both mathematical and algorithmic aspects deserve further study. Our future work therefore will take a closer look on the pros and cons in connection with other established objectives in the field of compressed sensing, as initiated in section 2.5. Furthermore, in view of established optimal first-order methods with $O(k^{-2})$ convergence rate, we will study from a more general mathematical viewpoint surrogate objectives based on non-quadratic proximation that lead to efficient two-step iterations with multiplicative updates, with a focus on the resulting convergence rates.

Appendix

Properties of the Kullback-Leibler Distance. For positive scalars a, b, define $KL(a,b) = a\log(a/b) + b - a$, $KL(0,b) = b$ and $KL(a,0) = +\infty$. The Kullback-Leibler distance can be extended to nonnegative vectors

$$KL(x,y) := \sum_{j=1}^{n} \left(x_j \log\left(\frac{x_j}{y_j}\right) + y_j - x_i \right). \tag{31}$$

It is well known that for all $x, y \geq 0$, we have $KL(x,y) \geq 0$ and $KL(x,y) = 0$ iff $x = y$. Furthermore, by Jensen's inequality, we have (see, e.g., [11, Thm. 2.7.1])

$$\sum_{i=1}^{n} x_i \log\frac{x_i}{y_i} \geq \Big(\sum_{i=1}^{n} x_i\Big) \log\frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} y_i}, \qquad \forall x, y \in \mathbb{R}^n_+. \tag{32}$$

Proposition 2. *For $A \geq 0$ with $1^\top A = 1^\top$, we have*

$$KL(Ax, Ay) \leq KL(x,y), \qquad \forall x, y \in \mathbb{R}^n_+. \tag{33}$$

Proof. We compute

$$\begin{aligned} KL(x,y) &= \sum_{j=1}^{n}\Big(x_j\log\frac{x_j}{y_j}+y_j-x_i\Big) = \sum_{j=1}^{n}\underbrace{\sum_{i=1}^{m}A_{ij}}_{=1}\Big(x_j\log\frac{x_j}{y_j}+y_j-x_i\Big) \\ &= \sum_{i=1}^{m}\Big(\sum_{j=1}^{n}A_{ij}x_j\log\frac{A_{ij}x_j}{A_{ij}y_j}+\sum_{j=1}^{n}A_{ij}y_j-\sum_{j=1}^{n}A_{ij}x_j\Big) \\ &\overset{\text{Eq. (32)}}{\geq} \sum_{i=1}^{m}\Big[\Big(\sum_{j=1}^{n}A_{ij}x_j\Big)\log\frac{\sum_{j=1}^{n}A_{ij}x_j}{\sum_{j=1}^{n}A_{ij}y_j}+\sum_{j=1}^{n}A_{ij}y_j-\sum_{j=1}^{n}A_{ij}x_j\Big] \\ &= KL(Ax,Ay). \end{aligned}$$

Lemma 3. *For any $x, y \geq 0$, with $x \geq t$ and $y \geq t$, we have $KL(x-t, y-t) \geq KL(x,y)$.*

Proof. Let $g(t) = KL(x-t, x-t)$. Then $g'(t) = \frac{x-t}{y-t} - 1 - \log\left(\frac{x-t}{y-t}\right) \geq 0$. Thus $g(t) \geq g(0)$.

This immediately implies

Proposition 3. *For $A \geq 0$ with $1^\top A = 1^\top$ and $x, y \in [l,u]$, we have*

$$\begin{aligned} KL(Ax,Ay) &\leq KL(Ax-Al, Ay-Al) + KL(Au-Ax, Au-Ay) &(34)\\ &\leq KL(x-l, y-l) + KL(u-x, u-y)\ . &(35)\end{aligned}$$

Proof of Lemma 2

Proof. In the case $X = \mathbb{R}^n_+$, we may assume $\|x\|_1 \leq R$ for some sufficiently large $R > 0$, due to Assumption A, part (c). Hence

$$\begin{aligned} B_{\varphi_1}(x^*,x^0) &= \sum_i\big(x_i^*\log\frac{x_i^*}{x_i^0}+x_i^0-x_i^*\big) = \|x^*\|_1\sum_i\frac{x_i^*}{\|x^*\|_1}\log\frac{x_i^*}{x_i^0}+\sum_i(x_i^0-x_i^*) \\ &= \|x^*\|_1\sum_i\frac{x_i^*}{\|x^*\|_1}\big(\log\frac{x_i^*}{\|x^*\|_1}+\log\|x^*\|_1-\log x^0_{\min}\big)+\sum_i(x_i^0-x_i^*) \\ &\leq \|x^*\|_1(\log\|x^*\|_1-\log x^0_{\min}-1)+\|x^0\|_1 \leq R(\log R-\log x^0_{\min}-1)+\|x^0\|_1 \end{aligned}$$

In the case $X = \Delta_n$, we have $R = 1$ and may choose $x^0 = n^{-1}\mathbb{1}$. In the case $X = [l,u]$, the last two summands in (31) cancel. A similar computation then yields

$$\begin{aligned} B_{\varphi_2}(x^*,x^0) &\leq \|x^*-l\|_1(\log\|x^*-l\|_1-\log x^0_{\min})+\|u-x^*\|_1(\log\|u-x^*\|_1-\log x^0_{\min}) \\ &\leq 2\|u-l\|_1(\log\|u-l\|_1-\log x^0_{\min}). \end{aligned}$$

□

References

1. Atkinson, C., Soria, J.: An efficient simultaneous reconstruction technique for tomographic particle image velocimetry. Experiments in Fluids 47(4), 553–568 (2009)
2. Batenburg, K.J.: A network flow algorithm for reconstructing binary images from discrete x-rays. J. Math. Imaging Vis. 27(2), 175–191 (2007)
3. Bauschke, H.H., Borwein, J.M.: Legendre Functions and the Method of Random Bregman Projections. J. Convex Analysis 4(1), 27–67 (1997)
4. Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Operations Research Letters 31(3), 167–175 (2003)
5. Beck, A., Teboulle, M.: A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. SIAM J. Img. Sci. 2, 183–202 (2009)
6. Berinde, R., Gilbert, A.C., Indyk, P., Karloff, H., Strauss, M.J.: Combining Geometry and Combinatorics: A Unified Approach to Sparse Signal Recovery. CoRR (2008); Preprint arXiv:0804.4666
7. Byrne, C.L.: Iterative image reconstruction algorithms based on cross-entropy minimization. IEEE Transactions on Image Processing, 96–103 (1993)
8. Candès, E.: Compressive sampling. In: Int. Congress of Math, Madrid, Spain, vol. 3 (2006)
9. Candes, E., Rudelson, M., Tao, T., Vershynin, R.: Error Correcting via Linear Programming. In: 46th Ann. IEEE Symp. Found. Computer Science (FOCS 2005), pp. 295–308 (2005)
10. Chen, G., Teboulle, M.: Convergence analysis of a proximal-like minimization algorithm using Bregman functions. SIAM Journal on Optimization 3(3), 538–543 (1993)
11. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons (1991)
12. Donoho, D.: Compressed Sensing. IEEE Trans. Information Theory 52, 1289–1306 (2006)
13. Donoho, D.L., Tanner, J.: Counting the Faces of Randomly-Projected Hypercubes and Orthants, with Applications. Discrete & Computational Geometry 43(3), 522–541 (2010)
14. Eckstein, J.: Nonlinear Proximal Point Algorithms using Bregman Functions, with Applications to Convex Programming. Math. Oper. Res. 18(1), 202–226 (1993)
15. Kivinen, J., Warmuth, M.: Exponentiated Gradient versus Gradient Descent for Linear Predictors. Inform. Comput. 132, 1–63 (1997)
16. Kullback, S.: A Lower Bound for Discrimination Information in Terms of Variation. IEEE Trans. Inf. Theory 13(1), 126–127 (1967)
17. Mangasarian, O.L., Recht, B.: Probability of Unique Integer Solution to a System of Linear Equations. European Journal of Operational Research 214(1), 27–30 (2011)
18. Nesterov, Y.: A method of solving a convex programming problem with convergence rate O (1/k2). Soviet Mathematics Doklady 27(2), 372–376 (1983)
19. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. 103(1), 127–152 (2005)
20. Nesterov, Y.E., Nemirovski, A.S.: Interior-Point Polynomial Algorithms in Convex Programming (Studies in Applied and Numerical Mathematics). Society for Industrial Mathematics (1994)

21. Petra, S., Schnörr, C., Schröder, A.: Critical Parameter Values and Reconstruction Properties of Discrete Tomography: Application to Experimental Fluid Dynamics. Fundamenta Informaticae (2013); To appear and arXiv:1209.4316 (2012)
22. Slawski, M., Hein, M.: Non-negative least squares for sparse recovery in the presence of noise. In: Proc. SPARS (2011)
23. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. Submitted to SIAM J. Control Optim. (2008)
24. Wang, M., Xu, W., Tang, A.: A Unique "Nonnegative" Solution to an Underdetermined System: From Vectors to Matrices. IEEE Transactions on Signal Processing 59(3), 1007–1016 (2011)

Epigraphical Projection for Solving Least Squares Anscombe Transformed Constrained Optimization Problems

Stanislav Harizanov[1], Jean-Christophe Pesquet[2], and Gabriele Steidl[1]

[1] Department of Mathematics, University of Kaiserslautern, Germany
[2] Laboratoire d'Informatique Gaspard Monge, Université Paris-Est, France

Abstract. This paper deals with the restoration of images corrupted by a non-invertible or ill-conditioned linear transform and Poisson noise. Poisson data typically occur in imaging processes where the images are obtained by counting particles, e.g., photons, that hit the image support. By using the Anscombe transform, the Poisson noise can be approximated by an additive Gaussian noise with zero mean and unit variance. Then, the least squares difference between the Anscombe transformed corrupted image and the original image can be estimated by the number of observations. We use this information by considering an Anscombe transformed constrained model to restore the image. The advantage with respect to corresponding penalized approaches lies in the existence of a simple model for parameter estimation. We solve the constrained minimization problem by applying a primal-dual algorithm together with a projection onto the epigraph of a convex function related to the Anscombe transform. We show that this epigraphical projection can be efficiently computed by Newton's methods with an appropriate initialization. Numerical examples demonstrate the good performance of our approach, in particular, its close behaviour with respect to the I-divergence constrained model.

1 Introduction

The Poisson distribution exhibits a mean/variance relationship. This mean/variance dependence can be reduced by using variance-stabilizing transformations (VST), one of which is the *Anscombe transform* [1] defined as

$$T\colon [0,+\infty)^n \to (0,+\infty)^n\colon v = (v_i)_{1\le i\le n} \mapsto 2\left(\sqrt{v_i + \frac{3}{8}}\right)_{1\le i\le n}.$$

It transforms Poisson noise to approximately Gaussian noise with zero-mean and unit variance (if the variance of the Poisson noise is large enough). The Anscombe transform has been employed in order to solve inverse problems where one wants to recover an original signal $\overline{u} \in [0,+\infty)^m$ from observations

$$f = \mathcal{P}(H\overline{u}),$$

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 125–136, 2013.

where $\mathcal{P}$ denotes an independent Poisson noise corruption process and $H \in [0,+\infty)^{n\times m}$ is a linear degradation operator, e.g. a blur. Note that we consider images of size $M \times N$ columnwise reshaped as vectors of length $m = MN$.

In this context, one of the possible uses of the Anscombe transform is *i*) to transform the degraded observations f, *ii*) to apply a data recovery technique which is valid for an additive white zero-mean Gaussian model and *iii*) to apply an inverse transform to the so-recovered signal [7] (see also [18] for more recent developments). Note that this method appears mainly to be well-founded for denoising problems. When a linear degradation operator H is present, a better approach consists of adopting a variational framework [8,13] where one minimizes a data fidelity term

$$u \mapsto \|T(Hu) - T(f)\|_2^2 \tag{1}$$

penalized by a (sum of) regularization term(s) serving to incorporate prior information about the sought signal $\overline{u}$. The approach is also closely related to a Maximum A Posteriori (MAP) estimate, where the function in (1) is substituted for the neg-log-likelihood of the Poisson noise, i.e., the I-divergence (generalized Kullback-Leibler divergence)

$$u \mapsto D(f,Hu) := \begin{cases} \langle \mathbf{1}_n, f\log\frac{f}{Hu} - f + Hu\rangle & \text{if } Hu > 0,\\ +\infty & \text{otherwise,}\end{cases}$$

where $\langle\cdot\,,\cdot\rangle$ denotes the standard Euclidean inner product and $\mathbf{1}_n$ denotes the vector consisting of n entries equal to 1 (see [14,21]). One of the drawbacks of these penalized methods is that multiplicative constants weighting the regularization terms (the so-called regularization parameters) need to be set carefully, which may be a difficult task.

A way of circumventing this problem consists of adopting a constrained approach instead of a regularized one, by imposing that

$$\|T(Hu) - T(f)\|_2^2 \leq \tau \tag{2}$$

where $\tau \in [0,+\infty)$. Based on the statistical properties of the Anscombe transform and the law of large numbers, a consistent choice for the above bound is $\tau = n$, when the number of observations n is large. In this work, we will investigate such an approach by solving the following problem:

$$\underset{u\in C}{\text{minimize}}\ \ \Phi(Lu) \quad \text{subject to} \quad \|T(Hu) - T(f)\|_2^2 \leq \tau, \tag{3}$$

where C is a nonempty closed convex subset of $[0,+\infty)^m$, $L \in \mathbb{R}^{q\times m}$, and $\Phi\colon \mathbb{R}^q \to (-\infty,+\infty]$ is a proper, lower-semicontinuous, convex function. A typical choice for C is the nonnegative orthant of $\mathbb{R}^m$. The classical Total Variation objective function [20] is obtained, as a special case, when Φ is an $\ell_{2,1}$ norm and L corresponds to a discrete gradient operator. Constrained models based on the I-divergence have been considered in [5,22], where in the second paper special attention was paid to the relation between the parameters of the constrained

and the penalized problem via discrepancy principles. Note that recently penalized versus constrained problems in a rather general form were handled in [2]. In [9], the I-divergence constraint was replaced through a polyhedral approximation technique and an epigraphical projection method was applied to solve the problem. In this work, we will also take advantage of an epigraphical projection approach to solve the Anscombe constrained model (3) and we will show that the required epigraphical projections can be easily determined in this context.

The structure of this paper is as follows: Section 2 recalls the notation. In Section 3 we determine the epigraphical projection for a function related to constraint (2) which plays a central role in the primal dual algorithms established in Section 4. In particular, we provide a good starting point for the involved Newton method. Numerical examples are presented in Section 5 emphasizing the good approximation of the I-divergence constrained approach achieved by our Anscombe constrained model. Finally, a summary of our contribution and some conclusions are given in Section 6.

2 Notation

Let $\Gamma_0(\mathbb{R}^n)$ denote the set of proper, lower-semicontinuous, convex functions mapping from $\mathbb{R}^n$ to $(-\infty, +\infty]$. The *epigraph* of $\varphi \in \Gamma_0(\mathbb{R}^n)$ is the nonempty, closed, convex subset of $\mathbb{R}^{n+1}$ defined as

$$\operatorname{epi} \varphi := \{(v, \zeta) \in \mathbb{R}^n \times \mathbb{R} : \varphi(v) \leq \zeta\}.$$

For a nonempty, closed, convex set $C \subset \mathbb{R}^m$ we denote by $\iota_C \in \Gamma_0(\mathbb{R}^m)$ its *indicator function*

$$\iota_C(u) := \begin{cases} 0 & \text{if } u \in C, \\ +\infty & \text{otherwise,} \end{cases}$$

and by P_C the *orthogonal projector* onto C. Beyond epigraphs of functions from $\Gamma_0(\mathbb{R}^n)$ we will consider the half-space $V_\tau := \{\zeta \in \mathbb{R}^n : \langle \mathbf{1}_n, \zeta \rangle \leq \tau\}$. Using this notation and defining, for every $i \in \{1, \ldots, n\}$,

$$\varphi_i \colon [0, +\infty) \to [0, +\infty) \colon s \mapsto \left(2\sqrt{s} - \big(T(f)\big)_i\right)^2,$$

problem (3) can be rewritten as

$$\underset{(u,\zeta)\in\mathbb{R}^m\times\mathbb{R}^n}{\text{minimize}} \ \iota_C(u) + \Phi(Lu) + \sum_{i=1}^{n} \iota_{\operatorname{epi}\varphi_i}\big((Hu)_i + \frac{3}{8}, \zeta_i\big) + \iota_{V_\tau}(\zeta). \tag{4}$$

Now one can choose a primal-dual splitting algorithm as those proposed in [4,6,11,12,23] to solve this problem. One step in all these algorithms consists of the orthogonal projections onto the epigraphs of φ_i for all $i \in \{1, \ldots, n\}$ which is the topic of the next section.

3 Epigraphical Projection

In this section, we deal with the projection onto the epigraph of the function $\varphi \in \Gamma_0(\mathbb{R})$ defined as

$$\varphi(s) := \begin{cases} (2\sqrt{s} - z)^2 & \text{if } s \geq 0, \\ +\infty & \text{otherwise,} \end{cases} \tag{5}$$

where $z > 0$, see Fig. 1.

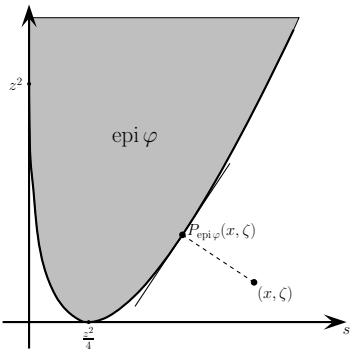


Fig. 1. The epigraph of φ for $z = 3$ and the epigraphical projection $P_{\text{epi}\,\varphi}(x,\zeta)$ of some point (x,ζ)

Proposition 1. *Let φ be defined by (5) with $z > 0$. Then the epigraphical projection of $(x,\zeta) \in \mathbb{R}^2$ is given by*

$$P_{\text{epi}\,\varphi}(x,\zeta) = \begin{cases} (\max\{x,0\},\zeta) & \text{if } \varphi(\max\{x,0\}) \leq \zeta, \\ \left(\left(\frac{t_+ + z}{2}\right)^2, t_+^2\right) & \text{if } \varphi(\max\{x,0\}) > \zeta \text{ and } 4x \geq z^2, \\ \left(\left(\frac{t_- + z}{2}\right)^2, t_-^2\right) & \text{if } \varphi(\max\{x,0\}) > \zeta \text{ and } 4x < z^2, \end{cases}$$

where t_+, resp. t_- is the unique root in $[0,+\infty)$, resp. in $(-z,0)$ of the cubic polynomial

$$p\colon t \mapsto 17t^3 + 3zt^2 + (3z^2 - 16\zeta - 4x)t + z(z^2 - 4x). \tag{6}$$

Proof. The function φ fulfills $\varphi(0) = z^2$ and

$$\varphi'(s) = 4 - \frac{2z}{\sqrt{s}} \begin{cases} < 0 & \text{if } 0 < s < \frac{z^2}{4}, \\ = 0 & \text{if } s = \frac{z^2}{4}, \\ > 0 & \text{if } s > \frac{z^2}{4} \end{cases}$$

and therefore $\lim_{\substack{s\to 0\\ s>0}} \varphi'(s) = -\infty$. Thus, if $x \le 0$ and $\zeta \ge z^2$, then $P_{\operatorname{epi}\varphi}(x,\zeta) = (0,\zeta)$. In addition, if $(x,\zeta) \in \operatorname{epi}\varphi$, then $P_{\operatorname{epi}\varphi}(x,\zeta) = (x,\zeta)$.

We consider the remaining cases when $(\max\{x,0\},\zeta) \notin \operatorname{epi}\varphi$. The tangent vector of the curve associated with the graph of φ reads $(1,\varphi'(s))$, $s>0$. The uniquely determined orthogonal projection $(\hat{x},\hat{\zeta}) := P_{\operatorname{epi}\varphi}(x,\zeta)$ has to satisfy

$$\left(\begin{pmatrix} x \\ \zeta \end{pmatrix} - \begin{pmatrix} \hat{x} \\ \hat{\zeta} \end{pmatrix}\right) \perp \begin{pmatrix} 1 \\ \varphi'(\hat{x}) \end{pmatrix} \quad \text{and} \quad \hat{\zeta} = \varphi(\hat{x}), \quad \hat{x} > 0 \tag{7}$$

which leads to

$$0 = (x-\hat{x})\sqrt{\hat{x}} + 2\left(\zeta - (2\sqrt{\hat{x}} - z)^2\right)(2\sqrt{\hat{x}} - z), \quad \hat{x} > 0.$$

Substituting $\hat{t} := 2\sqrt{\hat{x}} - z > -z$, this can be rewritten as

$$0 = 17\hat{t}^3 + 3z\hat{t}^2 + (3z^2 - 16\zeta - 4x)\hat{t} + z(z^2 - 4x) = p(\hat{t}), \quad \hat{t} > -z.$$

Conversely, if $\hat{t} > -z$ is a root of the polynomial p in (6), then $\hat{x} = \left(\frac{\hat{t}+z}{2}\right)^2$ fulfills (7). When $x \ge z^2/4$ then also $\hat{x} \ge z^2/4$ (see Fig. 1), thus we are interested in the restriction of φ to $[z^2/4, +\infty)$ i.e., the nonnegative roots of p. The restriction of φ to $[z^2/4, +\infty)$ is convex, monotonically increasing, and $(x,\zeta) \notin \operatorname{epi}\,\varphi$. Hence, there is a unique point $(\hat{x},\hat{\zeta})$ on its graph that satisfies (7), i.e., p has a unique root in $[0,+\infty)$. Analogously, when $x < z^2/4$ then also $\hat{x} < z^2/4$, thus we are interested in the restriction of φ to $[0, z^2/4)$ i.e., the roots of p in the interval $(-z, 0)$. The restriction of φ to $[0, z^2/4)$ is convex and monotonically decreasing, and the uniqueness of the root follows by the same arguments. Finally, it can be noticed that $\hat{\zeta} = \varphi\left(\left(\frac{\hat{t}+z}{2}\right)^2\right) = \hat{t}^2$ since $\hat{t} > -z$, which completes the proof. □

The next proposition states that the root t_+, resp. t_-, of polynomial p can be computed efficiently by Newton's method with initial value $t_0 := 2\sqrt{\max\{x,0\}} - z$. Indeed, we have seen in our numerical examples that t_0 is a very good starting point.

Proposition 2. *Let* $(\max\{x,0\},\zeta) \notin \operatorname{epi}\varphi$ *and* $t_0 := 2\sqrt{\max\{x,0\}} - z$. *Let the polynomial* p *be defined by* (6). *Then, after a finite number of steps, the Newton method for finding a zero of* p *with initial value* t_0 *converges monotonically to the root* t_+ *if* $4x \ge z^2$, *resp.,* t_- *if* $4x < z^2$.

Proof. 1. First we show that

i) $p(t_0)p(0) \le 0$,
ii) $p'(t_0) > 0$,

where equality in i) holds true iff $4x = z^2$.

If $x < 0$, then $t_0 = -z$ and consequently, since $(0,\zeta) \notin \operatorname{epi}\varphi$, i.e., $z^2 > \zeta$, we obtain: $p(0) = z(z^2 - 4x) > 0$ and $p(t_0) = p(-z) = 16z(\zeta - z^2) < 0$, which proves i). Further, since $p'\colon t \mapsto 51t^2 + 6zt + 3z^2 - 4x - 16\zeta$, we obtain

$$p'(t_0) = p'(-z) = 48z^2 - 16\zeta - 4x \ge 16(4z^2 - \zeta) > 0.$$

If $x \geq 0$, then $t_0 = 2\sqrt{x} - z$. Consequently, we have $p(0) = -z(z + 2\sqrt{x})t_0$ and

$$p(t_0) = 17t_0^3 + 2zt_0^2 + (3z^2 - 16\zeta - 4x)t_0 + (t_0^2 + z^2 - 4x)z = 16t_0(t_0^2 - \zeta).$$

Since $(x, \zeta) \notin \operatorname{epi} \varphi$, i.e., $t_0^2 > \zeta$ we conclude that i) holds true. Finally ii) follows by

$$p'(t_0) = 51t_0^2 + 6zt_0 + 3z^2 - 16\zeta - 4x = 8(6t_0^2 - 2\zeta + x) = 16(t_0^2 - \zeta) + 8(4t_0^2 + x) > 0.$$

Since, in both cases, $p(t_0) \neq 0$, equality arises in i) iff $p(0) = 0$ i.e. $4x = z^2$.

2. The following result is well-known, see, e.g., [15, Theorem 18.3]: Newton's method for finding the unique root of a differentiable, convex, strictly increasing function on an interval converges monotonically if we start at the right endpoint of the interval. There is an analogous result for concave functions.

3. Since $p'' \colon t \mapsto 6(17t + z)$, p is convex on $[-\frac{z}{17}, +\infty)$ and concave on $(-\infty, -\frac{z}{17}]$.

3.1 Let $t_0 > 0$ which implies $4x - z^2 > 0$. Then $p(0) < 0$ and, according to Part 1i), $p(t_0) > 0$. Hence, by Proposition 1, t_+ is the unique root of p in $(0, t_0)$. Since p is continuous, $p(t_+) = 0$ and $p(t_0) > 0$, we necessarily have $p'(t_+) \geq 0$ (otherwise there would exist another root of p on (t_+, t_0)). Thus, since p is strictly convex, it is strictly monotone increasing on $[t_+, t_0]$ and we can invoke the argument in Part 2 of the proof.

3.2 Let $t_0 < 0$ which implies $4x - z^2 < 0$. Then, $p(0) > 0$ and $p(t_0) < 0$ and by Proposition 1 we know that $t_- \in (t_0, 0)$ is the unique root of p in $(-z, 0)$. If $t_- \leq -\frac{z}{17}$, then we are done by similar arguments as in 3.1 for concave functions. It remains to study the case when $t_- > -\frac{z}{17}$.

If $t_0 > -\frac{z}{17}$, then we know by the strict convexity of p on $[t_0, +\infty)$ that p' is strictly increasing on this interval and by Part 1ii) we further have $p'(t) > p'(t_0) > 0$ for every $t > t_0$. Thus, p itself is strictly increasing on $[t_0, +\infty)$. Consequently, one Newton step with initialization t_0 generates $t_1 = t_0 - p(t_0)/p'(t_0)$ and the convexity inequality $p(t_1) \geq p(t_0) + p'(t_0)(t_1 - t_0)$ shows that $p(t_1) \geq 0$. We thus are in the setting of Part 2 and the method converges monotonically.

Finally, let $t_0 \leq -\frac{z}{17}$ so that

$$4x \leq \left(\frac{16}{17}\right)^2 z^2. \tag{8}$$

Since we must have $p(-\frac{z}{17}) < 0$, a straightforward calculation yields

$$p\left(-\frac{z}{17}\right) = \frac{2z}{17}\left(\frac{z^2}{17} + 7z^2 + 8\zeta - 32x\right) < 0 \;\Leftrightarrow\; 8\zeta < 32x - \frac{z^2}{17} - 7z^2. \tag{9}$$

We shall now prove that $p'(t) > 0$ for all $t \in \mathbb{R}$, so that p is strictly increasing. Since p' is a quadratic polynomial with minimimum at $-\frac{z}{17}$, we have only to show that $p'\left(-\frac{z}{17}\right) = 2\left(\frac{24}{17}z^2 - 8\zeta - 2x\right) > 0$. Plugging in (8) and (9), we indeed obtain

$$\frac{1}{2}p'\left(-\frac{z}{17}\right) > \frac{24}{17}z^2 - 34x + \frac{z^2}{17} + 7z^2 > \frac{25}{17}z^2 - \frac{128}{17}z^2 + 7z^2 = \frac{16}{17}z^2 > 0.$$

The sequence $(t_k)_{k\in\mathbb{N}}$ generated by Newton's algorithm is such that there exists $k_0 \in \mathbb{N}\setminus\{0\}$ such that $t_{k_0} \geq t_-$. Otherwise, $(t_k)_{k\in\mathbb{N}}$ would be an increasing sequence which would necessarily converge to t_- and there would exist $k_1 \in \mathbb{N}$ such that p is convex over $[t_{k_1}, +\infty[$. Then, we would have $p(t_{k_1+1}) \geq 0$.

Thus, after a finite number of steps, the algorithm arrives at $t_{k_0} \geq t_-$ and we can apply Part 2 of the proof.

3.3 Let $t_0 = 0$ which implies $4z - t^2 = 0$. Then, we get $t_+ = t_0$. □

4 Primal-Dual Algorithms

We can apply the projection onto the epigraph of φ in combination with any primal-dual algorithm proposed in [4,6,11,12,23] or an alternating direction method of multipliers. For example, we use here the **p**rimal-**d**ual **h**ybrid **g**radient algorithm from [6,19] with an extrapolation (**m**odification) of the dual variable which will be designated by PDHGMp. Based on the following reformulation of (4),

$$\begin{aligned} &\underset{(u,\zeta),(v_1,v_2,\eta)}{\text{minimize}} \quad \iota_C(u) + \iota_{V_\tau}(\zeta) + \Phi(v_2) + \sum_{i=1}^{n} \iota_{\operatorname{epi}\varphi_i}(v_{1,i}, \eta_i) \\ &\text{subject to} \quad \left(\begin{array}{c|c} H & 0 \\ L & 0 \\ \hline 0 & I \end{array}\right) \begin{pmatrix} u \\ \zeta \end{pmatrix} + \begin{pmatrix} 3/8 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \eta \end{pmatrix}, \end{aligned} \tag{10}$$

this algorithm reads:

Algorithm 1 (PDHGMp for solving the Anscombe constrained problem)
Initialization: $u^{(0)}, \zeta^{(0)}, (p_j^{(0)})_{1\leq j\leq 3} = (\bar{p}_j^{(0)})_{1\leq j\leq 3}$, $\theta \in (0,1]$, $(\rho,\sigma) \in (0,+\infty)^2$ with $\rho\sigma < 1/\max\{1, \|H^*H + L^*L\|_2\}$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

1. $u^{(k+1)} = P_C\left(u^{(k)} - \sigma\rho\Big(H^*\bar{p}_1^{(k)} + L^*\bar{p}_2^{(k)}\Big)\right)$
2. $\zeta^{(k+1)} = P_{V_\tau}\left(\zeta^{(k)} - \sigma\rho\bar{p}_3^{(k)}\right)$
3. $(v_{1,i}^{(k+1)}, \eta_i^{(k+1)}) = P_{\operatorname{epi}\varphi_i}\left(p_{1,i}^{(k)} + (Hu^{(k+1)})_i + 3/8\,,\, p_{3,i}^{(k)} + \zeta_i^{(k+1)}\right), \quad i = 1, \ldots, n$
4. $v_2^{(k+1)} = \operatorname{prox}_{\sigma^{-1}\Phi}(p_2^{(k)} + Lu^{(k+1)})$
5. $p_1^{(k+1)} = p_1^{(k)} + Hu^{(k+1)} + 3/8 - v_1^{(k+1)}$
6. $p_2^{(k+1)} = p_2^{(k)} + Lu^{(k+1)} - v_2^{(k+1)}$
7. $p_3^{(k+1)} = p_3^{(k)} + \zeta^{(k+1)} - \eta^{(k+1)}$
8. $\bar{p}_j^{(k+1)} = p_j^{(k+1)} + \theta(p_j^{(k+1)} - p_j^{(k)}), \quad j = 1, 2, 3.$

The projection in step 1 is quite simple if C is the nonnegative orthant of $\mathbb{R}^m$, as well as the projection onto the closed half-space V_τ in step 2. Step 3 requires the epigraphical projections discussed in the previous section, Step 4 can be performed by coupled soft shrinkage with threshold σ^{-1} if we use the $\ell_{2,1}$-norm. The other steps can be computed in a straightforward way.

We will compare this algorithm with PDHGMp applied to the I-divergence constrained problem

$$\underset{u\in C}{\text{minimize}}\ \Phi(Lu) \quad \text{subject to} \quad D(f,Hu)\leq \tau_I \tag{11}$$

using a similar splitting to (10) but without the extra-variables ζ and η:

Algorithm 2 (PDHGMp for solving the I-divergence constrained problem)
Initialization: $u^{(0)}, (p_j^{(0)})_{1\leq j\leq 2} = (\bar{p}_j^{(0)})_{1\leq j\leq 2}$, $\theta\in(0,1]$, $(\rho,\sigma)\in(0,+\infty)^2$ with $\rho\sigma < 1/\|H^*H+L^*L\|_2$
For $k=0,1,\ldots$ repeat until a stopping criteria is reached

1. Step 1 of Algorithm 1
2. $\underset{v_1}{\text{minimize}}\ \|v_1-(p_1^{(k)}+Hu^{(k+1)})\|_2^2$ subject to $D(f,v_1)\leq\tau_I$ as in [22].

3. – 5. Steps 4. - 6. of Algorithm 1

6. $\bar{p}_j^{(k+1)} = p_j^{(k+1)}+\theta(p_j^{(k+1)}-p_j^{(k)}), \quad j=1,2.$

In [24] (see also [3]) statistical arguments were used to show that $\tau_I = \frac{1}{2}n$ is a good estimate in case of moderate Poisson noise. In [5] this estimate was improved in case f has many zero components.

5 Numerical Examples

In this section, we demonstrate the performance of our algorithm by numerical examples implemented in MATLAB (Intel Core i7-870 Processor with 8M Cache, 2.93 GHz, 8 GB physical memory). We have tested the two original images $\overline{u}$, namely 'cameraman' (256×256) and 'brain' (184×140), depicted in Fig. 2 and denoted by B1, resp. B2 in the following.

Fig. 2. Original images 'cameraman' (left) and phantom of a brain image (right)

The images were blurred by a matrix H corresponding to a convolution with a Gaussian kernel with standard deviation 1.3 and mirrored boundary (we have then $m=n$). Their gray values are interpreted as photon counts in the range $[0,\,\nu]$, where ν is the intensity of the image. We tested $\nu = 100, 600, 1200, 2000,$

Table 1. The original values of $D(f, H\overline{u})/n$, $\|T(H\overline{u}) - T(f)\|_2^2/n$ and PSNR, MAE of f

	$D(f, H\overline{u})/n$	$\|T(H\overline{u}) - T(f)\|_2^2/n$	PSNR	MAE
$B1_{100}$	0.5075	1.0086	20.58	66.41e-3
$B1_{600}$	0.5020	1.0034	23.38	41.59e-3
$B1_{1200}$	0.5018	1.0039	23.79	37.17e-3
$B1_{2000}$	0.4979	0.9960	23.97	34.87e-3
$B1_{3000}$	0.4994	0.9989	24.06	33.58e-3
$B2_{100}$	0.4954	0.8866	18.34	82.45e-3
$B2_{600}$	0.5131	1.0004	20.17	61.01e-3
$B2_{1200}$	0.5122	1.0178	20.37	57.90e-3
$B2_{2000}$	0.5063	1.0085	20.49	56.02e-3
$B2_{3000}$	0.4956	0.9899	20.52	55.16e-3

3000 and denoted the blurred, noisy images by $B1_\nu$ and $B2_\nu$. In order to synthetically add Poisson noise to the noise-free image, we applied the MATLAB routine `imnoise(X,'poisson')`. For a quantitative comparison of the images, we computed the peak signal to noise ratio (PSNR) and the MAE defined by $\mathrm{PSNR} = 10\log_{10}\frac{|\max\overline{u}-\min\overline{u}|^2}{\frac{1}{n}\|u-\overline{u}\|_2^2}$, and $\mathrm{MAE} = \frac{1}{n\nu}\|\overline{u}-u\|_1$. The 'true' constraints between the blurred, noisy image f and the original image $\overline{u}$ are given in Table 1. As can be seen, the estimates $\tau_I = n/2$ and $\tau = n$ are good approximations of the true constraints $D(f, H\overline{u})$ and $\|T(H\overline{u}) - T(f)\|_2^2$.

We computed a minimizer of our functional (3) with $C = [0, +\infty)^n$, the $\ell_{2,1}$-norm for Φ, and the discrete gradient operator L ($q = 2n$) by using Algorithm 1 with $\tau = n$. We compared the result with the I-divergence constrained approach (11) and Algorithm 2 with $\tau_I = n/2$. The parameters σ and ρ appearing in PDHGMp (in this setting convergence is theoretically guaranteed for $\sigma\rho < 1/9$) are fitted such that the algorithms give (up to two digits after the comma) the same PSNR, MAE and TV semi-norm (times 10^5 or 10^6) after 1000 iterations as after 100000 iterations. The small values of the constraint misfit are yet another indication that our results are in the vicinity of the true limit points and we have not terminated the algorithms prematurely. Furthermore, we set $\theta = 1$. Fig. 3 shows the restoration results for $B1_{1200}$ and Fig. 4 for $B2_{1200}$.

Fig. 3. Result for the 'cameraman' image $B1_{1200}$ corresponding to Table 3. Corrupted image (left), restoration result by the Anscombe constrained model and Alg. 1 (middle), difference image between the middle image and the one recovered by I-divergence constrained model and Alg. 2 (right). The gray values in the difference image are between -10 and 10, while the image values were scaled up to 1200.

Table 2. Results of Algorithms 1 and 2 with $\tau = n$ and $\tau_I = \frac{n}{2}$

image	σ	ρ	$\|T(H\overline{u}) - T(f)\|_2^2 - n$	$D(f, Hu) - n/2$	TV semi-norm	PSNR	MAE
$B1_{100}$	0.09	1.225	2.2837	-	1.2217e+5	24.28	30.76e-3
	0.12		-	2.5643	1.2646e+5	24.39	30.36e-3
$B1_{600}$	0.0599	2.8	-1.5323	-	8.9230e+5	25.58	25.69e-3
			-	0.0049	8.9402e+5	25.59	25.67e-3
$B1_{1200}$	0.042	3	9.0207	-	1.9190e+6	26.08	24.16e-3
			-	0.045	1.9198e+6	26.08	24.15e-3
$B1_{2000}$	0.027	3.03	0.2634	-	3.3007e+6	26.35	23.30e-3
			-	-0.1911	3.3017e+6	26.36	23.30e-3
$B1_{3000}$	0.0329	4.001	-0.3559	-	5.1667e+6	26.64	22.50e-3
			-	0.2767	5.1673e+6	26.64	22.49e-3
$B2_{100}$	0.55	0.25	0.9730	-	0.9349e+5	19.91	59.78e-3
			-	0.001	1.0459e+5	20.39	53.35e-3
$B2_{600}$	0.040	3.04	-2.7434	-	7.4158e+5	21.81	40.05e-3
	0.050		-	9.0129	7.5693e+5	21.91	39.28e-3
$B2_{1200}$	0.034	3.97	0.0079	-	1.5598e+6	22.33	36.26e-3
	0.042		-	4.2683	1.5673e+6	22.36	36.11e-3
$B2_{2000}$	0.021	4.108	0.7789	-	2.6780e+6	22.72	33.53e-3
	0.041		-	0.8585	2.6885e+6	22.73	33.48e-3
$B2_{3000}$	0.0182	5.413	-0.5284	-	4.0565e+6	22.93	32.08e-3
	0.0282		-	1.5536	4.0603e+6	22.94	32.03e-3

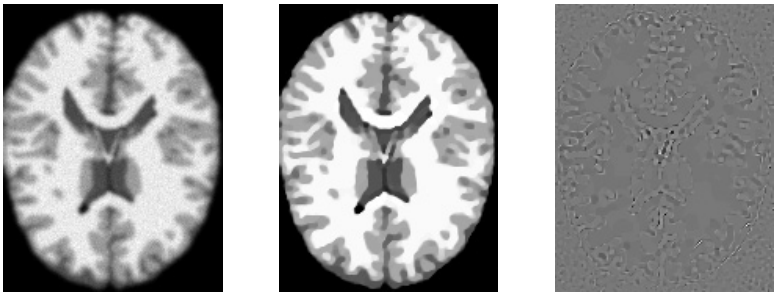


Fig. 4. Result for the 'brain' image $B2_{1200}$ corresponding to Table 3. Corrupted image (left), restoration result by the Anscombe constrained model and Alg. 1 (middle), difference image between the middle image and the one recovered by the I-divergence constrained model and Alg. 2 (right). The gray values in the difference image are between -15 and 15, while the image values were scaled up to 1200.

Table 2 summarizes the results for the different intensities. In the I-divergence constrained approach with the brain data, we stopped after 5000 iterations, while in all the other cases we stopped after 1000 iterations. As expected, we observe that the outcomes of the two algorithms are very similar. More precisely, if u_A, resp. u_I denotes the output of the restoration procedure with Anscombe, resp. I-divergence constraints, then we get for image B1 that $\|u_A - u_I\|_2/(\nu\sqrt{n})$ ranges from 0.004 to 1.74e-4 and $\max|u_A - u_I|/\nu$ from 0.0612 to 0.0031 for the different noise levels.

Finally, Table 3 compares Algorithms 1 and 2 for different constraints τ and τ_I and a central part of the cameraman of size 130×130 with $\nu = 3000$.

Table 3. Results of Algotithms 1 and 2 after 1000 iterations on a part of $B1_{3000}$ for different constraining parameters $\tau = \text{scale} \cdot n$, $\tau_I = \text{scale} \cdot \frac{n}{2}$. The optimal scale is computed as in Table 1.

scale	σ	ρ	$\|T(H\overline{u}) - T(f)\|_2^2 - n$	$D(f, Hu) - n/2$	TV semi-norm	PSNR	MAE
0.8	0.2	0.975	-0.0813	-	2.2400e+6	26.51	22.03e-3
			-	-0.0585	2.2416e+6	26.52	22.02e-3
1.0	0.0269	4	-2.4868	-	1.7071e+6	25.60	21.20e-3
			-	-0.0425	1.7073e+6	25.60	21.19e-3
1.0095 (optimal)	0.0239	4.002	-0.0616	-	1.6964e+6	25.56	21.29e-3
			-	-0.0927	1.6971e+6	25.56	21.29e-3
1.2	0.011	8	1.1405	-	1.5563e+6	24.93	22.96e-3
			-	0.0611	1.5565e+6	24.93	22.96e-3
2	0.004	30.065	-1.7501	-	1.3388e+6	23.70	27.02e-3
			-	-0.0257	1.3391e+6	23.71	27.02e-3

6 Summary and Conclusions

We have considered a constrained restoration model for images corrupted by a linear transform and Poisson noise by making use of the Anscombe transform. In contrast with penalized approaches, a main advantage of the proposed one is that it makes it possible to employ a simple estimate for the model parameter. We have provided proximal algorithms to find a minimizer of the model, which are based on epigraphical projections, and we have shown that the performance is similar to a recently introduced I-divergence constrained model. Future research directions include the following: *i)* replacing or combining the discrete gradient operator L with other ones (discrete higher order operators, nonlocal means, wavelet-like transforms) and handling other problems than deblurring ones, *ii)* considering convex optimization problems involving multiple constraints for which the epigraphical projection approach may be quite efficient, see [10], *iii)* restoring images with Poisson+Gauss noise, see [16,17], and *iv)* finding numerically efficient methods to map the constraint bound to the parameter of the corresponding penalized functional.

References

1. Anscombe, F.J.: The transformation of Poisson, binomial and negative-binomial data. Biometrika 35, 246–254 (1948)
2. Aravkin, A.Y., Burkey, J.V., Friedlander, M.P.: Variational properties of value functions. Preprint Univ. British Columbia (2012)
3. Bardsley, J.M., Goldes, J.: Regularization parameter selection methods for ill-posed Poisson maximum likelihood estimation. Inverse Problems 25(9), 095005 (2009)
4. Bot, R.I., Hendrich, C.: Convergence analysis for a primal-dual monotone + skew splitting algorithm with application to total variation minimization. Preprint Univ. Chemnitz (2012)
5. Carlavan, M., Blanc-Féraud, L.: Sparse Poisson noisy image deblurring. IEEE Transactions on Image Processing 21(4), 1834–1846 (2012)
6. 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)

7. Chaux, C., Blanc-Féraud, L., Zerubia, J.: Wavelet-based restoration methods: Application in 3d confocal microscopy images. In: Proc. SPIE Conf. Wavelets, San Diego, p. 67010E (2007)
8. Chaux, C., Pesquet, J.-C., Pustelnik, N.: Nested iterative algorithms for convex constrained image recovery problems. SIAM Journal on Imaging Science 2(2), 730–762 (2009)
9. Cherchia, G., Pustelnik, N., Pesquet, J.-C., Pesquet-Popescu, B.: A proximal approach for constrained cosparse modelling. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Kyoto, Japan (2012)
10. Chierchia, G., Pustelnik, N., Pesquet, J.-C., Pesquet-Popescu, B.: Epigraphical projection and proximal tools for solving constrained convex optimization problems - part I (2012) (preprint)
11. Combettes, P.L., Pesquet, J.-C.: Proximal splitting methods in signal processing. In: Bauschke, H.H., Burachik, R.S., Combettes, P.L., Elser, V., Luke, D.R., Wolkowicz, H. (eds.) Fixed-Point Algorithms for Inverse Problems in Science and Engineering, pp. 185–212. Springer, New York (2011)
12. Combettes, P.L., Pesquet, J.-C.: Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. Set-Valued and Variational Analysis 20(2), 307–330 (2012)
13. Dupé, F.-X., Fadili, J., Starck, J.-L.: A proximal iteration for deconvolving Poisson noisy images using sparse representations. IEEE Transactions on Image Processing 18(2), 310–321 (2009)
14. Figueiredo, M.A.T., Bioucas-Dias, J.M.: Restoration of Poissonian images using alternating direction optimization. IEEE Transactions on Image Processing 19(12), 3133–3145 (2010)
15. Hanke–Bourgeois, M.: Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens. Teubner, Stuttgart (2002)
16. Jezierska, A., Chouzenoux, E., Pesquet, J.-C., Talbot, H.: A primal-dual proximal splitting approach for restoring data corrupted with Poisson-Gaussian noise. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), Kyoto, Japan (2012)
17. Li, J., Shen, Z., Jin, R., Zhang, X.: A reweighted ℓ_2 method for image restoration with Poisson and mixed Poisson-Gaussian noise. UCLA Preprint (2012)
18. Mikkitalo, M., Foi, A.: Optimal inversion of the Anscombe transformation in low-count Poisson image denoising. IEEE Transactions on Image Processing 20(1), 99–109 (2011)
19. 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, pp. 810–817 (2009)
20. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
21. Setzer, S., Steidl, G., Teuber, T.: Deblurring Poissonian images by split Bregman techniques. Journal of Visual Communication and Image Representation 21(3), 193–199 (2010)
22. Teuber, T., Steidl, G., Chan, R.-H.: Minimization and parameter estimation for seminorm regularization models with I-divergence constraints. Preprint Univ. Kaiserslautern (2012)
23. Vu, B.C.: A splitting algorithm for dual monotone inclusions involving cocoercive operators. Advances in Computational Mathematics (2012) (accepted)
24. Zanella, R., Boccacci, P., Zanni, L., Bertero, M.: Efficient gradient projection methods for edge-preserving removal of Poisson noise. Inverse Problems 25(4), 045010 (2009)

Static and Dynamic Texture Mixing Using Optimal Transport

Sira Ferradans[1,⋆], Gui-Song Xia[2],
Gabriel Peyré[1], and Jean-François Aujol[3]

[1] Ceremade, Univ. Paris-Dauphine
{sira.ferradans,gabriel.peyre}@ceremade.dauphine.fr
[2] LIERSMARS, Wuhan University
guisong.xia@whu.edu.cn
[3] IMB, Université Bordeaux 1
Jean-Francois.Aujol@math.u-bordeaux1.fr

Abstract. This paper tackles the problem of mixing static and dynamic texture by combining the statistical properties of an input set of images or videos. We focus on Spot Noise textures that follow a stationary and Gaussian model which can be learned from the given exemplars. From here, we define, using Optimal Transport, the distance between texture models, derive the geodesic path, and define the barycenter between several texture models. These derivations are useful because they allow the user to navigate inside the set of texture models, interpolating a new one at each element of the set. From these new interpolated models, new textures can be synthesized of arbitrary size in space and time. Numerical results obtained from a library of exemplars show the ability of our method to generate new complex and realistic static and dynamic textures.

1 Introduction

The problem of synthesizing new textures is central in Image Processing and Computer Graphics. In order to render scenes for video games or animation films, a texture is mapped onto a given surface. Because the shape and extension of the surface may vary, the main goal of texture synthesis is to be able to generate as much texture as it is needed in a fast and realistic way. This problem has been addressed since the beginning of Computer Graphics, so we can find many solutions in the literature.

1.1 Previous Works

Copy-Based Methods. These methods are adapted to complicated (not even random) textures. The main assumption is that textures contain repeating local patterns. They synthesize new textures by copying patches or pixels from the original image in a way that preserves local structure. First proposed by Popat

⋆ This work has been supported by the European Research Council (ERC project SIGMA-Vision).

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 137–148, 2013.

and Picard [1] in the context of clustering, it was simplified and popularized by Efros and Leung [2] for texture synthesis. For a thorough review of copy-based methods we refer the reader to the article [3].

Statistical Texture Models. Statistical parametric models are generally not as good in handling complex texture patterns, but are more flexible and fast, see for instance [4]. The main assumption of these models is that textures are modeled by a probability distribution. Thus, texture analysis consists in estimating the probability function and texture synthesis amounts to generate new realizations of this probability distribution. Many methods have been proposed within this category, specially relevant is the use of Markov random fields (i.e. [5]) which model also copy-based methods (see for instance [6]) or stationary Gaussian random fields [7].

Spot Noise models were first introduced by van Wijk [8] and are stationary models that replicate, in random locations, simple spot images. Galerne et al.[7] analyze the asymptotical behavior of Van Wijk's method to propose a new method (Asymptotic Discrete Spot Noise), which consists in modeling texture with a stationary Gaussian distribution. In this paper, we focus our attention on this texture model and extend this framework to texture mixing.

Dynamic Texture Synthesis. Many methods for static image synthesis have been adapted to the dynamic scenario (see for example [9], [6] in the context of the copy-based methods), but very few have studied the specific dynamics of texture in time. In the context of Gaussian textures, linear dynamical systems [10] and dynamic multiscale autoregressive models [11] have been proposed to model the evolution of texture with time. However, these methods define models that are difficult to manipulate (for instance to achieve model mixing.) Recently, an extension to Galerne et al.'s model [7] for stationary Gaussian dynamic textures has been proposed by Xia et al. [12]. In this paper, Xia et al. model dynamic texture as a 3D Gaussian random field, with stationarity in space *and* time. Here, we take advantage of this extension to generate new mixed models from input dynamic textures.

Texture Mixing. More complex textures can be obtained by texture mixing which extends the traditional texture synthesis by considering the interplay between several texture models. This is a difficult problem since it requires to average very distinct statistical features. Previous works make use of mixture models, see for instance [13]. The use of non-parameteric histogram averaging has also been proposed for grayscale [14] as well as color and wavelets features [15]. We propose here a simpler approach that makes use of a parameterization of the Gaussian texture model. Defining a geodesic path with Optimal Transport (OT) between the original Gaussian models, we can generate new textures sharing the characteristics of the input ones. The proposed method ensures that the new texture model stays Gaussian.

1.2 Contributions

We propose a new framework for texture synthesis based on the definition of geodesic paths between stationary Gaussian texture models. Our first

contribution is the definition of the geodesic distance, according to OT, between texture models, and the geodesic path associated to such distance. The straightforward consequence of having this geodesic path is that we obtain a method for interpolating new texture models with the statistical properties of the input textures. Our second contribution consists in the extension of the interpolation formula between two models to *several* models by defining the OT barycenter. The final algorithm is solidly founded, the texture synthesis is fast, and the obtained results look natural.

2 Spot Noise Texture Model

We model textures as stationary Gaussian random fields. These assumptions allow us to learn the texture model parameters from a single texture exemplar.

2.1 Notations

Deterministic input exemplar textures are represented as $f \in \mathbb{R}^{N\times d}$, where $N = \prod_{j=1}^{k} N_j$ is the number of pixels ($k = 2$ for image and $k = 3$ for videos) and d is the number of channels ($d = 1$ for grayscale and $d = 3$ for color datasets). We refer to $f(x) \in \mathbb{R}^3$ to the color vector at position x, where there are N such positions x. We denote Gaussian distributions as $\mu = \mathcal{N}(m, \Sigma)$ where $m \in \mathbb{R}^{N\times d}$ is the mean of the distribution and $\Sigma \in \mathbb{R}^{Nd\times Nd}$ is a positive semi-definite covariance matrix.

The k-dimensional discrete Fourier transform $\hat{f} \in \mathbb{R}^{N\times d}$ of $f \in \mathbb{R}^{N\times d}$ is defined as

$$\forall\, \omega = (\omega_1, \ldots, \omega_k), \quad \hat{f}(\omega) = \sum_x f(x) e^{-\sum_j \frac{2i\pi}{N_j}\omega_j x_j} \in \mathbb{R}^d.$$

It is computed in $O(Nd\log(Nd))$ operations and it is inverted with the same complexity using the inverse FFT.

Given two periodic images or videos $f, g \in \mathbb{R}^N$, we define the convolution $h = f \star g$ of f and g as

$$h(x) = \sum_y f(x-y)g(y) \quad \Longleftrightarrow \quad \hat{h}(\omega) = \hat{f}(\omega)\hat{g}(\omega). \tag{1}$$

2.2 Stationary Gaussian Models

We model a texture as a random vector X distributed according to some Gaussian distribution μ, which we denote $X \sim \mu$. A random vector X is stationary if the distribution of $X(\cdot)$ and $X(\cdot + \tau)$ are the same, for any translation vector $\tau \in \mathbb{Z}^k$, where we assume periodic boundary conditions. Section 2.4 details how to learn the parameters when the input exemplar is non-periodic.

The fact that X is stationary implies that the mean $m(x) \in \mathbb{R}^d$ is independent of the position x and the covariance operator Σ is block-diagonal over the Fourier domain, thus it can be computed using convolutions, that is to say, the covariance operator $y = \Sigma f$ can be applied over the Fourier domain as $\hat{y}(\omega) = \hat{\Sigma}(\omega)\hat{f}(\omega)$ where $\hat{\Sigma}(\omega) \in \mathbb{C}^{d\times d}$ is a positive Hermitian matrix.

2.3 Spot Noise Model

A Spot Noise (SN) random vector $X = (X_1, \ldots, X_d)$ is a Gaussian texture model obtained from an input texture $f = (f_1, \ldots, f_d)$ by convolving each channel with the same Gaussian white noise [7]. This reads

$$\forall\, j = 1, \ldots, d, \quad X_j = m_j + f_j \star W \tag{2}$$

where $\star$ is the k-dimensional periodic convolution and the W is a white noise $W \sim \mathcal{N}(0, \mathrm{Id}_N/\sqrt{N})$, and m_j is the mean of f_j. We denote $\mu = \mu(f)$ the distribution of this random vector X, which is the SN distribution associated to the exemplar f.

Equivalently, Spot Noise models are the stationary Gaussian vectors for which the matrices $\hat{\Sigma}(\omega)$ are rank one, and can thus be decomposed as

$$\hat{\Sigma}(\omega) = \hat{f}(\omega)\hat{f}(\omega)^*, \tag{3}$$

where $u^* \in \mathbb{C}^d$ is the complex conjugate transpose of $u \in \mathbb{C}^d$.

2.4 Stationary Gaussian Model Synthesis

Once the parameters Σ and m of the Gaussian model $\mu = \mathcal{N}(m, \Sigma)$ have been computed, the synthesis of a texture $g \in \mathbb{R}^{N \times d}$ is obtained using a realization of the Gaussian process.

For a generic stationary model, this is achieved by computing the Cholesky factorization of the frequency covariance $\hat{\Sigma}(\omega) = \hat{A}(\omega)\hat{A}(\omega)^*$ where $A(\omega)^* \in \mathbb{C}^{d \times d}$ is the complex conjugate transpose of the matrix $A(\omega) \in \mathbb{C}^{d \times d}$. Then, we compute $\hat{g}(\omega) = \hat{A}(\omega)\hat{w}(\omega)$ for $\omega \neq 0$ where w is a realization of $\mathcal{N}(0, \mathrm{Id}_{Nd}/\sqrt{N})$ and $\hat{g}(0) = Nm(0)$ is the constant mean of the model.

In the special case where the model is a Spot Noise $\mu(f)$, meaning that $\hat{\Sigma}(\omega) = \hat{f}(\omega)\hat{f}(\omega)^*$, the synthesis is even faster using, for $\omega \neq 0$, $\hat{g}(\omega) = \hat{w}(\omega)\hat{f}(\omega)$, or equivalently using a realization of the convolution formula (2).

Boundary Conditions. Up to now, the image is assumed to be periodic in our texture model. To be able to learn the parameters from a non-periodic image, a preprocessing is required. Symmetrizing the image with respect to the boundaries introduces axis-aligned artifacts. Following [7], we substitute each channel f_j of the input exemplar by its periodic component as defined by Moisan [16].

Extending the Texture Size. In our context, the process of extending the input texture of size $N_1 \times N_2 \times N_3$ to any arbitrary size $M_1 \times M_2 \times M_3$ can be done following the method proposed by Galerne et al. [7]. The periodic component of the original texture is located at the center of a flat new image (or video) of value m and dimensions $M_1 \times M_2 \times M_3$. To avoid the introduction of high frequencies, the new borders are smoothed with a spatial windowing function. This extended image or video is then used to learn a texture model of size $M_1 \times M_2 \times M_3$.

3 Optimal Transport Geodesic of Spot Noise

Now that we have defined our texture model we proceed to the exposition of our model mixing method. It operates using OT geodesic over the set of Gaussian distributions. This is achieved by defining the OT geodesic interpolation [17] over the space of Gaussian models.

3.1 Optimal Transport Geodesics of Gaussian Fields

The first step is to define a geodesic distance between texture models, that is to say, between two arbitrary stationary Gaussian distributions. The L^2 OT distance between $\mu_i = \mathcal{N}(m_i, \Sigma_i)$ reads:

$$d(\mu_0, \mu_1)^2 = \operatorname{tr}(\Sigma_0 + \Sigma_1 - 2\Sigma_{0,1}) + \|m_0 - m_1\|^2,$$

where $\Sigma_{0,1} = (\Sigma_0^{1/2}\Sigma_1\Sigma_0^{1/2})^{1/2}$ (see for instance [18]).

This distance is known to be geodesic, meaning that $d(\mu_0, \mu_1)$ is equal to the length of the shortest path (the so-called geodesic path) $t \in [0,1] \mapsto \mu_t$ between μ_0 and μ_1. This geodesic path satisfies

$$\forall\, t \in [0,1], \mu_t = \operatorname*{argmin}_{\mu} (1-t)d(\mu_0, \mu)^2 + td(\mu_1, \mu)^2,$$

where $t \mapsto \mu_t$ parameterizes the path, so μ_t can also be understood as a weighted barycenter of the input texture models. The following proposition shows that this geodesic path is composed of Gaussian models, so that the set of Gaussian models are geodesically convex for the OT distance [19].

Proposition 1. *If* $\ker(\Sigma_0) \not\subset \ker(\Sigma_1)^\perp$ *and* $\operatorname{rank}(\Sigma_0) \geqslant \operatorname{rank}(\Sigma_1)$*, the unique Gaussian OT-geodesic of Gaussian distributions* $\mu_i = \mathcal{N}(m_i, \Sigma_i)$ *(for* $i = 0, 1$*) is a Gaussian distribution* $\mathcal{N}(m_t, \Sigma_t)$ *where* $m_t = (1-t)m_0 + tm_1$ *and*

$$\Sigma_t = [(1-t)\mathrm{Id} + t\Pi]\Sigma_0[(1-t)\mathrm{Id} + t\Pi] \tag{4}$$

where $\Pi = \Sigma_1^{1/2}\Sigma_{0,1}^{+}\Sigma_1^{1/2}$ *and where* A^+ *is the Moore-Penrose pseudo-inverse and* $A^{1/2}$ *is the unique positive square root of a symmetric semi-definite matrix.*

Proof. The proof follows the one in [19] with the extra care that the covariance can be rank-deficient, hence requiring a pseudo-inverse.

Note that the condition $\operatorname{rank}(\Sigma_0) \geqslant \operatorname{rank}(\Sigma_1)$ is not restrictive since one can otherwise exchange the roles of Σ_0 and Σ_1 and replace t by $1-t$ when computing the geodesic path.

We now show that if the input models μ_0, μ_1 are Spot Noise, then the geodesic interpolation is also Spot Noise. This means that the texture models we consider are geodesically convex.

Theorem 1. *For $i = 0, 1$, let $\mu_i = \mu(f^{[i]})$ be Spot Noise distributions associated with $f^{[0]}, f^{[1]} \in \mathbb{R}^{N\times d}$. The OT geodesic path μ_t defined in equation (4) is a Spot Noise model $\mu_t = \mu(f^{[t]})$ where $f^{[t]} = (1-t)f^{[0]} + tg^{[1]}$ with*

$$\forall\, \omega, \quad \hat{g}^{[1]}(\omega) = \hat{f}^{[1]}(\omega)\frac{\hat{f}^{[1]}(\omega)^*\hat{f}^{[0]}(\omega)}{|\hat{f}^{[1]}(\omega)^*\hat{f}^{[0]}(\omega)|}. \tag{5}$$

Proof. The covariance operator is a matrix-convolution operator, thus we can define in the Fourier domain its associated kernel as $\hat{\Sigma}_i(\omega) = \hat{f}^{[i]}(\omega)\hat{f}^{[i]}(\omega)^* \in \mathbb{C}^{d\times d}$. The symmetric operator Π from equation (4) is also a matrix convolution $\Pi g = \pi \star g$ with kernel whose Fourier transform is

$$\hat{\pi}(\omega) = \hat{\Sigma}_1^{\frac{1}{2}}(\omega)\left(\hat{\Sigma}_1^{\frac{1}{2}}(\omega)\hat{\Sigma}_0(\omega)\hat{\Sigma}_1^{\frac{1}{2}}(\omega)\right)^{-\frac{1}{2}}\hat{\Sigma}_1^{\frac{1}{2}}(\omega).$$

Note that the square root of a rank-1 matrix can be easily computed as

$$\forall\, u \in \mathbb{C}^d, \quad (uu^*)^{1/2} = \frac{1}{|u|}uu^* \in \mathbb{C}^{d\times d}.$$

Using this property, together with the definition of $\hat{\Sigma}$, and denoting $u_i = \hat{f}^{[i]}(\omega)$ one proves that

$$\hat{\pi}(\omega) = \frac{1}{|u_1^*u_0|}u_1u_1^*(u_1u_1^*)^{-1}u_1u_1^* = \frac{u_1u_1^*}{|u_1^*u_0|}. \tag{6}$$

Observe that although the matrix $u_1^*u_0$ is non invertible, the above expression is correct because the mapping $\pi(\omega)$ is zero on the orthogonal of u_1.

The expression (4) of the covariance implies that it is also a matrix-convolution operator with kernel defined over the Fourier domain as

$$\hat{\Sigma}_t(\omega) = \hat{f}^{[t]}(\omega)\hat{f}^{[t]}(\omega)^* \in \mathbb{C}^{d\times d},$$

where

$$\hat{f}^{[t]}(\omega) = [(1-t)\mathrm{Id} + t\hat{\pi}(\omega)]\hat{f}^{[0]} \in \mathbb{C}^d.$$

Using the expression (6) for $\hat{\pi}(\omega)$, one thus has that $\mu_t = \mu(f^{[t]})$ is a Spot Noise model where $f^{[t]}$ is defined as

$$\hat{f}^{[t]}(\omega) = (1-t)\hat{f}^{[0]}(\omega) + t\underbrace{\frac{\hat{f}^{[1]}(\omega)^*\hat{f}^{[0]}(\omega)}{|\hat{f}^{[1]}(\omega)^*\hat{f}^{[0]}(\omega)|}\hat{f}^{[1]}(\omega)}_{\hat{g}^{[1]}(\omega)} \in \mathbb{C}^d.$$

Therefore the new interpolated models μ_t are Gaussian, Spot Noise, and their covariances can be computed by a suitable averaging of the Fourier transforms of the input exemplars. The pseudocode of the proposed method is provided in Fig. 1.

Table 1. Pseudocode for geodesic mixing between two input exemplars

Input: exemplars $(f^{[0]}, f^{[1]})$, weight $t \in [0,1]$.
Output: realization h of the interpolated model μ_t between $\mu_i = \mu(f^{[i]})$ for $i = 0, 1$.
Preprocessing: for $i = 0, 1$,
- Replace $f^{[i]}$ by its periodic component. If needed extend its size by zero padding.
- Compute the mean m_i and subtract it, $\forall\, x, f^{[i]}(x) \leftarrow f^{[i]}(x) - m_i$.
- Computes the Fourier transforms $\hat{f}^{[i]}$ of $f^{[i]}$ using FFTs.

Model mixing:
- Compute $\hat{g}^{[1]}$ with equation (5).
- $\forall\, \omega$, compute $\hat{f}^{[t]}(\omega) = (1-t)\hat{f}^{[0]}(\omega) + t\hat{g}^{[1]}(\omega)$
- Compute $m_t = (1-t)m_0 + tm_1 \in \mathbb{R}^d$.

Spot Noise synthesis:
- Compute a realization $w \in \mathbb{R}^N$ of $\mathcal{N}(0, \mathrm{Id}_N/\sqrt{N})$ (using e.g. Matlab `randn`).
- Compute the Fourier transform $\hat{w}$ of w using FFT.
- $\forall\, \omega \neq 0$, $\forall\, j = 1, \ldots, d$ compute $\hat{h}_j(\omega) = \hat{f}_j^{[t]}(\omega)\hat{w}(\omega)$. Set $\hat{h}(0) = Nm_t$.
- Compute $h \in \mathbb{R}^{N \times d}$ from $\hat{h}$ using the inverse FFT.

3.2 Numerical Results

Let us now show some results obtained with the Spot Noise geodesic mix method explained in this section. Each row of Figure 1 corresponds to a single experiment which consists in learning the Gaussian model of two input textures ($f^{[0]}$ and $f^{[1]}$) and interpolate new Gaussian models following the path between model $f^{[0]}$ and $f^{[1]}$. Note that the images in columns $t = 0, 1$ are instances of the original models. We would like to point out how this instances are perceptually similar to the original input textures.

Regarding the columns for $t = 1/3$ and $2/3$, we would like to point out how the color changes gradually as we move along the geodesic path and that the spacial patterns of the original textures are being mixed also in different proportion.

An example of dynamic texture mixing can be observed in Figure 2. Each row corresponds to a single video, where every image is a single frame, ordered from left to right. The first and last rows are the inputs and the two middle ones where interpolated with the geodesic mix method.

4 Optimal Transport Barycenter of Spot Noise

In the previous section, we explained how to create new texture models by following a geodesic path between the two input models. This section extends this idea to more exemplars using a geodesic barycenter of the models. In the case of 3 exemplars (resp. 4), this can be visualized by locating the input models on the vertices of a 2-D triangle (resp. 3-D tetrahedron). Computing the OT barycenter allows one to navigate inside the triangle (resp. tetrahedron).

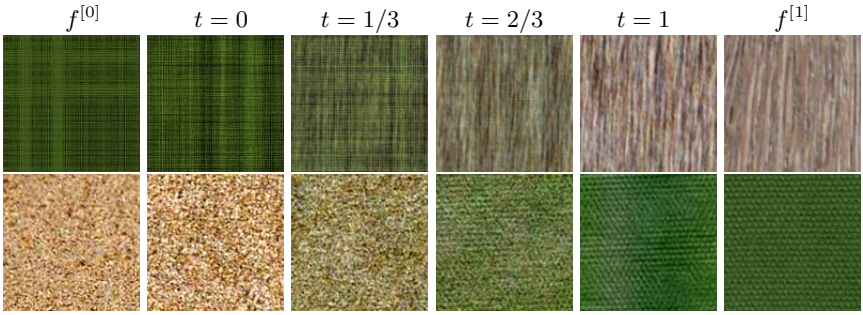


Fig. 1. $f^{[0]}$ and $f^{[1]}$ are the input texture images. After learning the input models, we interpolate new ones ($t = 0, 1/3, 2/3, 1$) along the OT geodesic path from $f^{[0]}$ to $f^{[1]}$.

4.1 Optimal Transport Barycenter

Given a family of Gaussian distributions $(\mu_i)_{i \in I}$ and weights ρ_i with $\sum_i \rho_i = 1$, where $\rho_i \geq 0$, the OT barycenter is defined as

$$\mu^\star = \underset{\mu}{\operatorname{argmin}} \sum_{i \in I} \rho_i d(\mu_i, \mu)^2. \tag{7}$$

Note that for $|I| = 2$ we retrieve the geodesic path by setting $t = \rho_2$. For the special case of a Gaussian distribution $\mu_i = \mathcal{N}(m_i, \Sigma_i)$, there is no close form solution if $|I| > 2$. The barycenter can be shown to be Gaussian [20]

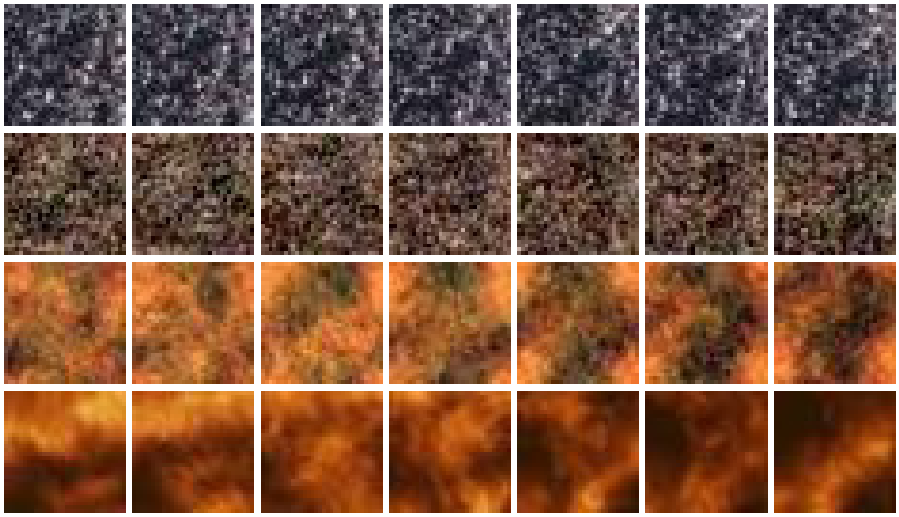


Fig. 2. Example of dynamic textures mixing. The first and last row correspond to $f^{[0]}$ and $f^{[1]}$ respectively, being the order of the frames from left to right. The central rows are instances of the interpolated dynamic texture models.

Table 2. Pseudocode for mixing several input exemplars

Input: exemplars $(f^{[i]})_{i\in I}$, weight $(\rho_i)_i$ with $\sum_{i\in I}\rho_i = 1$.
Output: realization h of the interpolated model $\mu^\star$ between $\mu_i = \mu(f^{[i]})$ for $i \in I$.
Preprocessing: for $i \in I$, apply the pre-processing step of Table 1.
Model mixing: for each ω, do
- $\forall\, i \in I$, compute $\hat{\Sigma}_i(\omega) = (\hat{f}_p^{[i]}(\omega))(\hat{f}_p^{[i]}(\omega))^* \in \mathbb{C}^{d\times d}$
- Initialize $\hat{\Sigma}^{(0)}(\omega) = 0 \in \mathbb{C}^{d\times d}$.
- Repeat until convergence $\hat{\Sigma}^{(k+1)}(\omega) = \Phi_\omega(\hat{\Sigma}^{(k)}(\omega))$ (see (9)), $k \leftarrow k+1$.
- Set $\hat{\Sigma}^\star(\omega) = \hat{\Sigma}^{(k)}(\omega)$. Compute the Cholesky factorization $\hat{\Sigma}^\star(\omega) = \hat{A}(\omega)\hat{A}(\omega)$ of $\hat{\Sigma}^\star(\omega)$.

Gaussian model synthesis:
- Compute $m^\star = \sum_{i\in I}\rho_i m_i \in \mathbb{R}^d$.
- Compute a realization $w \in \mathbb{R}^{N\times d}$ of $\mathcal{N}(0, \mathrm{Id}_{Nd})$.
- Compute the Fourier transform $\hat{w} \in \mathbb{R}^{N\times d}$ of w using FFT.
- $\forall\, \omega \neq 0$, compute $\hat{h}(\omega) = \hat{\Sigma}^\star(\omega)\hat{w}(\omega) \in \mathbb{C}^d$. Set $\hat{h}(0) = m^\star$.
- Compute $h \in \mathbb{R}^{N\times d}$ from $\hat{h}$ using the inverse FFT.

$\mu^\star = \mathcal{N}(m^\star, \Sigma^\star)$, where $m^\star = \sum_{i\in I}\rho_i m_i$ and the covariance matrix is solution of the fixed point equation $\Phi(\Sigma^\star) = \Sigma^\star$ where

$$\Phi(\Sigma) = \sum_{i\in I}\rho_i \left(\Sigma^{1/2}\Sigma_i\Sigma^{1/2}\right)^{1/2}. \tag{8}$$

This barycenter can be shown to be unique if one of the Σ_i is full rank [20]. We leave for future work the theoretical analysis of the uniqueness when all the covariances are rank-deficient.

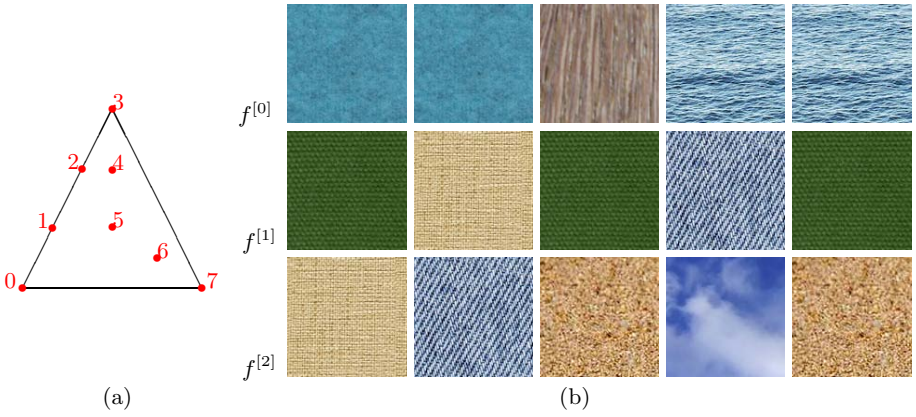


Fig. 3. (a) Spatial location scheme. (b) Each column corresponds to a single experiment, where $f^{[0]}, f^{[1]}, f^{[2]}$ are the original textures located at the vertices of the triangle in positions $0, 3, 7$, respectively. The other numbers correspond to the interpolated Gaussian models. Instances of all of these models can be observed in Figure 4 (a)-(e).

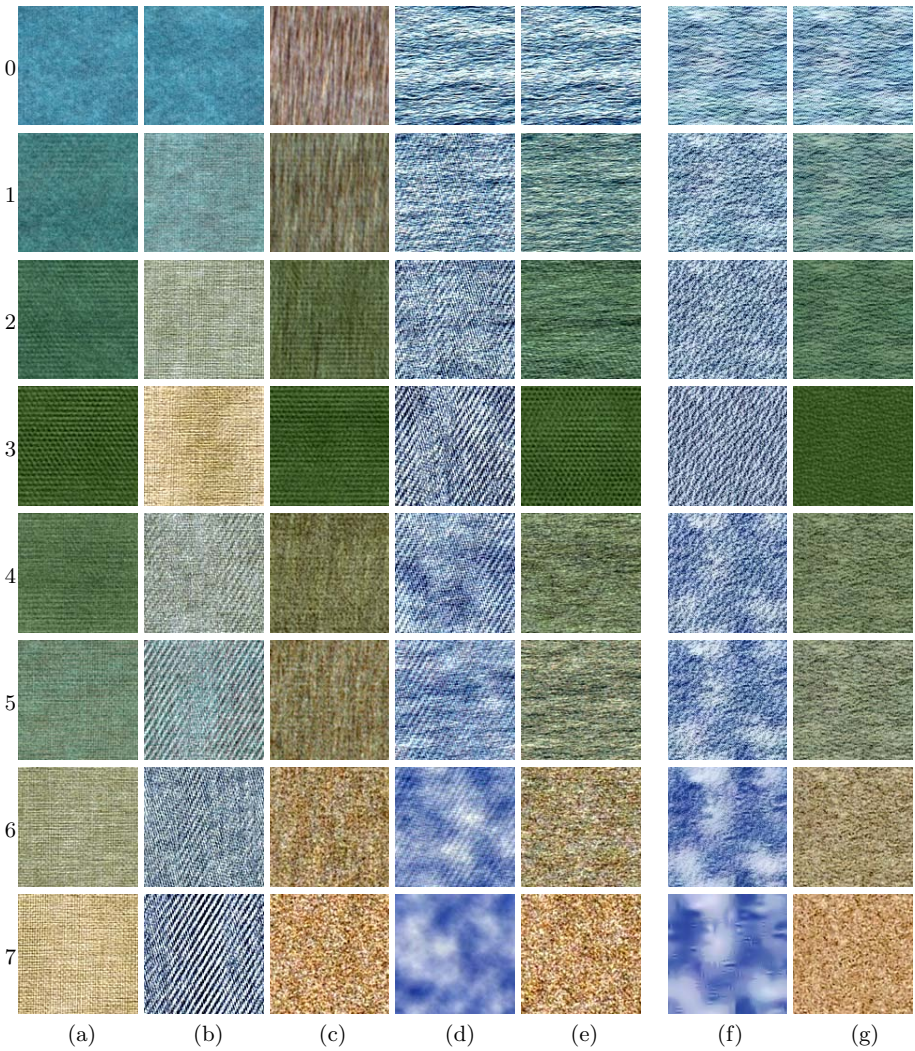


Fig. 4. Each column corresponds to a single experiment. The parameter $\rho = (\rho_1, \rho_2, \rho_3)$ of equation 7 is defined according to the triangle coordinates of the points in Figure 3(a). **(a)-(e)** Images obtained with the barycenter mix method whose input textures are shown as columns in Figure 3, respectively. **(f)(g)** Results obtained by the first method proposed by Rabin et al. [15].

4.2 Spot Noise Barycenter

When μ_i are Spot Noises, the covariance $\Sigma^\star$ of the barycenter is block diagonal over the Fourier domain, and the blocks $\hat{\Sigma}^\star(\omega)$ satisfy the fixed point equation $\hat{\Sigma}^\star(\omega) = \Phi_\omega(\hat{\Sigma}^\star(\omega))$ with

$$\Phi_\omega(\Sigma) = \sum_{i \in I} \rho_i \left(\Sigma^{1/2} \hat{\Sigma}_i(\omega) \Sigma^{1/2} \right)^{1/2}. \tag{9}$$

We note that in general, $\mu^\star$ is not Spot Noise because $\hat{\Sigma}^\star(\omega)$ is not necessarily rank one.

Numerical Computation. Following [21], we propose to compute $\hat{\Sigma}^\star(\omega)$ by iterating the mapping Φ_ω, i.e. compute the sequence $\hat{\Sigma}^{(k+1)}(\omega) = \Phi_\omega(\hat{\Sigma}^{(k)}(\omega))$. Although the mapping Φ_ω is not strictly contracting, we observe numerically the convergence $\hat{\Sigma}^{(k)}(\omega) \to \hat{\Sigma}^\star(\omega)$ when $k \to +\infty$. The numerical computation of Φ_ω in the case $d = 3$ requires the computation of the square root of 3×3 matrices, which is performed explicitly by computing the eigenvalue of the symmetric matrix as the root of a third order polynomial. The pseudocode of the method is detailed in Table 2.

4.3 Numerical Examples

Given three input textures, $f^{[0]}, f^{[1]}, f^{[2]}$, and the path defined in Figure 3(a) by the red numbers in increasing order, we generate the Gaussian models associated to each point. A realization of each of these models can be observed in Figure 4 (a)-(e) using as input textures the columns of Figure 3, respectively. Note how, as we approach an input model, the features of it tend to predominate in the synthesized texture and how the color and the texture patterns are smoothly interpolated along the geodesic path. We would also like to note that this method is also able to reproduce small periodic patterns. Finally, in Figure 4 (f) (g) we show the results obtained with the method by Rabin et al. [15], to be compared with the columns Figure 4 (d) (e), respectively.

5 Conclusion

We have presented a new method for texture mixing that enables the creation of new complex textures from a set of exemplars.

Given two texture models, we used the OT geodesic path over Gaussian distributions to interpolate new texture models. The numerical results show how the method is able to merge the visual features of the original images into new complex textures. We also generalized this OT geodesic method to the mixing of an arbitrary number of models using OT barycenters. We postpone for later research a thorough perceptual evaluation of the output textures.

References

1. Popat, K., Picard, R.W.: Novel cluster-based probability model for texture synthesis, classification, and compression. In: Visual Communications and Image Processing, pp. 756–768 (1993)

2. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proc. of ICCV 1999, p. 1033 (1999)
3. Wei, L.Y., Lefebvre, S., Kwatra, V., Turk, G.: State of the art in example-based texture synthesis. In: Eurographics 2009, State of the Art Report, EG-STAR. Eurographics Association (2009)
4. Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. Int. Journal of Computer Vision 40, 49–70 (2000)
5. 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, 107–126 (1998)
6. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. ACM Trans. Graph. 22, 277–286 (2003)
7. Galerne, B., Gousseau, Y., Morel, J.M.: Random phase textures: Theory and synthesis. IEEE Transactions on Image Processing 20, 257–267 (2011)
8. van Wijk, J.J.: Spot noise texture synthesis for data visualization. In: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991, pp. 309–318. ACM, New York (1991)
9. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, pp. 479–488. ACM Press/Addison-Wesley Publishing Co., New York (2000)
10. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. International Journal of Computer Vision 51, 91–109 (2003)
11. Doretto, G., Jones, E., Soatto, S.: Spatially homogeneous dynamic textures. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004, Part II. LNCS, vol. 3022, pp. 591–602. Springer, Heidelberg (2004)
12. Xia, G.S., Ferradans, S., Peyré, G., Aujol, J.F.: Compact representations of stationary dynamic textures. In: Proc. ICIP 2012 (2012)
13. Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M.: Texture mixing and texture movie synthesis using statistical learning. IEEE Tr. on Vis. and Comp. Graph. 7, 120–135 (2001)
14. Matusik, W., Zwicker, M., Durand, F.: Texture design using a simplicial complex of morphable textures. ACM Transactions on Graphics 24, 787–794 (2005)
15. Rabin, J., Peyré, G., Delon, J., Bernot, M.: Wasserstein barycenter and its application to texture mixing. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 435–446. Springer, Heidelberg (2012)
16. Moisan, L.: Periodic plus smooth image decomposition. Journal of Mathematical Imaging and Vision 39, 161–179 (2011)
17. Villani, C.: Topics in Optimal Transportation. American Mathematical Society (2003)
18. Dowson, D.C., Landau, B.V.: The fréchet distance between multivariate normal distributions. J. Multivariate Anal. 3, 450–455 (1982)
19. Takatsu, A.: Wasserstein geometry of gaussian measures. Osaka J. Math (2011)
20. Agueh, M., Carlier, G.: Barycenters in the wasserstein space. SIAM J. on Mathematical Analysis 43, 904–924 (2011)
21. Knott, M., Smith, C.S.: On a generalization of cyclic monotonicity and distances among random vectors. Linear Algebra and its Applications 199, 363–371 (1994)

A TGV Regularized Wavelet Based Zooming Model

Kristian Bredies and Martin Holler

Institute of Mathematics and Scientific Computing, University of Graz,
Heinrichstraße 36, A-8010 Graz, Austria

Abstract. We propose and state a novel scheme for image magnification. It is formulated as a minimization problem which incorporates a data fidelity and a regularization term. Data fidelity is modeled using a wavelet transformation operator while the *Total Generalized Variation* functional of second order is applied for regularization. Well-posedness is obtained in a function space setting and an efficient numerical algorithm is developed. Numerical experiments confirm a high quality of the magnified images. In particular, with an appropriate choice of wavelets, geometrical information is preserved.

1 Introduction

We consider the problem of obtaining a high resolution image from low resolution data. This can be seen as an inverse problem, where the objective is the inversion of a downsampling operator denoted by A. This problem is ill-posed since the kernel of A is large. A standard technique to obtain well-posedness is to apply Tikhonov regularization with a regularization functional that we denote by G. The task of reconstructing a high resolution image $\hat{u}$ that fits to given low resolution image data d can then be realized by solving

$$\min_u G(u) + \mathcal{I}_{U_D}(u) \Leftrightarrow \min_{Au=d} G(u),$$

where $U_D = \{u \,|\, Au = d\}$ and $\mathcal{I}_{U_D}$ is the convex indicator function w.r.t. U_D. In order to achieve a natural-looking result, we have to make appropriate choices for the downsampling operator A and the regularization term G, both having strong influence on the obtained reconstruction quality.

Downsampling. The first question is how to describe the downsampling procedure, i.e. the process of obtaining discrete pixel values from an image u defined, for instance, on the unit square. The multiresolution approach of wavelet bases provides a framework to describe downsampling procedures: In a simple, one dimensional setting, given orthogonal scaling and wavelet functions $(\phi_{j,k})_{j,k\in\mathbb{Z}}$ and $(\psi_{j,k})_{j,k\in\mathbb{Z}}$, respectively, any signal $u \in L^2(\mathbb{R})$ can be fully described by the L^2- inner products

$$(u, \phi_{R,k})_2, \qquad (u, \psi_{j,k})_2, \qquad \text{for } j, k \in \mathbb{Z}, j \leq R,$$

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 149–160, 2013.

and for any $R \in \mathbb{Z}$ fixed. In this context, the inner products $((u, \phi_{R,k})_2)_{k\in\mathbb{Z}}$, can be interpreted to be the values of the signal u at resolution R, while the inner products $((u, \psi_{j,k})_2)_{j\leq R,k\in\mathbb{Z}}$ contain all remaining detail information. Thus, the mapping that asserts to any given signal u the inner products $((u, \phi_{R,k})_2)_{k\in\mathbb{Z}}$ can be seen as a subsampling operation to a resolution R. Since this multiresolution framework can be considered for any choice of wavelet basis, even for non-orthogonal Riesz bases, it allows a general approach to a downsampling operator for the zooming problem. Thus, given the multiresolution framework of any wavelet basis and fixed a resolution level $R \in \mathbb{Z}$, we define the linear downsampling operator to be

$$u \mapsto ((u, \phi_{R,k})_2)_{k\in\mathbb{Z}}.$$

Regularization. The second choice is the regularization term. Naturally, this choice should reflect typical properties of realistic images, in particular allow jump discontinuities. We will use the *Total Generalized Variation (TGV)* functional of second order for regularization (see [3]). As the well known *Total Variation* functional, it allows jump discontinuities in the continuous setting, but is also aware of second-order features. As a result, it favors piecewise linear reconstructions, yielding an improved image quality (see for example [1–3]).

Thus, given the scaling functions $(\phi_{R,k})_{k\in\mathbb{Z}}$, in order to obtain a high resolution image from low resolution data $d = (d_k)_{k\in\mathbb{Z}}$, our aim is to solve

$$\min_u \ \mathrm{TGV}^2_\alpha(u) + \mathcal{I}_{U_D}(u) \tag{1}$$

where

$$U_D = \{u \,|\, (u, \phi_{R,k})_2 = d_k \text{ for all } k \in \mathbb{Z}\}. \tag{2}$$

We ask for the readers patience until Section 2 for a definition of TGV^2_α.

The idea of image zooming by interpolating wavelet coefficients is not new; we refer to [12] and the references therein for an overview. However, the crucial point of these approaches is how to obtain the missing detail coefficients. In contrast to the methods in [12], we propose to use a variational technique, in particular TGV regularization, to resolve this issue.

Variational methods have already been applied for the related problem of recovering wavelet data of JPEG 2000 compressed images, missing due to transmission errors: We refer to [15] for a nonlocal TV regularized model and the references therein. Last but not least we refer to [6,13] for TV regularized zooming methods.

TGV regularization has already been applied by the authors in [2] to a similar problem setting in the context of JPEG decompression. Even though the setting of [2] also allows for TGV regularized zooming, the analytical as well as numerical framework relies on orthonormal basis transforms and thus is not applicable for general wavelet transforms.

The present paper is structured as follows: In the next section we rigorously state the minimization problem (1) in a function space setting and show existence of a solution. In the third section, we provide an algorithm for the numerical

solution and in the last section we present numerical results that illustrate the good reconstruction quality of our approach.

2 Problem Statement

Let $\Omega \subset \mathbb{R}^2$ be a bounded Lipschitz domain. The total generalized variation functional, as introduced in [3], can be defined for arbitrary order $K \in \mathbb{N}$ and is a non-trivial generalization of the total variation (TV) functional in the sense that it is equivalent to the TV functional for $K = 1$.

We are interested in the second order TGV functional, which can be defined, for $\alpha \in (\mathbb{R}^+)^2$, $u \in L^1_{\mathrm{loc}}(\Omega)$, as

$$\mathrm{TGV}^2_\alpha(u) = \sup\Big\{ \int_\Omega u \operatorname{div}^2 v \,\mathrm{d}x \,\Big|\, v \in \mathcal{C}^2_c, \|v\|_\infty \le \alpha_0, \|\operatorname{div} v\|_\infty \le \alpha_1 \Big\}. \tag{3}$$

It has been shown in [5], that TGV^2_α can equivalently be written as

$$\mathrm{TGV}^2_\alpha(u) = \min_{v \in \mathrm{BD}(\Omega,\mathbb{R}^2)} \left(\alpha_1 \| \mathrm{D}\, u - v\|_{\mathcal{M}} + \alpha_0 \|\mathcal{E} v\|_{\mathcal{M}}\right), \tag{4}$$

with D and $\mathcal{E}$ being the weak gradient and the weak symmetrized gradient, respectively, $\mathrm{BD}(\Omega,\mathbb{R}^2)$ being the set of $L^1(\Omega,\mathbb{R}^2)$ functions v such that $\mathcal{E}v$ is a finite Radon measure and $\|\cdot\|_{\mathcal{M}}$ being the Radon norm. This gives insight to the structure of TGV^2_α: Its evaluation can be interpreted as local optimal balancing between the first and the second order derivative of u, penalizing jumps in the original function as well as the derivative, but not penalizing linear ascent. Thus one would expect TGV^2_α not to suffer from first order staircasing effects, as has been confirmed in [4] in a particular setting.

The following proposition summarizes analytical properties of the TGV^2_α functional [3, 4].

Proposition 21. *Let $\Omega \subset \mathbb{R}^2$ be a bounded Lipschitz domain and $\alpha \in (\mathbb{R}^+)^2$.*

- *TGV^2_α is proper, convex and lower semi-continuous as function from $L^1(\Omega)$ to $\mathbb{R} \cup \{\infty\}$.*
- *TGV^2_α and $\mathrm{TGV}^2_{\tilde\alpha}$ are equivalent for any $\tilde\alpha \in (\mathbb{R}^+)^2$.*
- *There exist constants $c, C > 0$ such that*

$$c\left(\|u\|_1 + \mathrm{TV}(u)\right) \le \|u\|_1 + \mathrm{TGV}^2_\alpha(u) \le C\left(\|u\|_1 + \mathrm{TV}(u)\right)$$

  for any $u \in L^1(\Omega)$.
- *There exists a constant $C > 0$ such that*

$$\|u - P_1(u)\|_2 \le C\, \mathrm{TGV}^2_\alpha(u)$$

  for any $u \in L^1(\Omega)$, where P_1 is a linear projection to the space of affine functions.

These properties will allow us to obtain existence of a solution for the TGV^2_α regularized wavelet based zooming problem.

For data fidelity, in particular for the subsampling operation, we want to use an arbitrary Riesz basis [14, Section 1.8] related to a wavelet based multiresolution framework: Given a function $\phi \in L^2(\mathbb{R})$, the scaling function, it has been shown in [8] that, under certain assumptions, one can define a corresponding mother wavelet function $\psi \in L^2(\mathbb{R})$. With that, a Riesz basis of $L^2(\mathbb{R})$ can be constructed from translations and dilatations of the scaling function and the mother wavelet. We will in particular only consider scaling functions ϕ having compact support, which then results in compactly supported basis elements.

This basis can then be used to first construct a Riesz basis of $L^2((0,1))$ by applying a folding technique [9, Section 2] that corresponds to natural boundary extension. Subsequently, a Riesz basis of $L^2((0,1)\times(0,1))$ can be obtained using tensor products of the $L^2((0,1))$-basis elements, similar as in [10, Section 10.1]. Thus, given any suitable $\phi \in L^2(\mathbb{R})$, and fixed a resolution level $R \in \mathbb{Z}$, we can construct a Riesz basis of $L^2((0,1)\times(0,1))$ that will be denoted by

$$(\phi_{R,\mathbf{k}})_{\mathbf{k}\in M_R} \qquad (\psi_{j,\mathbf{k}})_{j\leq R,\mathbf{k}\in L_j}, \tag{5}$$

with $M_R, (L_j)_{j\leq R}$ finite index sets in $\mathbb{Z}^2$. Note that finiteness of those index sets is due to the folding and that $(\psi_{j,\mathbf{k}})_{j\leq R,\mathbf{k}\in L_j}$ has infinitely many elements since j is an arbitrary integer less or equal to R.

For any Riesz basis, there also exists a dual Riesz basis [14, Chapter 1], which in this setting again results from translations and dilatations of functions $\tilde{\phi}$ and $\tilde{\psi}$ [8,9], and the dual basis to (5) can be denoted by

$$(\tilde{\phi}_{R,\mathbf{k}})_{\mathbf{k}\in M_R} \qquad (\tilde{\psi}_{j,\mathbf{k}})_{j\leq R,\mathbf{k}\in L_j}. \tag{6}$$

Further, any $u \in L^2(\Omega)$, with $\Omega := (0,1)\times(0,1)$, can be written as

$$\begin{aligned} u = &\sum_{\mathbf{k}\in M_R} (u,\tilde{\phi}_{R,\mathbf{k}})_2 \phi_{R,\mathbf{k}} + \sum_{j\leq R,\mathbf{k}\in L_j} (u,\tilde{\psi}_{j,\mathbf{k}})_2 \psi_{j,\mathbf{k}} \\ = &\sum_{\mathbf{k}\in M_R} (u,\phi_{R,\mathbf{k}})_2 \tilde{\phi}_{R,\mathbf{k}} + \sum_{j\leq R,\mathbf{k}\in L_j} (u,\psi_{j,\mathbf{k}})_2 \tilde{\psi}_{j,\mathbf{k}}. \end{aligned}$$

Now assuming a low resolution image $u_0 \in \mathrm{span}\{\tilde{\phi}_{R,\mathbf{k}} | \mathbf{k} \in M_R\}$ to be given by $((u_0,\phi_{R,\mathbf{k}})_2)_{\mathbf{k}\in M_R}$, our aim is to reconstruct a high resolution image

$$u \in L^2(\Omega) = \mathrm{span}\left(\{\tilde{\phi}_{R,\mathbf{k}} | \mathbf{k} \in M_R\} \cup \{\tilde{\psi}_{j,\mathbf{k}} | j \leq R, \mathbf{k} \in L_j\}\right)$$

such that $(u,\phi_{R,\mathbf{k}})_2 = (u_0,\phi_{R,\mathbf{k}})_2$ for all $\mathbf{k} \in M_R$. This amounts to solve

$$\min_{u\in L^2(\Omega)} \mathrm{TGV}^2_\alpha(u) + \mathcal{I}_{U_D}(u), \tag{7}$$

where $U_D = \{u \in L^2(\Omega) \,|\, (u,\phi_{R,\mathbf{k}})_2 = (u_0,\phi_{R,\mathbf{k}})_2 \text{ for all } \mathbf{k} \in M_R\}$. In order to obtain well posedness of (7), we need the additional assumption that the dual

scaling basis $(\tilde{\phi}_{R,\mathbf{k}})_{\mathbf{k}\in M_R}$ is contained in $\mathrm{BV}(\Omega)$ and that at least three of the functions $(\phi_{R,\mathbf{k}})_{\mathbf{k}\in M_R}$ have support contained in Ω.

These assumptions, however, are quite weak: By using the TGV^2_α functional as regularization we implicitly assume that images are contained in $\mathrm{BV}(\Omega)$, thus it is natural to require this weak form of regularity also for the basis functions of the low resolution images. Also, one of the main points for wavelet bases to be practicable is that they are compactly supported. In that case, the support assumption is satisfied if the resolution R is sufficiently fine, i.e. if the discrete image contains sufficiently many pixels. Using the Haar wavelet, for example, this requires the image to have more that 2×2 pixels.

The support assumption is necessary due to the folding: In order to ensure that some folded functions $(\phi_{R,\mathbf{k}})_{\mathbf{k}\in M_R}$ are indeed translations of each other, their support must not intersect the boundary.

Proposition 22. *Fixed any $R \in \mathbb{Z}$, assume that the functions $(\tilde{\phi}_{R,\mathbf{k}})_{\mathbf{k}\in M_R}$ are contained in $\mathrm{BV}(\Omega)$. Further, assume that there exists $\mathbf{k}_0 = (k_0^1, k_0^2) \in M_R$ such that*

$$\mathrm{supp}(\phi_{R,k_0^1+l_1,k_0^2+l_2}) \subset \Omega$$

for all $(l_1, l_2) \in \{(0,0),(1,0),(0,1)\}$. Then, the minimization problem (7) admits a solution $\hat{u} \in \mathrm{BV}(\Omega)$.

Proof (Sketch of proof). We have that $u_0 \in \mathrm{BV}(\Omega) \cap U_D$ by being a finite linear combination of $\mathrm{BV}(\Omega)$ functions, thus the objective functional of (7) is proper (see Proposition 21) and it is non-negative. Taking $(u_n)_{n\in\mathbb{N}}$ to be a minimizing sequence, by the estimate on $\|u_n - P_1(u_n)\|_2$ as in Proposition 21, it suffices to bound $(\|P_1(u_n)\|_2)_{n\in\mathbb{N}}$ in order to bound $(\|u_n\|_2)_{n\in\mathbb{N}}$. We denote

$$(P_1(u_n))(x,y) = c_n^1 + c_n^2 x + c_n^3 y.$$

Now due to the support restriction we get that

$$\phi_{R,k_0^1+1,k_0^2}(x,y) = \phi_{R,k_0^1,k_0^2}(x-1,y) \quad \text{and} \quad \phi_{R,k_0^1,k_0^2+1}(x,y) = \phi_{R,k_0^1,k_0^2}(x,y-1).$$

Thus, denoting by $\mathtt{1}, \mathtt{x}, \mathtt{y}$ the functions mapping each (x,y) to $1, x, y$, respectively, if follows

$$\begin{aligned}
(P_1(u_n), \phi_{R,k_0^1,k_0^2})_2 &= c_n^1(\phi_{R,\mathbf{k}_0}, \mathtt{1})_2 + c_n^2(\phi_{R,\mathbf{k}_0}, \mathtt{x})_2 + c_n^3(\phi_{R,\mathbf{k}_0}, \mathtt{y})_2,\\
(P_1(u_n), \phi_{R,k_0^1+1,k_0^2})_2 &= (c_n^1 + c_n^2)(\phi_{R,\mathbf{k}_0}, \mathtt{1})_2 + c_n^2(\phi_{R,\mathbf{k}_0}, \mathtt{x})_2 + c_n^3(\phi_{R,\mathbf{k}_0}, \mathtt{y})_2,\\
(P_1(u_n), \phi_{R,k_0^1,k_0^2+1})_2 &= (c_n^1 + c_n^3)(\phi_{R,\mathbf{k}_0}, \mathtt{1})_2 + c_n^2(\phi_{R,\mathbf{k}_0}, \mathtt{x})_2 + c_n^3(\phi_{R,\mathbf{k}_0}, \mathtt{y})_2.
\end{aligned}$$

Now, by $(u_0, \phi_{R,\mathbf{k}}) = (u_n, \phi_{R,\mathbf{k}})$ for all $\mathbf{k} \in M_R$ and $n \in \mathbb{N}$, and by boundedness of $(\|u_n - P_1(u_n)\|_2)_{n\in\mathbb{N}}$, the left hand sides of these equations are bounded. An easy calculation hence yields boundedness of $((c_n^1, c_n^2, c_n^3))_{n\in\mathbb{N}}$ and, consequently, boundedness of $(\|P_1(u_n)\|_2)_{n\in\mathbb{N}}$. Thus $\|u_n\|_2$ is bounded and, since bounded sets in $L^2(\Omega)$ are relatively weakly compact, there exists a subsequence, converging to some $\hat{u} \in L^2(\Omega)$ weakly in $L^2(\Omega)$. By convexity and norm closedness of U_D we get weak closedness, thus $\hat{u} \in U_D$, and by L^1 lower semi-continuity and convexity of TGV^2_α that $\hat{u}$ is indeed a minimizer of (7).

Within the assumptions of Proposition 22 we can now freely choose a scaling function ϕ and the resulting multiresolution framework. In the following we will briefly discuss some possible choices and their interpretation.

Choice of Scaling and Wavelet Functions. For a simple interpretation of the choice of a scaling- and wavelet function, we will, for the rest of this section, go back to the unconstrained, one dimensional setting. Assuming a discrete signal to be given by $((u, \phi_{R,k})_2)_{k\in\mathbb{Z}}$, for fixed $R \in \mathbb{Z}$, i.e. $u \in \operatorname{span}\{\tilde{\phi}_{R,k} \,|\, k \in \mathbb{Z}\} \subset L^2(\mathbb{R})$, its projection onto the smaller, low resolution subspace $\operatorname{span}\{\tilde{\phi}_{R+1,k} \,|\, k \in \mathbb{Z}\}$ is described by

$$(u, \phi_{R+1,k})_2 = \sum_{l\in\mathbb{Z}} h_l (u, \phi_{R,l+2k})_2, \quad k \in \mathbb{N}, \tag{8}$$

i.e. linear filtering followed by subsampling, where the filters can be constructed from ϕ (see [8]). Similar, obtaining a higher resolution representation from low resolution data amounts to set

$$(u, \phi_{R,m})_2 = \sum_{k\in\mathbb{Z}} \left[\tilde{h}_{m-2k} (u, \phi_{R+1,k})_2 + (-1)^{m-2k} \tilde{h}_{1-(m-2k)} (u, \psi_{R+1,k})_2 \right],$$

i.e. upsampling followed by linear filtering, where again the filters can be constructed from ϕ. Not knowing the coefficients $(u, \psi_{R+1,k})_2$, a straightforward upsampling can be obtained by assuming them to be zero, thus

$$(u, \phi_{R,m})_2 \approx \sum_{k\in\mathbb{Z}} \tilde{h}_{m-2k} (u, \phi_{R+1,k})_2. \tag{9}$$

We will now interpret this approximations for different choices of scaling functions.

Haar Wavelet. A first, intuitive choice of one dimensional scaling function, from which the two dimensional scaling and wavelet functions can be obtained, would be to define

$$\tilde{\phi}(x) = \chi_{[0,1)}(x).$$

This yields the well known Haar wavelet (cf. [8, Section 6.A]), and the filters associated with ϕ and $\tilde{\phi}$ are given by

$$2^{-1/2} h_0 = 2^{-1/2} \tilde{h}_0 = \frac{1}{2}, \quad 2^{-1/2} h_1 = 2^{-1/2} \tilde{h}_1 = \frac{1}{2}.$$

Thus, the down- and upsampling as in Equations (8),(9) is given by

$$2^{-1/2} (u, \phi_{R+1,k})_2 = \frac{1}{2} \left[(u, \phi_{R,2k})_2 + (u, \phi_{R,1+2k})_2 \right],$$

$$2^{-1/2} (u, \phi_{R,2l})_2 \approx \frac{1}{2} (u, \phi_{R+1,l})_2, \quad 2^{-1/2} (u, \phi_{R,2l+1})_2 \approx \frac{1}{2} (u, \phi_{R+1,l})_2.$$

This corresponds to downsampling by averaging and upsampling by pixel repetition.

LeGall Wavelet. Another choice is to define

$$\tilde{\phi}(x) = (1+x)\chi_{[-1,0)}(x) + (1-x)\chi_{[0,1]}(x)$$

i.e. a piecewise linear scaling function. This yields the LeGall wavelet used for lossless coding in JPEG 2000 compression (cf. [8, Section 6.A]), and the filters associated with ϕ and $\tilde{\phi}$ are given by

$$2^{-1/2}\tilde{h}_0 = \frac{1}{2},\ 2^{-1/2}\tilde{h}_{\pm 1} = \frac{1}{4},\ 2^{-1/2}h_0 = \frac{3}{4},\ 2^{-1/2}h_{\pm 1} = \frac{1}{4},\ 2^{-1/2}h_{\pm 2} = -\frac{1}{8}.$$

The down- and upsampling as in Equations (8),(9) can then be given by

$$2^{-1/2}(u, \phi_{R+1,k})_2 = \frac{3}{4}(u, \phi_{R,2k})_2 + \frac{1}{4}\sum_{l=\pm 1}(u, \phi_{R,2k+l})_2 - \frac{1}{8}\sum_{l=\pm 2}(u, \phi_{R,2k+l})_2,$$

$$2^{-1/2}(u, \phi_{R,2l})_2 \approx \frac{1}{2}(u, \phi_{R+1,l})_2,$$

$$2^{-1/2}(u, \phi_{R,2l+1})_2 \approx \frac{1}{4}(u, \phi_{R+1,l-1})_2 + \frac{1}{4}(u, \phi_{R+1,l})_2.$$

This corresponds to upsampling by linear interpolation.

CDF 9/7 Wavelet. At last we will also use the CDF 9/7 wavelets, which are the basis for lossy JPEG 2000 coding, and whose filters can be found in [8, Table 6.2]. Again, the upsampling process can be seen as linear filtering, but we do not have a direct interpretation.

3 Discretization

For the discrete setting, we define $U = \mathbb{R}^{N\times N}$, $N \in \mathbb{N}$, to be the space of discrete, high resolution images, equipped with

$$\|u\|_U^2 = \sum_{0\le i,j<N} u_{i,j}^2.$$

Given scaling functions $(\phi_{j,\mathbf{k}})_{j\in\mathbb{Z},\,\mathbf{k}\in\mathbb{N}_0^2}$ and the corresponding wavelet functions $(\psi_{j,\mathbf{k}})_{j\in\mathbb{Z},\,\mathbf{k}\in\mathbb{N}_0^2}$ with their duals $(\tilde{\phi}_{j,\mathbf{k}})_{j\in\mathbb{Z},\,\mathbf{k}\in\mathbb{N}_0^2}$ and $(\tilde{\psi}_{j,\mathbf{k}})_{j\in\mathbb{Z},\,\mathbf{k}\in\mathbb{N}_0^2}$, we assume that the pixels of any discrete image $u \in U$ can be described by the coefficients

$$(u, \phi_{0,\mathbf{k}})_2, \quad 0 \le \mathbf{k} < N,$$

where $0 \le \mathbf{k} < N$ is meant component wise. For $R \in \mathbb{N}$, a low resolution image $\tilde{v}_0 \in \mathbb{R}^{(2^{-R}N)\times(2^{-R}N)}$ can then be obtained from $v_0 \in U$ by applying a discrete wavelet transform operator $W : U \to U$ and taking

$$(Wv_0)_{\mathbf{k}} = (v_0, \phi_{R,\mathbf{k}})_2, \quad 0 \le \mathbf{k} < 2^{-R}N,$$

to be its pixel values. The other way around, assuming $\tilde{v}_0 \in \mathbb{R}^{(2^{-R}N)\times(2^{-R}N)}$ to be given, one aims to find $v_0 \in U$ such that

$$(Wv_0)_{\mathbf{k}} = (\tilde{v}_0)_{\mathbf{k}}, \quad 0 \leq \mathbf{k} < 2^{-R}N,$$

i.e. an image $v_0 \in U$ such that v_0 yields $\tilde{v}_0$ when subsampled using the wavelet transform.

Thus, given a discrete image $u_0 \in \mathbb{R}^{(2^{-R}N)\times(2^{-R}N)}$ with $R \in \mathbb{N}$ and a wavelet transform operator W corresponding to scaling functions $(\tilde{\phi}_{R,\mathbf{k}})_{\mathbf{k}\in\mathbb{N}_0^2}$, the discrete data set U_D can be written as

$$U_D = \quad \{u \in U \,|\, (Wu)_{\mathbf{k}} = (u_0)_{\mathbf{k}}, \text{ for all } 0 \leq \mathbf{k} < 2^{-R}N\}. \tag{10}$$

To define the TGV^2_α functional we need the operators $\nabla : U \to U^2$, $\mathcal{E} = \frac{1}{2}(J + J^T) : U^2 \to U^3$, where ∇ and J are discrete gradient and Jacobian operators, using forward and backward differences, respectively. Motivated by the representation (4) we then define the discrete TGV^2_α functional $\mathrm{TGV}^2_\alpha : U \to \mathbb{R}$ as

$$\mathrm{TGV}^2_\alpha(u) = \min_{v\in U^2} \alpha_1 \| \nabla u - v\|_1 + \alpha_0 \|\mathcal{E}v\|_1, \tag{11}$$

with

$$\|v\|_1 = \sum_{i,j} \sqrt{(v^1_{i,j})^2 + (v^2_{i,j})^2}, \qquad \|w\|_1 = \sum_{i,j} \sqrt{(w^1_{i,j})^2 + (w^2_{i,j})^2 + 2(w^3_{i,j})^2}.$$

Note that we abuse notation by using the same symbol for a L^1 type norm on both U^2 and U^3 . Defining the spaces $X = U^3$ and $Z = U^6$, the discrete minimization problem for wavelet based zooming can be written as

$$\min_{x\in X} F(Kx) \tag{12}$$

with $F : Z \to \mathbb{R} \cup \{\infty\}$, $K : X \to Z$, defined by

$$F(v, w, r) = \alpha_1\|v\|_1 + \alpha_0\|w\|_1 + \mathcal{I}_D(r), \qquad K = \begin{pmatrix} \nabla & -\mathrm{id}_V \\ 0 & \mathcal{E} \\ W & 0 \end{pmatrix}$$

and $D = \{r \in U \,|\, (r)_{\mathbf{k}} = (u_0)_{\mathbf{k}}, \text{ for all } 0 \leq \mathbf{k} < 2^{-R}N\}$. For numerical solution of this problem, we apply a primal-dual algorithm as in [7] to the equivalent saddle point problem

$$\min_{x\in X} \max_{z\in Z} (Kx, z) - F^*(z), \tag{13}$$

with F^* the convex conjugate of F.

For this setting, the updates performed in the algorithm consist of simple arithmetic operations and the evaluation of $(\mathrm{id}_Z + \sigma\,\partial\, F^*)^{-1}(z)$ for $\sigma > 0$. By subdifferential calculus, it can be shown that this reduces to

$$(\mathrm{id}_Z + \sigma\,\partial\, F^*)^{-1}((v, w, r)) = \begin{pmatrix} P_{\{\|\cdot\|_\infty \leq \alpha_1\}}(v) \\ P_{\{\|\cdot\|_\infty \leq \alpha_0\}}(w) \\ \mathrm{assign}_{(u_0,\phi)}(r) \end{pmatrix},$$

where $P_{\{\|\cdot\|_\infty \leq \lambda\}}(y)$ is a pointwise projection of y such that $\|y\|_\infty \leq \lambda$, with $\|\cdot\|_\infty$ the dual norm of $\|\cdot\|_1$, and

$$\operatorname{assign}_{(u_0,\phi)}(r)_{i,j} = \begin{cases} r_{i,j} - \sigma(u_0, \phi_{R,(i,j)})_2 & \text{if } 0 \leq i,j < 2^{-R}N \\ 0 & \text{else.} \end{cases}$$

Global convergence of the resulting primal dual algorithm can be assured if the stepsize parameters τ, σ are such that $\sigma\tau \leq \|K\|^{-2}$, where $\|K\|$ is the norm of K as linear operator form X to Z. Even though this norm can be estimated analytically, the estimate becomes quite large especially for higher levels of wavelet decomposition, i.e. for larger zooming factors. Studying the convergence proof of the algorithm in [7, Theorem 1], one can observe that it is possible to violate $\sigma\tau\|K\|^2 < 1$ but still guarantee convergence, provided that

$$\|K(x^n - x^{n-1})\|_Z < \frac{1}{\sqrt{\sigma\tau}}\|x^n - x^{n-1}\|_X$$

is satisfied for all $n \in \mathbb{N}$ and $(x_n)_{n\in\mathbb{N}}$ the primal iterates. Ensuring this means to use only the span of the iterates for the estimation of $\|K\|$, one may hope that this allows a significantly increased stepsize. Thus we propose the following adaptive stepsize update, that allows to ensure global convergence of the primal dual algorithm:

$$\sigma_{n+1}\tau_{n+1} = S_K(\sigma_n, \tau_n) = \begin{cases} \frac{\|K(x^n - x^{n-1})\|_Z}{\|x^n - x^{n-1}\|_X} & \text{if } \theta\sigma_n\tau_n > \frac{\|K(x^n - x^{n-1})\|_Z}{\|x^n - x^{n-1}\|_X} \\ \theta\sigma_n\tau_n & \text{if } \sigma_n\tau_n > \frac{\|K(x^n - x^{n-1})\|_Z}{\|x^n - x^{n-1}\|_X} \geq \theta\sigma_n\tau_n \\ \sigma_n\tau_n & \text{if } \sigma_n\tau_n \leq \frac{\|K(x^n - x^{n-1})\|_Z}{\|x^n - x^{n-1}\|_X}. \end{cases}$$

where $0 < \theta < 1$. With that, the primal dual algorithm for solving the wavelet based zooming problem can be given as in Algorithm 1. The operants $\operatorname{div} : U^2 \to U$ and $\operatorname{div}_2 : U^3 \to U^2$ are defined as $\operatorname{div} = -\nabla^T$, $\operatorname{div}_2 = -\mathcal{E}^T$, i.e. discrete divergence operators using backward and forward differences, respectively.

As stopping criterion we use a parameter dependent modification of the primal dual gap as in [7], denoted by $\tilde{\mathcal{G}}$. Provided a good parameter choice, we obtain the estimate

$$\infty > \tilde{\mathcal{G}}(x_n, z_n) \geq \alpha_1\|\nabla u_n - v_n\|_1 + \alpha_0\|\mathcal{E}v_n\|_1 + I_{C_n}(Wu_n) - \mathrm{TGV}^2_\alpha(\hat{u}) \geq 0,$$

with $x_n = (u_n, v_n)$, $z_n = (p_n, q_n, w_n)$ being the iterates of Algorithm 1 and $\hat{x} = (\hat{u}, \hat{v})$ an optimal solution of (12). Here

$$I_{C_n}(r) = \sum_{0\leq i,j<2^{-R}N} (C_n)_{i,j}\big|r_{i,j} - (u_0, \phi_{R,(i,j)})_2\big|,$$

with $(C_n)_{i,j} = \gamma|(w_n)_{i,j}|$, $\gamma > 1$, incorporates data fidelity. Note that we cannot expect to get the estimate $\tilde{\mathcal{G}}(x_n, z_n) \geq \alpha_1\|\nabla u_n - v_n\|_1 + \alpha_0\|\mathcal{E}v_n\|_1 - \mathrm{TGV}^2_\alpha(\hat{u}) \geq 0$ since the iterates $(u_n)_{n\in\mathbb{N}}$ are only contained in the data set U_D in the limit and thus it is possible that $\alpha_1\|\nabla u_n - v_n\|_1 + \alpha_0\|\mathcal{E}v_n\|_1 < \mathrm{TGV}^2_\alpha(\hat{u})$. This was observed also in numerical experiments.

Algorithm 1. Scheme of implementation for wavelet based zooming

function TGV-ZOOM(u_0)
  $u \leftarrow W^{-1}(u_0)$
  $v \leftarrow 0, \overline{u} \leftarrow u, \overline{v} \leftarrow 0, p \leftarrow 0, q \leftarrow 0, w \leftarrow 0$
  choose $\sigma, \tau > 0$
  repeat
    $p \leftarrow \mathrm{proj}_{\alpha_1}(p + \sigma(\nabla \overline{u} - \overline{v}))$
    $q \leftarrow \mathrm{proj}_{\alpha_0}(q + \sigma(\mathcal{E}(\overline{v}))$
    $w_+ \leftarrow \mathrm{assign}_{(u_0,\phi)}(w + \sigma(W(\overline{u}))$
    $u_+ \leftarrow u - \tau(-\operatorname{div} p + W^* w_+)$
    $v_+ \leftarrow v - \tau(-p - \operatorname{div}_2 q)$
    $\overline{u} \leftarrow (2u_+ - u)$, $\overline{v} \leftarrow (2v_+ - v)$
    $\sigma_+ \tau_+ \leftarrow S_K(\sigma, \tau)$
    $u \leftarrow u_+$, $v \leftarrow v_+$, $\sigma \leftarrow \sigma_+$, $\tau \leftarrow \tau_+$
  until Stopping criterion fulfilled
  return u_+
end function

4 Numerical Experiments

Now we evaluate and compare numerical results obtained with the TGV based wavelet zooming algorithm. We will see that the algorithm performs well in general and in some situations it leads to highly improved results compared to standard zooming methods. We use Algorithm 1 to obtain the high resolution reconstruction, where we initialized the adaptive stepsizes with $\sigma = \tau = \frac{1}{3}$ and fixed the ratio between α_0 and α_1 for evaluation of the TGV functional to 4.

As stopping criterion, we require the normalized primal dual gap, $\overline{\mathcal{G}} = \tilde{\mathcal{G}}/N^2$, to be below 10^{-1} for all experiments. The purpose of using a normalized gap is to get an image size independent estimate. We tested three different wavelets, the Haar, Le Gall and CDF 9/7 wavelet as described in Section 2. For the Haar wavelet, due to orthogonality, we used a simplified version of the algorithm, similar to a the JPEG decompression algorithm presented in [2].

We first consider a four times magnification of a patch of the Barbara image, containing a stripe structure. For better comparability, we used the original image rather that a downsampled version. Thus the downsampling procedure is not kown and cannot favor any particular method, but also no ground truth is available. The results, using the three different wavelet types as well as interpolation with a Lanczos 2 [11] filter, are shown in Figure 1. As one can see, the linear filter based zooming leads to blurring of the stripes while our method yields a reconstruction appearing much sharper. Using the CDF 9/7 wavelets results in the best reconstruction quality. In particular, we observe that not only the edges are preserved, but also the geometrical information is extended in a natural manner for the CDF 9/7 wavelet (as opposed to the Haar wavelet, where "geometrical staircasing" occurs).

Next, Figure 2 shows results of the TGV based zooming method for the CDF 9/7 wavelet in the situation where the subsampling process is known and fits to the model assumption, i.e. the subsampling was done by applying a wavelet decomposition on the original image and neglecting the high resolution detail coefficients. On the left of the figure, we show the subsampled version of the image and its upsampling by setting the unknown detail coefficients to zero. This is the initial image for our TGV based method. On the right, we show the outcome of our method as the primal dual gap is below 10^{-1}. With that we compare the effect of TGV regularization independent of the wavelet basis. As one can see, indeed the reconstruction quality is clearly improved when TGV based regularization is applied, which justifies the application of TGV regularization instead of simple wavelet based upsampling. This is reflected also by an improved *Peak Signal to Noise Ratio* (PSNR) of the mean-value corrected images: The TGV based reconstruction yields a PSNR of 29.70 while the wavelet upsampling yields 29.27. The PSNR is also highly increased with respect to standard interpolation methods (Pixel repetition: 25.06, Cubic: 26.13, Lanczos 2: 26.13), however, this must be partly explained by the downsampling being done accordance with the wavelet model.

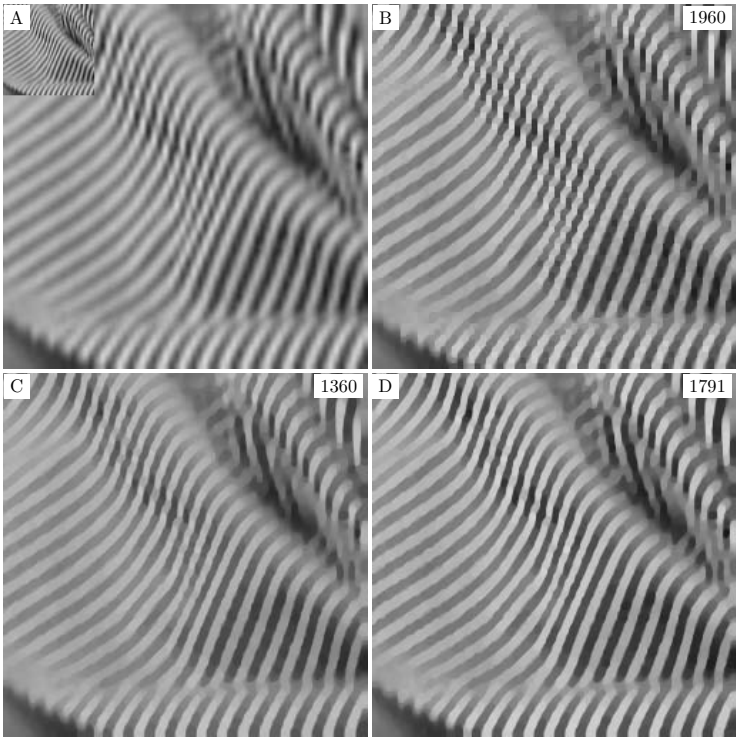


Fig. 1. A: 4 times magnification by Lanczos 2 filtering. B-D: 4 times magnification by TGV based wavelet zooming using the Haar, Le Gall and CDF 9/7 wavelet, with iteration number on top right. The stopping rule was $\overline{\mathcal{G}}(x_n, z_n) < 10^{-1}$.

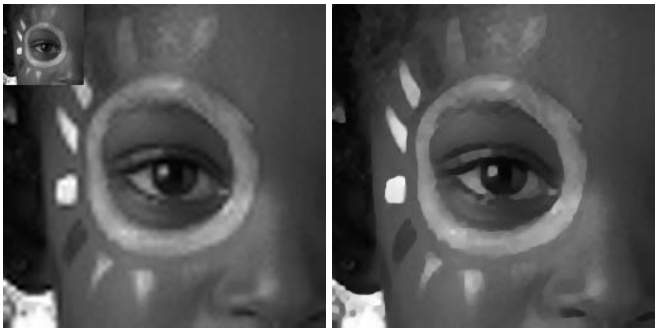


Fig. 2. Girls-eye image (256×256 pixels). Left: CDF 9/7-Wavelet down- and upsampled image (without TGV regularization). Right: TGV based wavelet upsampling using CDF 9/7 wavelet. The stopping rule was $\overline{\mathcal{G}}(x_n, z_n) < 10^{-1}$.

References

1. Bredies, K.: Recovering piecewise smooth multichannel images by minimization of convex functionals with total generalized variation penalty. SFB Report 2012-006 (2012), `http://math.uni-graz.at/mobis/publications.html`
2. Bredies, K., Holler, M.: Artifact-free decompression and zooming of JPEG compressed images with total generalized variation. In: Csurka, G., et al. (eds.) VISIGRAPP 2012. CCIS, vol. 359, pp. 242–258. Springer, Heidelberg (2013)
3. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imag. Sci. 3(3), 492–526 (2010)
4. Bredies, K., Kunisch, K., Valkonen, T.: Properties of L^1 − TGV^2: The one-dimensional case. J. Math. Anal. Appl. 389(1), 438–454 (2013)
5. Bredies, K., Valkonen, T.: Inverse problems with second-order total generalized variation constraints. In: Proceedings of SampTA, Singapore, (2011)
6. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. 20, 88–97 (2004)
7. 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)
8. Cohen, A., Daubechies, I., Feauveau, J.-C.: Biorthogonal bases of compactly supported wavelets. Commun. Pur. Appl. Math. 45(5), 485–560 (1992)
9. Cohen, A., Daubechies, I., Vial, P.: Wavelets on the interval and fast wavelet transforms. Appl. Comput. Harmon. Anal. 1(1), 54–81 (1993)
10. Daubechies, I.: Ten Lectures on Wavelets. CBMS-NSF Lecture Notes, vol. 61. SIAM (1992)
11. Duchon, C.E.: Lanczos Filtering in One and Two Dimensions. J. Appl. Meteor. 18(8), 1016–1022 (1979)
12. Kaulgud, N., Desai, U.B.: Image zooming: Use of wavelets. The International Series in Engineering and Computer Science, vol. 632, pp. 21–44. Springer (2002)
13. Malgouyres, F., Guichard, F.: Edge direction preserving image zooming: A mathematical and numerical analysis. SIAM J. Numer. Anal. 39, 1–37 (2001)
14. Young, R.M.: An Introduction to Nonharmonic Fourier Series. Academic Press (2001)
15. Zhang, X., Chan, T.F.: Wavelet inpainting by nonlocal total variation. Inverse Probl. Imag. 4, 191–210 (2010)

Anisotropic Third-Order Regularization for Sparse Digital Elevation Models

Jan Lellmann[1], Jean-Michel Morel[2], and Carola-Bibiane Schönlieb[1]

[1] DAMTP, University of Cambridge, United Kingdom
[2] CMLA, ENS Cachan

Abstract. We consider the problem of interpolating a surface based on sparse data such as individual points or level lines. We derive interpolators satisfying a list of desirable properties with an emphasis on preserving the geometry and characteristic features of the contours while ensuring smoothness across level lines. We propose an anisotropic third-order model and an efficient method to adaptively estimate both the surface and the anisotropy. Our experiments show that the approach outperforms AMLE and higher-order total variation methods qualitatively and quantitatively on real-world digital elevation data.

1 Introduction

We consider the problem of reconstructing an unknown two-dimensional height map $u : \Omega \to \mathbb{R}$ on a two-dimensional domain $\Omega \subseteq \mathbb{R}^2$, based on the values of u on a small number of level lines: $u(x) = l_i$ for $x \in C_i,\ i = 1, \dots, N$, where $C_i = u^{-1}(\{l_i\})$ are the known level lines.

This is a problem that often appears in connection with digital elevation maps (DEMs), such as in DEM reconstruction from sparse measurements or tidal coastline data. Efficient DEM reconstruction methods might also lead to more adapted compression algorithms for DEMs, although we will not consider this application here.

In this work we are particularly concerned with approaches that do not impose regularity on the level lines. This follows from the observation that in DEMs, kinks in the level lines are characteristic features of the underlying surface, and should therefore be propagated rather than removed.

We consider the following generic variational approach:

$$\min_u \{R(u), \quad u(x) = u_0(x) \text{ for } x \in C\}, \tag{1}$$

where $R(\cdot)$ is an appropriate regularizing term, and $C \subseteq \Omega$ is the set on which the data is known. This approach is slightly more generic than the reconstruction from full level lines, and can also be applied if only parts of the contours – or even only the values on a set of disjoint points – are known.

In particular, we do not require a parameterization of the level lines C_i, but rather rely on a grid discretization of the surface u only, as finding and matching such parameterizations is a major task in itself.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 161–173, 2013.

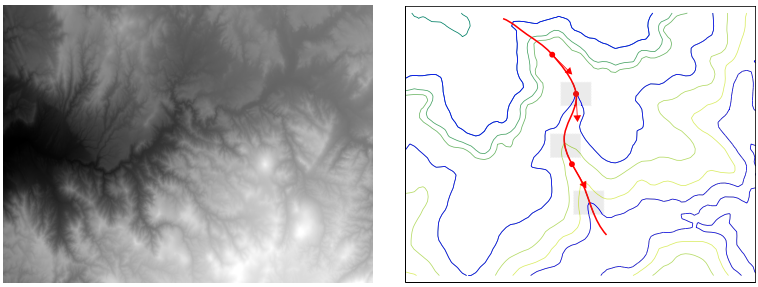


Fig. 1. Surface interpolation for digital elevation maps (DEMs). **Left:** Exemplary digital elevation map. DEMs have a unique structure which requires careful consideration when choosing a regularizer to avoid removing important features. **Right:** Surface interpolation problem. Based on the given level lines (blue) the task is to reconstruct the surface between the level lines. A particular difficulty is that level lines can have points of high curvature or even be non-smooth (marked rectangular regions), while there are generally no non-differentiabilities when *crossing* the contours along a path that associates similar points (red). The proposed approach relies on a vector field v (red arrows) that approximates the tangents to such paths in conjunction with a suitable anisotropic regularizer.

The main difficulty lies of course in the choice of the regularizer R in order to incorporate knowledge about the unique structure of DEMs (Fig. 1, left). The assumption underlying the remainder of this work is that the level lines of u can be non-smooth, but are generally "similar" to each other, i.e., points on two sufficiently close level lines can be associated with each other (Fig. 1, right). We therefore postulate the following three requirements on the reconstruction:

(P1) The surface should *coincide with the given data* on the set C.
(P2) The interpolated level lines should *preserve the geometry of the given level lines* – in particular, non-differentiabilities – as accurately as possible.
(P3) The interpolated surface should define a *smooth transition* – at least continuity of the gradient – *across* level lines (e.g, along the red path in Fig. 1).

Contribution. Based on these requirements and motivated by the recent success of higher-order total variation models, we discuss different choices for the regularizer R based on L^1-norms of second- and third-order derivatives.

We demonstrate that for an interpolation algorithm to fulfil the above requirements (P2)-(P3) a *third-order* regularizer R is needed, and moreover it is necessary to include *directional* information, i.e., anisotropy, in the form of an *auxiliary vector field* v that incorporates information about the relation between adjacent level lines. We propose an efficient method to approximate the unknown vector field v for a known surface u as the direction in which the *normals of the level lines change least* (Sect. 2).

The performance of the method on synthetic examples suggests that the proposed method satisfies (P1)–(P3), and moreover that the surface u and the directional vector field v can be efficiently jointly estimated. We conclude by

a quantitative comparison on real-world DEM data against AMLE and higher-order total variation methods (Sect. 3).

Related Models. In the literature on surface interpolation two main streams of methods can be found. The first one is the explicit parameterization of given level lines of the surface with subsequent pointwise matching and interpolation steps [12,16,15].

One standard method to construct DEMs from a given set of level lines is to use Geodesic Distance Transformations [19,20]. Here the interpolant between two level contours is constructed pointwise as the linear interpolation between their level values with respect to the geodesic distance.

A major drawback of such contour-based methods is that they require an explicit parameterization of the given level lines, which may require a substantial amount of preprocessing, intermediate reparameterisation, or may even fail in the presence of scattered and sparse surface data. Furthermore, they generally do not enforce a continuity of the slope across the level lines. For these reasons, we shall not consider them here.

Our proposed approach belongs to the second methodological stream: surface interpolation based on processing the surface as a function of height over a domain in $\mathbb{R}^2$.

One of the most successful and most widely used interpolation approaches within this class is the PDE-based *absolutely minimizing Lipschitz extension (AMLE)* interpolation method [3,6]. AMLE interpolation is a diffusion-based interpolation method that has been very successfully applied to the interpolation of elevation maps [2]. An interpolant u of height values ϕ given on the boundary of a hole $\Omega \subset \mathbb{R}^2$ is computed as a viscosity solution of

$$D^2u\left(\frac{Du}{|Du|}, \frac{Du}{|Du|}\right) = 0 \quad \text{in } \Omega, \quad u|_{\partial\Omega} = \phi, \tag{2}$$

where the quadratic form $D^2u(\cdot,\cdot)$ is defined as $D^2u\,(x,y) = \sum_{i,j} x_i x_j \frac{\partial^2 u}{\partial x_i \partial x_j}$.

As proved in [6], AMLE interpolation is able to interpolate data given in isolated points and on level lines. This property distinguishes AMLE from simpler PDE-based interpolation approaches such as the Laplace equation, and makes it an ideal candidate for surface interpolation. However, level lines of AMLE interpolants are smooth: in [18], Savin proved that a solution of (2) is C^1-regular in two space dimensions, which makes the perfect reconstruction of sharp cusps and kinks in a surface impossible.

Another drawback of AMLE interpolation is that it cannot interpolate slopes of a surface. In order to extend a PDE-interpolator like (2) to take into account gradient information as well, one requires to introduce fourth-order differential operators into the equation. Among others, the *thin plate spline* interpolator is one of the simplest fourth-order surface interpolation models, see [17,8,14,9,5] for instance. There, the interpolated surface is constructed by solving

$$\Delta^2 u = 0 \text{ in } \Omega, \quad u = \phi \text{ on } \partial\Omega, \quad \frac{\partial u}{\partial n} = \psi \text{ on } \partial\Omega, \tag{3}$$

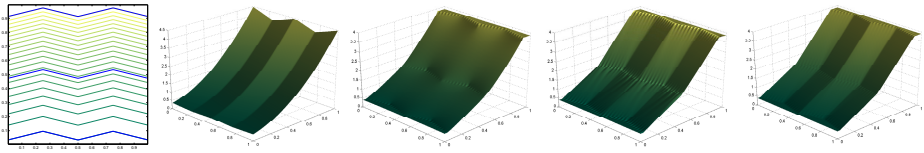


Fig. 2. Surface interpolation on synthetic data using quadratic interpolation and AMLE. The input contours are marked in blue and were prescribed on 25% of their points. Quadratic interpolation by solving Laplace's equation smoothes out the level lines; characteristic features such as the non-differentiabilities along the ridges are lost. AMLE preserves such features but does not cope very well with the sparsity of the data. With full data, AMLE introduces less artifacts but generates an additional kink on the middle blue level curve with prescribed value.

where ϕ and ψ are the given height and the gradient of the surface in normal direction to the curve $\partial\Omega$, respectively. While this model allows to incorporate both grey values and gradient information into the interpolation process, the interpolated surface is generally too smooth, still not preserving sharp surface features (see also Fig. 3 below).

PDE-based interpolation approaches such as (2) and (3) are closely related to certain types of statistical interpolation procedures. A standard technique within this framework is Kriging [13,7,11,21]. Here, the interpolated surface is defined as a realization of a random field, of which a finite number of values at some sites of $\mathbb{R}^2$ is fixed. For a detailed account on surface interpolation methods and their interrelations we refer to [1].

2 Anisotropic Higher-Order Regularizers

Proposed Model. We would first like to point out some observations to illustrate the points made in the introduction and to motivate the specific choice of an anisotropic third-order regularizer:

Non-differentiabilities in the level lines should be preserved. To motivate (P2), consider the synthetic example in Fig. 2, where the level lines of the ground truth are piecewise linear with quadratic spacing. The example demonstrates that preserving and extrapolating non-smooth features on the level lines according to (P2) is important for a good visual quality of the result: classic inpainting using $R(u) = \int_\Omega \|Du\|_{L^2}$ (i.e., solving the Laplace equation) results in smooth level lines. The example in Fig. 2 clearly shows that such features need to be preserved to obtain a good reconstruction.

The model should be able to cope with partial or sparse data. Data where only parts of the lines are known is common, as for example when extracting level lines from satellite images or individual measurements. Such data is hard to deal with using models that are based on matching and interpolation of explicit parameterizations of the contours, as obtaining the parameterizations then becomes a major problem. This motivates the "wholistic" variational approach (1). AMLE

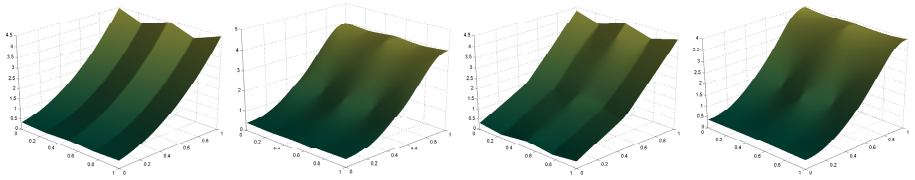


Fig. 3. Effect of *isotropic* second- and third-order TV-based regularization. Although less pronounced than in the quadratically regularized (cf. Fig. 2) and thin-plate model, isotropic higher-order regularization enforces too much smoothness and motivates the introduction of anisotropy.

seems to be prone to introducing artifacts when data is sparse or partly missing, and fails to correctly extrapolate data outside the region defined through the prescribed level lines (Fig. 2).

Non-smooth second-order models tend to introduce kinks. With full data on the level lines, AMLE performs better but introduces a sharp bend at the middle prescribed level line, i.e., a line where the slope changes. This is a typical feature of non-smooth second-order models such as (2). We refer to the experimental section for a comparison.

Isotropic higher-order regularization generates too much smoothness. One obvious choice for the regularizer is to use third-order total variation-based regularization by setting $R(u) = \int_\Omega \|D^3 u\|$. Unfortunately this enforces too much smoothness on u, as can be seen in Fig. 3; even second-order regularization is much too smooth.

In order to address the above issues, we propose to introduce an auxiliary vector field $v : \Omega \to \mathbb{R}^2$ and consider anisotropic models of the form

$$R_1^{(3)}(u) := \int_\Omega \|D^3 u(v,\cdot,\cdot)\|, R_2^{(3)}(u) := \int_\Omega \|D^3 u(v,v,\cdot)\|, R_3^{(3)}(u) := \int_\Omega \|D^3 u(v,v,v)\|, \tag{4}$$

where $D^3 u$ are the third-order derivatives of the height map u, and $\|\cdot\|$ refers to the usual Euclidean norm. In full generality, $D^3 u$ can be defined as a measure on the Borel functions from Ω to $\mathbb{R}^{2\times 2\times 2}$, see [4] for the technical details. For smooth functions, $D^3 u(x)$ is the $2 \times 2 \times 2$ tensor of all third-order partial derivatives.

The dot-notation refers to partial specialization: restricting ourselves to sufficiently smooth functions for simplicity, $D^3 u(v,\cdot,\cdot)(x)$ is a 2×2 matrix describing the derivative of the Hessian in the direction of v, $D^3 u(v,v,\cdot)(x)$ is the 2-dimensional vector of the second derivatives in the direction of v of the gradient, and $D^3 u(v,v,v)(x)$ is the (scalar) third derivative of u in the direction of v.The difference between the regularizers in (4) is thus the level of anisotropy: even for a constant vector field $v = (1, 0)$, the regularizer $R_1^{(3)}$ still includes some mixed derivatives, while $R_3^{(3)}$ uses purely derivatives in the direction of v.

In a similar manner we define the second-order anisotropic regularizers $R_1^{(2)} := \int_\Omega \|D^2 u(v,\cdot)\|$ and $R_2^{(2)} := \int_\Omega \|D^2 u(v,v)\|$. The isotropic regularizers $\mathrm{TV}^{(3)} :=$

$\int_\Omega \|D^3 u\|$ and $\mathrm{TV}^{(2)} := \int_\Omega \|D^2 u\|$ are just the usual second- and third-order total variation regularizers.

Crucially, the vector field v associates points on neighbouring level lines. More precisely, we assume that any path through v, i.e., a path $C : [c_1, c_2] \to \mathbb{R}^2$ with $c_1 < c_0 < c_2$ and $C(c_0) = x$ and tangents $v(C(c))$ relates the point x to matching points on other level curves (see Fig. 1).

As an example, consider $u(x_1, x_2) = x_1^2 - x_2$, where the level curves are parabolas translated along the x_2 axis. For $x = (x_1, 0)$, the path $C(c) = (0, x_1^2 - c)$ returns the point corresponding to x on the level set for level c, and we can set $v(x) = (0, -1)$.

The rationale behind this choice for v is precisely (P3): smoothness is desirable, but only *across* the level lines (P2). The vector field v gives meaning to the rather vague definition of "across". It is not obvious which of the variants is the best choice, therefore we refer to the experimental section for a discussion of their qualitative differences.

The Auxiliary Vector Field. Finding v is generally a difficult problem, since it requires matching corresponding points on different level lines. In this work, we propose to use for $v(x)$ the *direction in which* $Du/|Du|$ *changes least*. Under the simplifying assumption that the level lines are only translates of each other and have non-zero curvature, this allows to correctly recover v (note that $Du/|Du|$ are the normals of the level curves).

With the notation $A = D(Du/|Du|)(x) \in \mathbb{R}^{2\times 2}$, the vector $v(x) \in \mathbb{R}^2$ can be found by finding $w \in S^1$ minimizing $\|Aw\|_2$. This amounts to computing the basis vector associated with the minimal singular value of the 2×2 matrix A, for which a closed-form solution is available. In order to increase robustness we add a convolution with a Gaussian kernel K_σ with (small) variance σ^2 and obtain

$$v(x) = \arg \min_{w, \|w\|_2 = 1} \|K_\sigma * (D(Du/|Du|))(x)\, w\|_2. \tag{5}$$

While this gives good results when the level lines are sufficiently curved, the vector field tends to show erratic behaviour on straight sections of the level lines: if $Du/|Du|$ is locally almost constant, small variations in u can lead to random jumps in v due to the normalization to unit length. Therefore we solve an additional quadratic minimization problem to ensure that v is sufficiently smooth:

$$\min_{v'} \frac{1}{2} \int_\Omega w(x) \|v'(x) - v(x)\|_2^2 dx + \frac{\rho}{2} \int_\Omega \|Dv'(x)\|_2^2 dx. \tag{6}$$

While many choices for the weights w are conceivable, we found that the most robust is to set $w(x)$ to the largest singular value of $K_\sigma * (D(Du/|Du|))(x)$. This ensures that the smoothing is increased in areas where u is almost planar, and decreased in regions where the level lines have large curvature and v is therefore most likely accurate.

The solution of problem (6) can be easily found by solving a system of linear equations, and an additional normalization step ensures that the vectors v have unit length. In all our experiments we used a 9×9 convolution kernel with $\sigma = 2$.

A subtle difficulty when applying (6) is that, while the regularizers in (4) are invariant with respect to sign changes of v, problem (6) is not. We counter this by normalizing the vector field v so that $\langle v(x), Du(x) \rangle \leqslant 0$, i.e., v always points towards the negative gradient of u. While slightly heuristic, this scheme seems to work remarkably well in practice, and avoids having to solve more difficult non-convex optimization problems involving unit length constraints.

For unknown u, we start with a random field v^0 and alternate between minimizing (1) to find u^{k+1} from v^k and computing v^{k+1} from u^k as outlined above. Note that choosing v randomly approximately corresponds to using the isotropic regularizers $R_0^{(3)}$ or $R_0^{(2)}$, but has the additional advantage of introducing randomness that can help to solve ambiguous situations (see below).

3 Experimental Results

We used the MOSEK commercial interior-point package to solve the fully assembled problem. The examples were solved in less than one minute per outer iteration on an Intel Core 2 Duo 2.66 GHz with 4 GB of memory.

Fixed Directions. The first question to answer is which of the existing and proposed schemes performs best with respect to the requirements postulated in (P1)–(P3). In order to separate this aspect from the issue of finding the vector field v, we performed several experiments on synthetic examples with the directions v set to a known ground-truth.

In Fig. 4 we compare different levels of anisotropy on a "pyramid" example. The challenge comes from the fact that the contours around the tip cannot be interpolated between two given level lines, but must be extrapolated, preserving the non-smoothness in the level lines. Since in the ground truth the level lines are scaled copies of each other, the vector field v can be explicitly computed as $v(x) = (x - x_0)/\|(x - x_0)\|_2$ with center $x_0 = (1/2, 1/2)$.

It can be seen that the regularizers with a higher level of directionality generally perform better at reconstructing the pointed pyramid tip. However it should be noted that this example is not very typical for digital elevation maps, where a smoother result such as the one obtained using $\|D^3 u(v, \cdot, \cdot)\|$ is more likely appropriate. We also observed that higher levels of directionality seem to result in harder optimization problems. This results in longer computation times and sometimes less precise solutions with a slight smoothing effect.

The isotropic regularizers $R_0^{(3)}$ and $R_0^{(2)}$ enforce too much smoothness. The second-order methods tend to introduce sharp bends along the given contours due to their preference for piecewise planar surfaces.

Adaptive Directions. Figure 5 shows that in the case of the "pyramid" example, the vector field v can be effectively found through the iterative procedure outlined in Sect. 2, starting at random directions. The results are visually almost indistinguishable from the results in Fig. 4 that were computed with known ground truth v. We found that the number of required updates for v is very low, usually the result as well as v were stationary after five outer iterations.

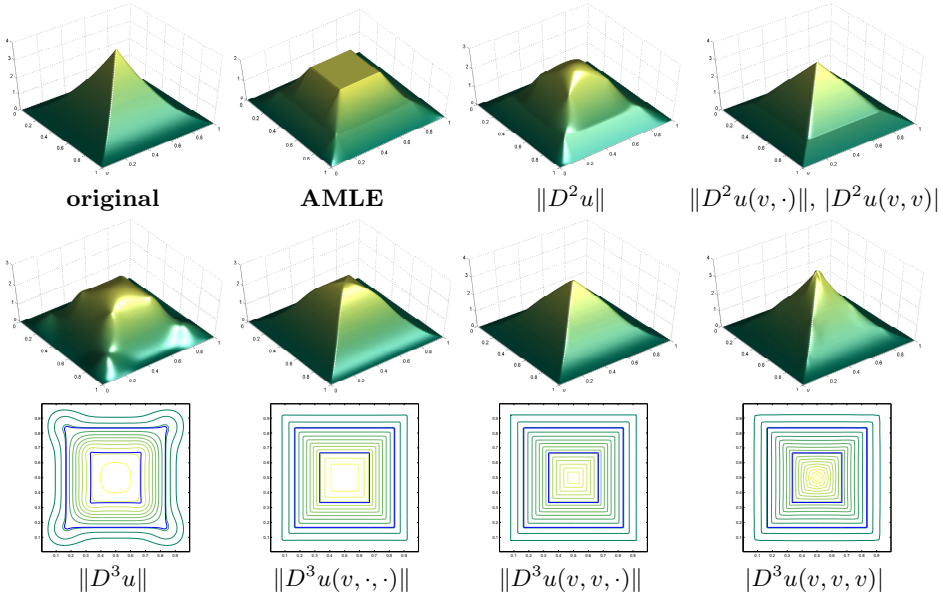


Fig. 4. Comparison of different notions of anisotropy with known directions v. The input consists of the contour lines marked in blue including the boundary of the domain, and contains a level line around a region with a local maximum. In consequence, this is an example for a problem that cannot be solved by pointwise interpolation between contour lines. AMLE does not extrapolate the tip, and introduces kinks along the given contours. The non-directional approaches result in smoothed-out contours. In contrast, the directional methods do not smooth the level lines, but they still regularize – as desired – the *spacing* of the contours to a varying amount.

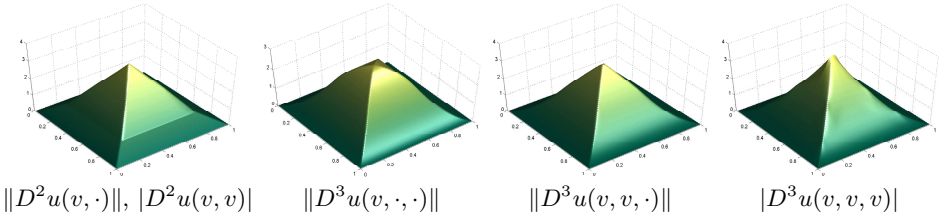


Fig. 5. Reconstruction of the pyramid example in Fig. 4 using adaptive adjustment of the vector field v after 50 iterations. The surfaces are visually identical to the results in Fig. 4, where v was set to the (known) ground truth.

In Fig. 6 we show a more challenging example that was deliberately chosen so that the solution is ambiguous. This highlights a particular issue with the isotropic regularizers $R_0^{(3)}$ and $R_0^{(2)}$: as their overall energy is convex, even assuming that both solutions are in fact minimizers of the isotropic energy, all convex combinations of these solutions also have to be minimizers. Therefore the result is an undesirable mixture of both solutions. The additional vector field v and the randomness introduced in the first step effectively resolve the ambiguity.

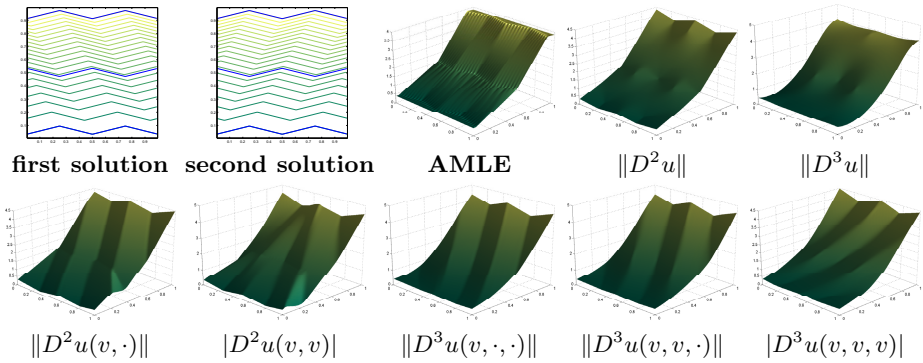


Fig. 6. Reconstruction of the "ambiguity" example using adaptive adjustment of the vector field v with 50 outer iterations. The input contours (blue) allow two equally good exact solutions. The non-directional approaches are entirely convex, and cannot be expected to pick one of the two solutions; AMLE fails equally. The second-order methods perform slightly better but introduce artificial non-differentiabilities. Using third-order methods the ambiguity is resolved and one of the possible solutions is correctly reconstructed.

In both examples third-order regularizers performed superior to second-order methods. We attribute this to the tendency of second-order methods to generate planar patches, which greatly aggravates the problem of reliably computing v as in (5). Again the algorithm settles quickly on one of the possible solutions and converges in less than 10 outer iterations.

We would like to emphasize that all the above examples were constructed using a minimal amount of level lines and specifically in order to highlight characteristics of the regularizer. On real-world data, the effects are generally much less pronounced.

Reconstruction of Digital Elevation Maps. Figure 7 shows the performance of the proposed approach on real-world DEM data extracted from the National Elevation Dataset (NED) [10]. The input consists of 10 contour lines with equally spaced heights, and contains approximately 5% of the original data points. We only compare the result of the anisotropic approach with $\|D^3u(v,\cdot,\cdot)\|$, as the other anisotropic approaches performed slightly worse.

The different approaches show remarkably similar behaviour as on the synthetic data: The nondirectional approaches generated overly smooth solutions. AMLE does not reconstruct the mountain peak correctly and introduces artifacts along the slopes and ridges. The directional third-order method gives a clean result and reconstructs most prominent features.

Quantitative Evaluation. In order to quantify the performance of the various methods, we compared the results on the DEM data to the known ground truth. Since the L^2-distance is not necessarily a good measure to judge visual quality, we also computed the error between the *normal fields* of the surfaces.

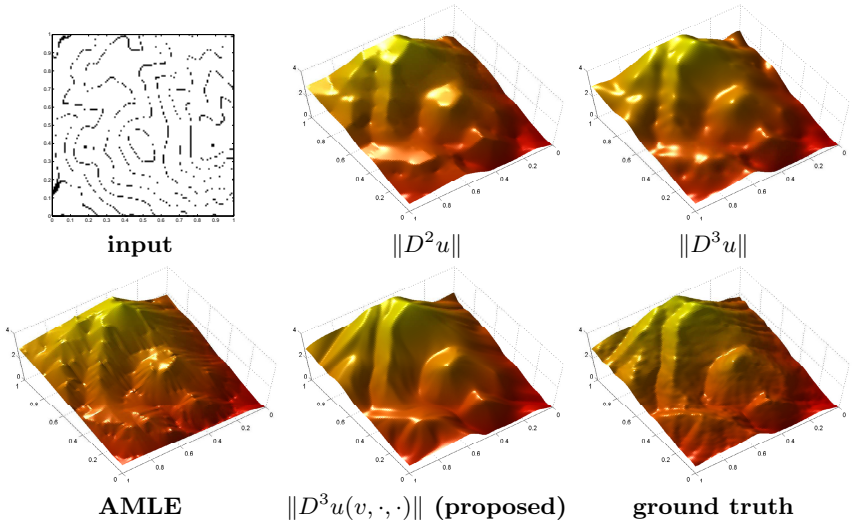


Fig. 7. Reconstruction of real-world digital elevation maps. **Top row:** input contours, second-order isotropic total variation, third-order isotropic total variation. **Bottom row:** AMLE, proposed method using $\|D^3u(v,\cdot,\cdot)\|$, ground truth. AMLE does not correctly recover the small peak and tends to hallucinate features. The proposed method correctly recovers the mountain tops and ridges (top left corner).

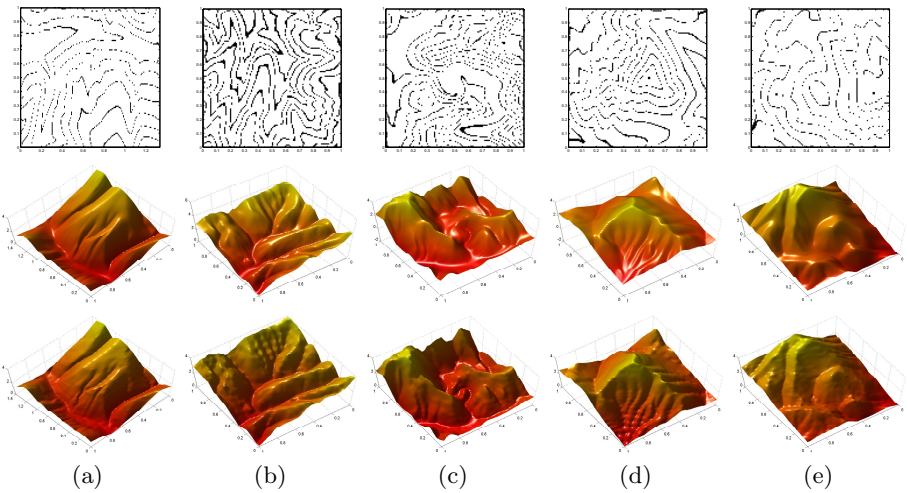


Fig. 8. Reconstruction of real-world digital elevation maps. **Top row:** input contours. **Middle row:** results using the proposed method with $\|D^3u(v,\cdot,\cdot)\|$. **Bottom row:** ground truth.

Table 1. Solution quality for example (a) in Fig. 8, measured in the absolute L^2 difference and L^2 distance of the normals (in parentheses). The best results for each method are marked in bold. Even without selecting the optimal smoothness parameter, good quality solutions can be obtained. The overall best result is achieved with the third-order anisotropic regularizer $R_1^{(3)}$ and $\rho = 1$ (underlined).

ρ	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}
Lapl.	144. (11.05)	144. (11.05)	144. (11.05)	144. (11.05)	144. (11.05)	144. (11.05)	144. (11.05)
AMLE	140. (15.58)	140. (15.58)	140. (15.58)	140. (15.58)	140. (15.58)	140. (15.58)	140. (15.58)
$\mathrm{TV}^{(3)}$	15.34 (3.52)	15.34 (3.52)	15.34 (3.52)	15.34 (3.52)	15.34 (3.52)	15.34 (3.52)	15.34 (3.52)
$R_1^{(3)}$	9.72 (2.41)	9.73 (2.41)	9.75 (2.40)	9.62 (2.30)	**8.55** (**2.15**)	9.83 (2.68)	15.48 (3.85)
$R_2^{(3)}$	11.84 (2.64)	12.11 (2.71)	12.61 (2.68)	13.75 (2.64)	**10.88** (**2.45**)	11.31 (3.21)	18.83 (4.81)
$R_3^{(3)}$	**10.17** (3.14)	10.58 (3.11)	10.68 (**2.93**)	17.96 (3.17)	15.14 (3.02)	14.75 (3.88)	28.13 (5.94)
$\mathrm{TV}^{(2)}$	31.07 (4.39)	31.07 (4.39)	31.07 (4.39)	31.07 (4.39)	31.07 (4.39)	31.07 (4.39)	31.07 (4.39)
$R_1^{(2)}$	**15.82** (**2.99**)	16.17 (3.01)	17.40 (3.09)	18.67 (3.12)	18.29 (3.35)	18.76 (4.16)	52.13 (7.28)
$R_2^{(2)}$	12.06 (3.46)	**11.09** (**3.33**)	14.69 (3.69)	21.35 (4.49)	22.87 (5.47)	30.55 (7.41)	70.46 (11.96)

Table 2. Quantitative evaluation for the data set in Fig. 8, measured in L^2 distance and L^2 distance of the normals (in parentheses). The proposed third-order directional regularizer consistently performed best (bold), followed by the second-order directional regularizers and AMLE.

data set	1	2	3	4	5
Lapl.	141.57 (10.88)	95.55 (21.73)	202.06 (29.96)	79.33 (13.28)	103.76 (9.86)
AMLE	138.15 (15.33)	83.91 (22.94)	235.86 (42.91)	56.36 (11.39)	88.91 (10.36)
$\mathrm{TV}^{(3)}$	15.10 (3.46)	13.61 (6.66)	29.61 (10.67)	18.93 (5.34)	34.47 (5.09)
$R_1^{(3)}$	**8.45** (**2.12**)	**11.10** (**5.77**)	**28.75** (**10.49**)	**8.59** (**3.31**)	**17.03** (**3.15**)
$R_2^{(3)}$	10.99 (2.43)	13.41 (7.04)	33.42 (12.64)	10.34 (3.74)	21.96 (3.74)
$R_3^{(3)}$	14.46 (2.95)	21.28 (10.85)	50.23 (19.12)	13.74 (4.76)	26.23 (4.65)
$\mathrm{TV}^{(2)}$	30.59 (4.32)	25.29 (9.45)	84.71 (15.32)	26.04 (6.78)	43.84 (5.77)
$R_1^{(2)}$	18.80 (3.35)	19.18 (9.07)	43.34 (15.02)	13.35 (4.44)	23.10 (4.41)
$R_2^{(2)}$	25.64 (5.49)	28.74 (13.71)	97.27 (21.74)	21.41 (6.63)	28.53 (5.68)

Table 1 shows the performance of the various approaches under varying smoothness parameter ρ. We found that the relative performance is almost independent of the choice of ρ, with the $R_0^{(3)}$ regularizer always being in the lead in both performance measures. For this example and all other DEM data that were tested, we found that setting $\rho = 1$ is nearly optimal.

To obtain more representative data on the relative performance, we evaluated the methods on a set of details from the NED (Figure 8) that include various qualitatively different features such as bifurcating and meandering valleys, sharp ridges, and several styles of mountain peaks. Again we set $\rho = 1$ for all of the examples. The results in Table 2 show that on all examples and across all performance measure, the directional $R_1^{(3)}$ regularizer worked best, followed by directional third-order-, directional second-order-, isotropic third-order-, isotropic second-order regularization, and finally AMLE.

4 Conclusion

In our view the reconstruction of digital elevation maps is a very interesting application for higher-order regularization, where minor variations in the regularizer can

have large effects on the result. In this work we left out most questions concerning the analysis such as how to choose suitable function spaces and the existence of solutions and fixed points. All of these seem to be challenging questions, and we leave them to future work.

From a practical viewpoint, the numerical results are very encouraging, and suggest that the directional third-order regularizers together with the proposed method of estimating the directional field v converge rapidly and give excellent results on synthetic as well as real-world data.

Acknowledgments. The authors would like to thank Andrea Bertozzi and Alex Chen for helpful discussions. This publication is based on work supported by Award No. KUK-I1-007-43, made by King Abdullah University of Science and Technology (KAUST), EPSRC first grant No. EP/J009539/1, EPSRC/Isaac Newton Trust Small Grant, and Royal Society International Exchange Award No. IE110314. J.-M. Morel was supported by MISS project of Centre National d'Etudes Spatiales, the Office of Naval Research under Grant N00014-97-1-0839 and by the European Research Council, advanced grant "Twelve labours".

References

1. Almansa, A.: échantillonnage, interpolation et détection. applications en imagerie satellitaire. Technical report, ENS Cachan (2002)
2. Almansa, A., Cao, F., Gousseau, Y., Rouge, B.: Interpolation of digital elevation models using AMLE and related methods. Geoscience and Remote Sensing 40, 314–325 (2002)
3. Alvarez, L., Guichard, F., Lions, P.L., Morel, J.M.: Axioms and fundamental equations of image processing. Arch. Rational Mech. 123, 199–257 (1993)
4. Ambrosio, L., Fusco, N., Pallara, D.: Functions of Bounded Variation and Free Discontinuity Problems. Clarendon Press (2000)
5. Carr, J.C., Fright, W.R., Beatson, R.K.: Surface interpolation with radial basis functions for medical imaging. Trans. Med. Imaging 16(1), 96–107 (1997)
6. Caselles, V., Morel, J.-M., Sbert, C.: An axiomatic approach to image interpolation. Trans. Image Proc. 7(3), 376–386 (1998)
7. Cressie, N.: Statistics for Spatial Data. Wiley, New York (1993)
8. Duchon, J.: Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. R.A.I.R.O. Anal. Numér. 10, 5–12 (1976)
9. Franke, R.: Scattered data interpolation: Test of some methods. Math. Comput. 38, 181–200 (1982)
10. Gesch, D., Evans, G., Mauck, J., Hutchinson, J., Carswell Jr., W.J.: The national map – elevation. U.S. Geological Survey Fact Sheet 3053 (2009)
11. Journel, A.G., Huijbregts, C.J.: Mining Geostatistics. Academic (1978)
12. Masnou, S., Morel, J.: Level lines based disocclusion. In: 5th IEEE Int'l Conf. on Image Processing, Chicago, IL, October 4-7, pp. 259–263 (1998)
13. Matheron, G.: La théorie des variables régionalisées, et ses applications. Technical Report 5, Les Cahiers du Centre de Morphol. Math. de Fontainebleau (1971)
14. Meinguet, J.: Surface Spline Interpolation: Basic Theory and Computational Aspects. In: Approximation Theory and Spline Functions, Dordrecht, Holland, pp. 124–142 (1984)

15. Meyer, T.: Coastal elevation from sparse level curves. Summer project under the guidance of T. Wittman, A. Bertozzi, and A. Chen, UCLA (2011)
16. Meyers, D., Skinner, S., Sloan, K.: Surfaces from contours. Trans. on Graphics 11(3), 228–258 (1992)
17. Mitas, L., Mitasova, H.: Spatial Interpolation. Wiley (1999)
18. Savin, O.: C^1 regularity for infinity harmonic functions in two dimensions. Arch. Ration. Mech. Anal. 176(3), 351–361 (2005)
19. Soille, P.: Spatial distributions from contour lines: An efficient method based on distance transformations. J. Vis. Commun. Image Represent. 2(2), 138–150 (1991)
20. Soille, P.: Generalized Geodesic Distances Applied to Interpolation and Shape Description. In: Mathematical Morphology and its Applications to Image Processing. Kluwer (1994)
21. Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging. Springer (1999)

A Fast Algorithm for Exact Histogram Specification. Simple Extension to Colour Images

Mila Nikolova

CMLA CNRS ENS Cachan
61 av. du Président Wilson, 94235 Cachan Cedex, France
nikolova@cmla.ens-cachan.fr
http://mnikolova.perso.math.cnrs.fr/

Abstract. In [12] a variational method using $\mathcal{C}^2$-smoothed ℓ_1-TV functionals was proposed to process digital (quantized) images so that the obtained minimizer is quite close to the input image but its pixels are all different from each other. These minimizers were shown to enable exact histogram specification outperforming the state-of-the-art methods [6], [19] in terms of *faithful total strict ordering*. They need to be computed with a high numerical precision. However the relevant functionals are difficult to minimize using standard tools because their gradient is nearly flat over vast regions.

Here we present a specially designed fixed-point algorithm enabling to attain the minimizer with remarkable speed and precision. This variational method applied with the new proposed algorithm is actually the best way (in terms of quality and speed) to order the pixels in digital images. This assertion is corroborated by exhaustive numerical tests.

We extend the method to color images where the luminance channel is exactly fitted to a prescribed histogram. We propose a new fast algorithm to compute the modified color values which preserves the hue and do not yield gamut problem. Numerical tests confirm the performance of the latter algorithm.

Keywords: Color image enhancement, Exact histogram specification, Fast smooth convex nonlinear minimization, Fixed point algorithm, Gamut preservation, Hue preservation, Minimizer analysis, Smoothed ℓ_1-TV functionals, Total strict ordering, Variational methods.

1 Introduction

Histogram processing is a technique with numerous applications. The goal of exact histogram specification (HS) is to transform an input image into an output image having a prescribed histogram. Histogram equalization (HE) is a particular case of HS. Among the applications of HS let us mention invisible watermarking, image normalization and enhancement, object recognition [7], [5], [16]. Let f be an input $M \times N$ digital image with L gray values. The set of values

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 174–185, 2013.

of f is denoted by[1] $\mathcal{Q} = \{q_1, \cdots, q_L\}$. To simplify the notation we reorder the image columnwise into a vector of size $n := MN$ and address the pixels by the index set[2] $\mathbb{I}_n := \{1, \cdots, n\}$. The histogram of f, denoted by h_f, is given by $h_f[q_k] = \sharp\{i \in \mathbb{I}_n \mid f[i] = q_k\}$, $\forall\, k \in \mathbb{I}_L$, where $\sharp$ stands for cardinality.

Exact HS is straightforward for images whose pixels values are all different from each other. However exact HS (and also exact HE) is an ill-posed problem for digital (quantized) images since the number of pixels[3] n is much larger than number the possible intensity levels L [6] ,[17]. The clue to achieving *exact* HS is to obtain a *meaningful* total strict ordering of all pixels in the input digital image. Research on this problem has been conducted for four decades already [8]. The Local Mean (LM) method of Coltuc, Bolon and Chassery [6], the wavelet-based approach (WA) of Wan and Shi in [19] and the specialized variational approach (SVA) of Nikolova, Wen and Chan [12] are the state-of-the-art methods. For any input pixel $f[i]$ in the input digital image f these methods extract K auxiliary information, say $a_k[i]$, $k \in \mathbb{I}_K$, based on f. For simplicity, we set $a_0 := f$. Then an ascending order "$\prec$" for all pixels is sought using the rule

$$i \prec j \quad \text{if} \quad f[i] \leq f[j] \quad \text{and} \quad a_k[i] < a_k[j] \quad \text{for some} \quad k \in \{0, \cdots, K\}. \tag{1}$$

The numerical results in [12] have shown that SVA clearly outperforms its main competitors—LM and WA—in terms of quality and memory requirements but not in speed. In section 3 we derive a specialized fixed point minimization algorithm that attains the minimizer with remarkable speed and precision. Convergence and parameter selection are briefly discussed. Numerical tests confirm that the SVA method along with the new FP algorithm outperforms by far all other relevant sorting methods.

In section 4 we focus on HS for color digital images. Extension of gray scale HS to color images is a quite complex task. As usual, a color image has three components: red (R), green (G) and blue (B). Applying HS to each color channel independently changes the hue of the image [17]. To avoid this problem, several ways to define a 3-D color histogram were proposed, e.g. [18], [10]. Recently, Han et al. [9] showed that these methods increase the brightness of the image and cannot fit the prescribed (uniform) histogram. In the same article, the authors propose to equalize the luminance (intensity) component of the image and apply the hue-preserving transformation proposed by Naik and Murthy [11] to assign the new color values. There are many methods that rely on modification of the histogram of the luminance component and deduce the needed change in the RGB space, see e.g. [2], [1], [16]. Our approach is to produce a correct template for the luminance part by HS. To compute the color components, we propose a new algorithm preserving the hue and the gamut, and ensuring that the resultant luminance component fits the specified histogram. The new algorithm share the same simplicity as the one used in [9] but provides much better results.

[1] For 8-bit images we have $L = 256$ and $\mathcal{Q} = \{0, \cdots, 255\}$.
[2] In what follows, $\mathbb{I}_m := \{1, \cdots, m\}$ for any integer m.
[3] E.g. for an 1024×1024 8-bit image we have $n = 1048576 \gg 256 = L$.

2 The Specialized Variational Approach (SVA)

The functionals proposed in [12] are of the form

$$J(u,f) := \Psi(u,f) + \beta\Phi(u), \quad \beta > 0 \tag{2}$$

with

$$\begin{aligned} \Psi(u,f) &:= \sum_{i\in\mathbb{I}_n} \psi(u[i]-f[i]), \\ \Phi(u) &:= \sum_{i\in\mathbb{I}_r} \varphi(g_i u) \end{aligned} \tag{3}$$

where $g_i \in \mathbb{R}^{1\times n}$, $i \in \mathbb{I}_r$ correspond to a forward discretization. More precisely,

– If only vertical and horizontal differences are considered

$$\begin{array}{llll} g_i[i]=-1, & g_i[i+1]=1 & \text{and} & g_i[k]=0 \quad \forall\, k\in\mathbb{I}_n\setminus\{i,i+1\}, \\ g_j[j]=-1, & g_j[j+M]=1 & \text{and} & g_i[k]=0 \quad \forall\, k\in\mathbb{I}_n\setminus\{j,j+M\}; \end{array} \tag{4}$$

– If diagonal differences are added, $\Phi(u)$ is nearly rotationally invariant and

$$\begin{array}{llll} g_i[i]=-1, & g_i[i+M-1]=1 & \text{and} & g_i[k]=0 \quad \forall\, k\in\mathbb{I}_n\setminus\{i,i+M-1\}, \\ g_j[j]=-1, & g_j[j+M+1]=1 & \text{and} & g_j[k]=0 \quad \forall\, k\in\mathbb{I}_n\setminus\{j,j+M+1\}. \end{array}$$

In both cases, Neumann or periodic boundary conditions are adopted. We denote

$$G = \left[g_1^T, \cdots, g_r^T\right]^T \in \mathbb{R}^{r\times n} ,$$

where the superscript T stands for transposed.

The functions $\psi(\cdot) := \psi(\cdot,\alpha_1) : \mathbb{R}\to\mathbb{R}$ and $\varphi(\cdot) := \varphi(\cdot,\alpha_2) : \mathbb{R}\to\mathbb{R}$ depend on two parameters $\alpha_1 > 0$ and $\alpha_2 > 0$, respectively. When necessary, we shall use the notation $\psi(\cdot,\alpha_1)$ and $\varphi(\cdot,\alpha_2)$. The functions ψ and ϕ in (3) belong to the *family of functions* $\theta(\cdot,\alpha) : \mathbb{R}\to\mathbb{R}$, $\alpha > 0$, satisfying the conditions H1 and H2 described below. We denote $\theta'(t,\alpha) := \frac{d}{dt}\theta(t,\alpha)$, and similarly for θ''.

H 1 *For any $\alpha > 0$ fixed, $\theta(\cdot,\alpha)$ is $\mathcal{C}^s$-continuous for $s \geq 2$, even—i.e. $\theta(-t,\alpha) = \theta(t,\alpha)$—and meets*

$$t \in \mathbb{R} \quad \Rightarrow \quad \theta''(t,\alpha) > 0 \ .$$

Note that by H1, for any α fixed, $t \to \theta'(t,\alpha)$ *is strictly increasing in t*. Further,

H 2 *For any $\alpha > 0$ given, $\theta'(t,\alpha)$ is upper bounded*[4] *and for $t > 0$ fixed, it is strictly decreasing in $\alpha > 0$ with*

$$\begin{aligned} \alpha > 0 \quad &\Rightarrow \quad \lim_{t\to\infty} \theta'(t,\alpha) = 1 \ , \\ t \in \mathbb{R} \quad &\Rightarrow \quad \lim_{\alpha\to 0} \theta'(t,\alpha) = 1 \quad \textit{and} \quad \lim_{\alpha\to\infty} \theta'(t,\alpha) = 0 \ . \end{aligned}$$

Table 1. Relevant choices for $\theta(\cdot,\alpha)$ obeying H1 and H2. When $\alpha > 0$ decreases towards zero, $\theta(\cdot,\alpha)$ becomes stiff near the origin.

	θ	θ'	θ''
f1	$\sqrt{t^2+\alpha}$	$\dfrac{t}{\sqrt{t^2+\alpha}}$	$\dfrac{\alpha}{\left(\sqrt{t^2+\alpha}\right)^3}$
f2	$\alpha \log\left(\cosh\left(\dfrac{t}{\alpha}\right)\right)$	$\tanh\left(\dfrac{t}{\alpha}\right)$	$\dfrac{1}{\alpha}\left(1-\left(\tanh\left(\dfrac{t}{\alpha}\right)\right)^2\right)$
f3	$\lvert t\rvert - \alpha \log\left(1+\dfrac{\lvert t\rvert}{\alpha}\right)$	$\dfrac{t}{\alpha+\lvert t\rvert}$	$\dfrac{\alpha}{(\alpha+\lvert t\rvert)^2}$

Under these assumptions, the functional $J(\cdot,f)$ in (2)-(3) is clearly a fully smoothed ℓ_1-TV model. Good choices for θ meeting H1 and H2 are given in Table 1.

Remark that θ'' is even, positive and its upper bound is finite and

$$\|\theta''\|_\infty = \theta''(0) > 0.$$

2.1 Preliminary Facts

Using H1 and H2, the properties listed below play a role in what follows.

1. For any $\beta > 0$ and any f, $J(\cdot,f)$ has a unique minimizer $\hat{u}$ [12, Proposition 1].
2. For any $\beta > 0$ and any f living in a *dense open subset of* $\mathbb{R}^n$, say $\mathbb{K}^n$, the minimizer $\hat{u}$ of $J(\cdot,f)$ satisfies [12, Theorem 1]

$$\begin{array}{ll} \hat{u}[i] \neq \hat{u}[j], & \forall\, i,j \in \mathbb{I}_n, \quad i \neq j; \\ \hat{u}[i] \neq f[i], & \forall\, i \in \mathbb{I}_n. \end{array} \tag{5}$$

   However, all digital images with L gray values (like f) belong to a subset $S^n_{\mathcal{Q}}$ which is closed and of null Lebesgue measure in $\mathbb{R}^n$. Using some results from number theory, the conclusion drawn in [12, sect. 2, Remark (b)] is that $\sharp\,(\mathbb{K}^n \cap S^n_{\mathcal{Q}})/\sharp\, S^n_{\mathcal{Q}}$ should be a number close to zero[5]. Then the minimizer $\hat{u}$ of $J(\cdot,f)$ for $f \in S^n_{\mathcal{Q}}$ satisfies (5) with a very high probability. Thus $\hat{u}$ provides the auxiliary information to strictly order the pixels in f using (1).
3. Sice $\psi'(\cdot,\alpha)$ is $\mathcal{C}^{s-1}$ and odd, it has an inverse function

$$\xi(\cdot,\alpha_1,) := (\psi')^{-1}(\cdot,\alpha) : (-1,1) \to \mathbb{R}\ , \tag{6}$$

   which is also odd, strictly increasing and $\mathcal{C}^{s-1}$ (inverse functions theorem).
4. For any $y \in (0,1)$, the function $\alpha \mapsto \xi(y,\alpha)$ is strictly increasing on $(0,+\infty)$ [3, Lemma 2].

[4] The upper bound of θ' is set to 1 only for definiteness.

[5] Note that no reasonable sorting algorithm can order strictly the pixels of *all* digital images. E.g., the pixels of a constant image should not be ordered in a strict way.

5. Let us denote $\eta := \|G\|_1$. If $\beta\eta < 1$ then [3, Theorem 1]

$$\|\hat{u} - f\|_\infty \leq \xi\big(\beta\eta, \alpha_1\big)$$

and $\alpha_1 \mapsto \xi\big(\beta\eta, \alpha_1\big)$ is strictly increasing on $(0, +\infty)$.
6. Further, $\|\hat{u} - f\|_\infty \nearrow \xi\big(\beta\eta, \alpha_1\big)$ as $\alpha_2 \searrow 0$ [3, Theorem 2].

3 A Fast Sorting Algorithm

3.1 Semi-explicit Formula for the Minimizer

The unique minimizer $\hat{u}$ of $J(\cdot, f)$ satisfies $\nabla J(\hat{u}, f) = 0$ where the gradient ∇ is taken with respect to the first variable, namely u. From the definition of J in (2) this is equivalent to $\nabla\Psi(\hat{u}, f) = -\beta\nabla\Phi(\hat{u})$. Using (3), we have

$$\frac{d\Psi(u, f)}{du[i]} = \psi'(u[i] - f[i]) \quad \text{and} \quad \frac{d\Phi(u)}{du[i]} = \sum_{j \in \mathbb{I}_r} \varphi'(g_j u) g_j[i]. \tag{7}$$

Thus the minimizer $\hat{u}$ satisfies

$$\psi'(\hat{u}[i] - f[i]) = -\beta \sum_{j \in \mathbb{I}_r} \varphi'(g_j \hat{u}) g_j[i], \quad \forall\, i \in \mathbb{I}_n.$$

Using the notation in (6), the latter equations are equivalent to

$$\hat{u}[i] = f[i] + \xi\left(-\beta \sum_{j \in \mathbb{I}_r} \varphi'(g_j \hat{u}) g_j[i]\right), \quad i \in \mathbb{I}_n. \tag{8}$$

The inverse function $\xi(y, \alpha) = (\theta')^{-1}(y, \alpha)$ in (6) has an explicit expression for f1, f2 and f3 in Table 1. This function and its derivative $\xi' := \frac{d}{dy}\xi(y, \alpha)$ is given in Table 2. Note that $\xi'(\cdot, \alpha)$ is even and strictly increasing on $[0, 1)$.

Table 2. The inverse function $\xi(y, \alpha) = (\theta')^{-1}(y, \alpha)$ in (6) and its derivative ξ' with respect to y for all functions in Table 1

	ξ	ξ'
f1	$y\sqrt{\dfrac{\alpha}{1-y^2}}$	$\dfrac{\sqrt{\alpha}}{(\sqrt{1-y^2})^3}$
f2	$\dfrac{\alpha}{2}\ln\dfrac{1+y}{1-y}$	$\dfrac{\alpha}{1-y^2}$
f3	$\dfrac{\alpha y}{1-\lvert y\rvert}$	$\dfrac{\alpha}{1-\lvert y\rvert}$

3.2 A Fixed Point (FP) Algorithm to Minimize J

The proposed algorithm uses (8) and Table 2. The iterations are given by

$$u_{k+1} = \mathcal{X}(u_k), \tag{9}$$
$$\mathcal{X}(u) := f + \xi\big(-\beta\nabla\Phi(u)\big), \tag{10}$$

where the function ξ, given in Table 2, is applied componentwise and $u_0 = f$.

Theorem 1. *Let α_1, α_2 and β be chosen so that $\beta\eta < 1$ and*

$$\beta\,\xi'(\beta\eta)\,\varphi''(0,\alpha_2)\,\|G^T G\|_\infty < 1, \tag{11}$$

where η is defined in 5, subsection 2.1. Then the iteration (9)-(10) *converges.*

Sketch of the proof. For any $\alpha_2 > 0$,

$$0 < \varphi''(g_i u,\, \alpha_2) \le \varphi''(0,\alpha_2) \quad \forall\, i \in \mathbb{I}_r.$$

Further, one derives

$$\|\nabla\mathcal{X}(u)\|_\infty \le \beta\xi'(\beta\eta)\|G^T \mathrm{diag}\big(\varphi''(g_i u)\big)G\|_\infty \le \beta\xi'(\beta\eta)\varphi''(0)\|G^T G\|_\infty.$$

Then $\|\nabla\mathcal{X}(u)\|_\infty < 1$. The spectral radius of $\nabla\mathcal{X}(u)$ meets $\rho(\nabla\mathcal{X}(u)) \le \|\nabla\mathcal{X}(u)\|_\infty$. Since $\mathcal{X}$ has a fixed point by (8), Ostrowski theorem [13] entails the result. □

Some practical values of the parameters ensuring convergence are given next.

- If G corresponds only to (4), then $\eta = \|G\|_1 = 4$ and $\|G^T G\|_\infty = 8$. For ψ and φ given by f1 in Table 1 and $\alpha_1 = 0.05$, $\alpha_2 = 0.3$ and $\beta = 0.1$ we have

$$\|\nabla\mathcal{X}(u)\|_\infty \le 0.7745 \quad \text{and} \quad \|\hat{u} - f\|_\infty \lessapprox 0.0976.$$

- If G corresponds to jointly (4) and (2), $\eta = \|G\|_1 = 8$ and $\|G^T G\|_\infty = 16$. For ψ and φ given by f1 and $\alpha_1 = 0.02$, $\alpha_2 = 0.4$ and $\beta = 0.07$ we have

$$\|\nabla\mathcal{X}(u)\|_\infty \le 0.6963 \quad \text{and} \quad \|\hat{u} - f\|_\infty \lessapprox 0.0956.$$

Remark 1. When initialized with a nonconstant image, the iteration (9)-(10) provides fast convergence even if (11) is not satisfied. One of the reason is that for many differences we have $\varphi''(g_i u) > 0$ in which case $\|G^T \mathrm{diag}\big(\varphi''(g_i u)\big)G\|_\infty \ll \varphi''(0)\|G^T G\|_\infty$. And when $\|\mathcal{X}(u_0) - \mathcal{X}(u_1)\| < 1$, then the iteration converges (see [15, p. 142]). Another reason is that $\rho(\nabla\mathcal{X}(u))$ is quite smaller than $\|\nabla\mathcal{X}(u)\|_\infty$ and so under the condition in (11), $\rho(\nabla\mathcal{X}(u))$ is quite smaller than 1.

3.3 Comparison with the State-of-the-Art Sorting Algorithms

The variational method provides one auxiliary information which is the minimizer $\hat{u}$ of $J(\cdot, f)$, i.e. $a_1[i] = \hat{u}[i]\ \forall\, i \in \mathbb{I}_n$ and ordering is obtained by (1). As in [12], $J(\cdot, f)$ was used with G corresponding to (4) and

$$\psi(t) = \sqrt{t^2 + \alpha_1}, \quad \varphi(t) = \sqrt{t^2 + \alpha_2}, \quad \alpha_1 = \alpha_2 = 0.05 \quad \text{and} \quad \beta = 0.01.$$

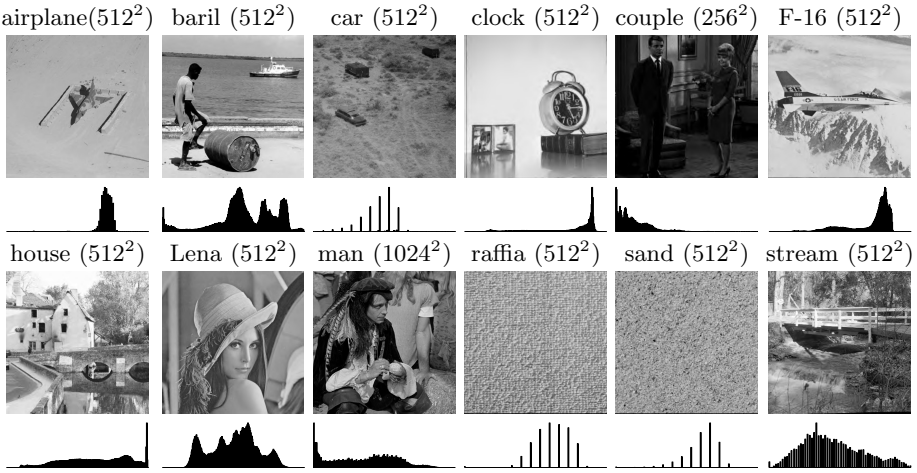


Fig. 1. All 12 digital 8 bit images used to compare the algorithms and their histograms. The gray values of these images belong to $\{0, \cdots, 255\}$.

We ran the Polak-Ribière (PR) CG minimization with stopping rule given by $\|J(u_k, f)\|_\infty \leq 10^{-6}$ and limiting the iteration number to 35, as in [12]. Our FP algorithm was applied with stopping rule $\|J(u_k, f)\|_\infty \leq 10^{-6}$. Our method was compared with the local mean (LM) algorithm [6] for $K = 6$ and with the wavelet-based algorithm (WA) [19] for Haar wavelet for $K = 9$. These values of K were recommended by the authors. The experiments were performed using a PC DELL Latitude E6220 with an Intel Core i7-2640M, 2.8 GHZ processor and 8 GB of RAM under Windows 7, using MATLAB v. 7.11.0.584, 64-bit.

Here we present sorting results on 12 digital images with various sizes and content, with gray values in $\{0, \cdots, 255\}$. The images and their histograms are shown in Fig. 1. Note that most of these histograms are quite singular.

Remark 2. Since our parameter choice guarantees that $\|\hat{u} - f\|_\infty < 0.1$ (see 5 in subsection 2.1), ordering the pixels according to (1) amounts just to sort $(\hat{u} + f)$.

This fact was not noticed in [12] where (1) was used directly which is computationally heavier. For the LM and WA methods, (1) must be applied for $K = 6$ and $K = 9$ images, respectively, which requires much more memory and computation than the SVA method [12] where $K = 1$.

The pixel ordering provided by the PR minimization in [12] and the new FP algorithm should be the same since the obtained PSNR values for HE inversion are the same (these experiments are not presented here). The experiments in [12, section 5] have shown that SVA outperforms by far the LM and WA methods in terms of PSNR in restoration of contrast compression and in HE inversion. The proposed FP minimization scheme gives rise to a much shorter CPU time and a more than 5 times better numerical precision. For fair comparison of the numerical schemes, Remark 2 was not used to generate the results in Table 3.

Table 3. Comparison with the state-of-the-art algorithms. Fail denotes the percentage of pixels that could not be sorted in a strict way. CPU is in seconds.

	Fail %			CPU				**SVA**	
Image	LM	WA	**SVA**	LM	WA	SVA PR	SVA **FP**	$\|\nabla J\|_\infty \times \mathbf{10^{-7}}$ PR	$\|\nabla J\|_\infty \times \mathbf{10^{-7}}$ **FP**
airplane	5.30	17.70	0.00	2.85	6.29	2.54	2.07	45.70	6.06
baril	0.17	0.24	0.00	2.15	5.34	2.37	1.92	3.83	6.45
car	7.84	19.91	0.00	3.67	6.51	2.28	2.22	5.96	7.30
clock	1.57	4.52	0.00	1.05	2.20	0.69	0.44	4.47	6.92
couple	2.50	3.30	0.00	0.94	2.11	0.53	0.37	6.68	7.38
F-16	0.18	0.57	0.00	2.48	5.21	5.21	2.11	63.19	7.22
houseB	0.36	1.58	0.00	1.62	5.40	2.40	2.37	15.02	5.71
Lena	0.00	0.20	0.00	2.84	4.91	4.57	1.67	58.95	5.81
man	0.34	0.68	0.00	7.58	15.68	9.24	9.38	29.13	7.65
raffia	13.66	16.05	0.00	2.71	6.99	2.37	1.87	10.60	5.43
sand	12.62	15.21	0.00	3.82	6.63	4.82	2.45	68.90	5.80
stream	0.41	0.75	0.00	2.75	4.98	2.34	2.29	6.08	7.22
means	2.97	5.28	0.00	2.19	4.46	2.46	1.79	20.23	4.88

When Remark 2 is applied, the mean CPU time for the SVA-FP algorithm is reduced to **1.54 sec.** *In terms of faithful total strict ordering and CPU time, the SVA with the proposed FP scheme and using the simple ordering rule in Remark 2 provides the best results.* The same conclusion was drawn on a test on 50 eight-bit gray-value images downloaded from http://sipi.usc.edu/database/.

4 HS for Color Images

4.1 Our Approach

Let $w = (w_1, w_2, w_3)$ be an input color image where w_1, w_2 and w_3 are its red, green and blue channels, respectively. Let $\zeta = (\zeta_1, \cdots, \zeta_L)$ be the prescribed histogram. As in the previous section, we consider that all w_k's are reordered columnwise as n-length vectors. The luminance of w is [4]

$$f = \frac{1}{3}(w_1 + w_2 + w_3) \in \mathbb{R}^n.$$

With the help of the ordering algorithm described in section 3, f is transformed into $\widehat{f} \in \mathbb{R}^n$ whose histogram is exactly ζ. We need a color image $\widehat{w}$ such that

$$\frac{1}{3}(\widehat{w}_1 + \widehat{w}_2 + \widehat{w}_3) = \widehat{f} \tag{12}$$

and satisfying the classical requirements:

(c1) $\widehat{w}$ has the same hue as w;
(c2) to avoid gamut problem, $0 \leq \widehat{w}_k[i] \leq L-1$, $\forall\, i \in \mathbb{I}_n$, $k = 1, 2, 3$.

It is well known (and easy to verify) that the hue of a pixel $w[i]$ is guaranteed to be preserved in the restored $\widehat{w}[i]$ only if $\widehat{w}[i]$ is obtained from $w[i]$ using an affine transform [4], [11].

Our method to compute the color channels from w and $\widehat{f}$ consists in a "slight" but important modification of the method proposed in [11] and used in [2], [9], among others. It is composed of a forward step followed by a correction step.

Algorithm for color assignment

(step 1) Compute $F \in \mathbb{R}^n$ according to

$$F[i] := \frac{\widehat{f}[i]}{f[i]} \quad \forall\, i \in \mathbb{I}_n$$

and assign

$$\widehat{w}_k[i] = F[i]\, w_k[i] \quad k = 1, 2, 3 \quad \forall\, i \in \mathbb{I}_n. \tag{13}$$

(step 2) Find the set

$$J := \{i \in \mathbb{I}_n \mid \widehat{w}_1[i] > L-1 \text{ or } \widehat{w}_2[i] > L-1 \text{ or } \widehat{w}_3[i] > L-1\}. \tag{14}$$

Compute $C \in \mathbb{R}^{\sharp\, J}$ by

$$C[i] := \frac{L-1-\widehat{f}[i]}{L-1-f[i]} \quad \forall\, i \in J$$

and correct the pixels in J as

$$\widehat{w}_k[i] = L-1-C[i]\,(L-1-w_k[i]) \quad k = 1, 2, 3 \quad \forall\, i \in J. \tag{15}$$

It is easy to check that that both modifications in (13) and (15) satisfy (12). A value $F[i] > 1$ means that the color at pixel i should be enhanced, i.e., $|w_k[i] - w_{k'}[i]| < |\widehat{w}_k[i] - \widehat{w}_{k'}[i]|$, $k \neq k'$, $k, k' \in \{1, 2, 3\}$. So we wish to keep the maximum number of pixels computed using (13). Some of them (quite a few in practice) will fail the constraint (c2)—these pixels form the set J in (14). Their value will be properly corrected at the correction step 2.

Remark 3. In the scheme of Naik and Murthy [11] (and the one of [16]) , step 1 is applied only if $F[i] \leq 1$ and in all other cases step 2 is applied. We can note that their strategy is quite conservative. For instance, if $w[i] = (10, 30, 50)$ and $F[i] = 5$, noticing that $f[i] = 30$, their strategy yields $\widehat{w}[i] = (140.67, 150, 159.33)$ which results in a nearly gray-value pixel. Instead, our approach yields $\widehat{w}[i] = (50, 150, 250)$, so the color is enhanced and the constraint (c2) is satisfied.

The computational cost of the algorithm of Naik and Murthy was analysed in [9]. The conclusion was that the computational complexity is proportional to the number of pixels. In all experiments, we observed that in mean 3.5 % of the pixels go through step 2. So the computational cost of our color assignment algorithm is nearly the same.

4.2 Numerical Results

Here we compare our algorithm for color assignment with the algorithm proposed in [11] and used in [9]. For fair comparison of the color assignment algorithms, *in all cases we used our sorting algorithm (section 3)*. Exhaustive comparison with other algorithms for HE of color images can be found in [9].

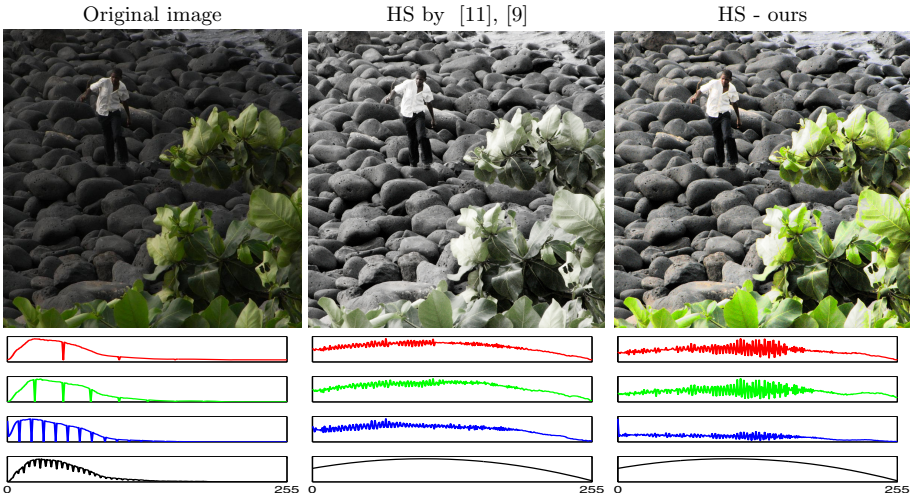


Fig. 2. Images and their histograms—R (■), G (■), B (■), luminance (■)

The original image ($800 \times 800 \times 3$) in Fig. 2 is underexposed and has a poor contrast. HE often produces overly enhanced unnatural looking images. The target histogram was chosen according to general recommendations of commercials in image processing (seen on You Tube). It is exactly satisfied and can be seen on the last row of the histograms of the restored images (in black). The image obtained by [11], [9] suffers from being too gray. This confirms our Remark 3. Using our algorithm (13)-(15)—only 4.7 % of the pixels needed the correction step 2. The image quality is really improved.

In Fig. 3 the original image ($750 \times 1000 \times 3$) seems nearly gray-valued. Following [1], the prescribed histogram is a linear combination of the histogram of the input image and a uniform histogram—see the last row of the histograms of the restored images. The image obtained by [11], [9] is almost gray-valued. Our method enables us to recover all colors. In this case, only 1.27 % of the pixels had to be rescaled using step 2.

In the original image in Fig. 4 ($1000 \times 1000 \times 3$) there is a snake that is not easy to distinguish from the surrounding landscape. Our goal was to modify the histogram so that the snake is clearly seen. This is the reason why we chose as target histogram the curve on the bottom row of the restored images. For our algorithm, only 3.3% of the pixels were reprocessed by step 2.

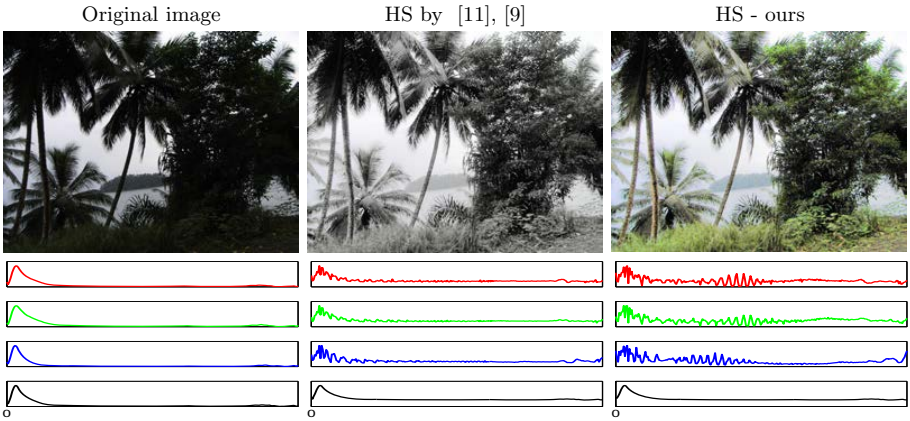


Fig. 3. Images and their histograms—R (■), G (■), B (■), luminance (■)

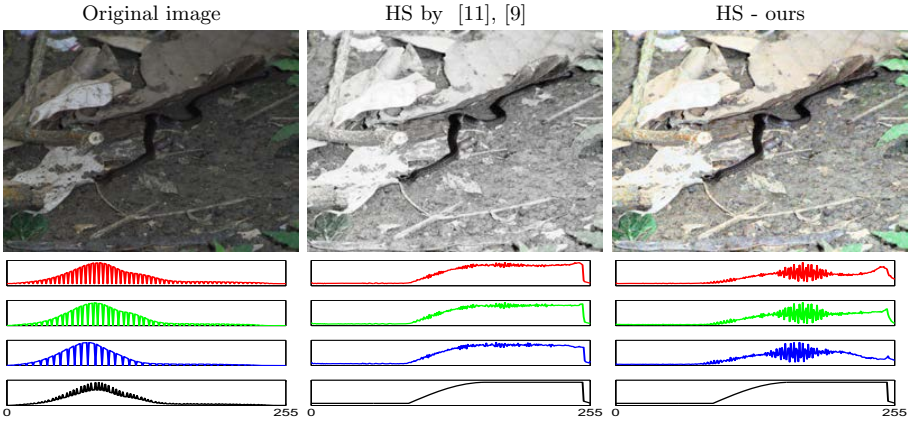


Fig. 4. Images and their histograms—R (■), G (■), B (■), luminance (■)

5 Conclusions and Perspectives

The sorting algorithm proposed in section 3 is for the present the best one. The proposed algorithm for color assignment in section 4 is fast and yields better results than the one used in [9]. However it does not exploit color perceptual facts that were used e.g. in [14]—but with an intensive computational cost. This point deserves further exploration.

References

1. Arici, T., Dikbas, S.: A histogram modification framework and its application for image contrast enhancement. IEEE Trans. on Image Proc. 18, 1921–1935 (2009)
2. Bassiou, N., Kotropoulos, C.: Color image histogram equalization by absolute discounting back-off. Computer Vision and Image Understanding 107 (2007)
3. Bauss, F., Nikolova, M., Steidl, G.: Fully smoothed ℓ_1-TV models: Bounds for the minimizers and parameter choice. J. of Math. Imaging and Vision (2013)
4. Berns, R.S.: Billmeyer and Saltzman Principles of Color Technology, 3rd edn. Willey & Sons, Roy S (2000)
5. Caselles, V., Lisani, J.L., Morel, J.M., Sapiro, G.: Shape preserving local histogram modification. IEEE Trans. on Image Processing 8, 220–229 (1999)
6. Coltuc, D., Bolon, P., Chassery, J.-M.: Exact histogram specification. IEEE Trans. on Image Processing 15, 1143–1152 (2006)
7. Gonzalez, R., Woods, R.: Digital Image Processing. Addison-Wesley (1993)
8. Hall, E.L.: Almost uniform distributions for computer image enhancement. IEEE Transactions on Computers C-23, 207–208 (1974)
9. Han, J.H., Yang, S., Lee, B.U.: A novel 3-D color histogram equalization method with uniform 1-D gray scale histogram. IEEE Trans. on Image Proc. 20, 506–512 (2011)
10. Menotti, L., Najman, L., de Araújo, A., Facon, J.: A fast hue-preserving histogram equalization method for color image enhancement using a bayesian framework. In: Proc. 14th Int. Workshop Syst., Signal Image Process, pp. 414–417 (2007)
11. Naik, S.F., Murthy, C.A.: Hue-preserving color image enhancement without gamut problem. IEEE Trans. on Image Processing 12, 1591–1598 (2003)
12. Nikolova, M., Wen, Y., Chan, R.: Exact histogram specification for digital images using a variational approach. J. of Mathematical Imaging and Vision (2012)
13. Ortega, J., Rheinboldt, W.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York (1970)
14. Palma-Amestoy, R., Provenzi, E., Bertalmio, M., Caselles, V.: A perceptually inspired variational framework for color enhancement. IEEE Trans. on Pattern Analysis and Machine Intelligence 31, 458–474 (2009)
15. Schwartz, L.: Analyse: Topologoe générale et analyse fonctionnelle, Hermann, Paris (1993)
16. Sen, D., Sankar, P.: Automatic exact histogram specification for contrast enhancement and visual system based quantitative evaluation. IEEE Trans. on Image Processing 20, 1211–1220 (2011)
17. Solomon, C., Breckon, T.: Fundamentals of Digital Image Processing. A Practical Approach with Examples in Matlab, 1st edn. John Wiley & Sons (2011)
18. Trahanias, P.E., Venetsanopoulos, A.: Color image enhancement through 3-D histogram equalization. In: Proc. 15th Int. Conf. Pattern Recognit., vol. 1, pp. 545–548 (1997)
19. Wan, Y., Shi, D.: Joint exact histogram specification and image enhancement through the wavelet transform. IEEE Trans. on Image Processing 16, 2245–2250 (2007)

Constrained Sparse Texture Synthesis

Guillaume Tartavel[1], Yann Gousseau[1], and Gabriel Peyré[2]

[1] LTCI, CNRS / Telecom ParisTech
{tartavel,gousseau}@telecom-paristech.fr
[2] Ceremade, Université Paris-Dauphine
gabriel.peyre@ceremade.dauphine.fr

Abstract. This paper presents a novel texture synthesis algorithm that performs a sparse expansion of the patches of the image in a dictionary learned from an input exemplar. The synthesized texture is computed through the minimization of a non-convex energy that takes into account several constraints. Our first contribution is the computation of a sparse expansion of the patches imposing that the dictionary atoms are used in the same proportions as in the exemplar. This is crucial to enable a fair representation of the features of the input image during the synthesis process. Our second contribution is the use of additional penalty terms in the variational formulation to maintain the histogram and the low frequency content of the input. Lastly we introduce a non-linear reconstruction process that stitches together patches without introducing blur. Numerical results illustrate the importance of each of these contributions to achieve state of the art texture synthesis.

Keywords: texture synthesis, sparse decomposition, dictionary learning, variational methods.

1 Introduction

Texture synthesis aims at generating an image that is visually similar to a given input exemplar but at the same time exhibits a strong randomness. Classical methods learn a global statistical model from the exemplar, and then sample a realization from this distribution. Simplest models consider independent stationary coefficients over a Fourier [6] or a wavelet basis [7,3]. More realistic syntheses are achieved by using an adapted representation learned from the exemplar [16] or by using higher order models taking into account dependencies among the coefficients [12].

Another class of methods are based on the Markov Random Field (MRF) assumption that each pixel of the texture depends only on its neighborhood. [2] introduced a parametric MRF model to textures. [4] and [15] propose a non-parametric MRF model where the probability law of a pixel given its neighbors is sampled directly from an exemplar of the texture to be synthesized. These approaches have been improved by several author, see for instance the recent review [14].

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 186–197, 2013.

These patch-based synthesis methods share similarities with recent sparsity-based methods developed for image restoration. These methods build a dictionary to perform a sparse expansion of the patches of the image in order to achieve state of the art denoising results, see for instance the work of Elad and Aharon [5]. Peyré shows in [11] that dictionary learning can be used for texture synthesis, the dictionary encoding in a compact manner the geometric features of the input image.

Our method builds upon the sparse texture synthesis method of Peyré [11], but extends it significantly to achieve state of the art results in terms of visual quality. We integrate several constraints to enrich the model and propose a variational energy that is minimized during the synthesis.

2 Dictionary Learning

The first step of our method trains a dictionary to approximate patches from the input exemplar. We take here the opportunity to introduce our notations while recalling the process of dictionary learning.

Matrix Notation. We denote a_i and a^j the rows and columns of a matrix $A = (a_i^j)_{i,j}$. The transposed matrix is denoted by A^*. Its ℓ^2 (Frobenius) norm is defined by $\|A\|^2 = \mathrm{tr}(A^*A) = \sum_{i,j} |a_i^j|^2$. The indicator function $\iota_{\mathcal{C}}$ of a set $\mathcal{C}$ is by definition equal to $+\infty$ outside $\mathcal{C}$ and equal to 0 inside $\mathcal{C}$. The ℓ^0 pseudo-norm of a vector w counts its non-zeros coordinates, $\|w\|_0 = \#\{i \setminus w_i \neq 0\}$.

Patches and Dictionary. We process and synthesize an image u by manipulating its patches. Given a set $(x_k)_{k=1}^K$ of K pixel locations, the patch extractor is $\Pi(u) = (p_k)_k \in \mathbb{R}^{L\times K}$ where for $t \in \{0,\dots,\tau-1\}^2$, $p_k(t) = u(x_k + t)$ defines the patch $p_k \in \mathbb{R}^L$ of $\tau \times \tau$ pixels, so that $L = d\tau^2$ where $d = 1$ for grayscale images and $d = 3$ for color images. We constrain the sampling locations on a regular grid $x_k = k\Delta$ for $k \in \mathbb{Z}^2$ where the spacing $\Delta > 0$ controls the amount of sub-sampling.

A dictionary $D = (d_n)_{n=1}^N \in \mathbb{R}^{L\times N}$ is used to approximate the patches $P = \Pi(u)$ as $P \approx DW$, where $W \in \mathbb{R}^{N\times K}$ are the coefficients of the approximation. Note that this corresponds to approximating independently each patch as $p_k \approx Dw_k$ within the dictionary. The quality of the approximation is measured using the ℓ^2 norm, $\|P - DW\|^2 = \sum_{k=1}^K \|p_k - Dw_k\|_2^2$.

Learning Stage. Given an exemplar u_0 of a texture we want to synthesize, an adapted dictionary $D_0 \in \mathbb{R}^{L\times N}$ is learned to provide an optimal sparse approximation of the patches $P_0 = \Pi(u_0) \in \mathbb{R}^{L\times K}$. Similarly to most dictionary learning methods, such as [5], we solve a non-convex optimization problem over the coefficients $W_0 \in \mathbb{R}^{N\times K}$ and the dictionary D_0

$$(W_0, D_0) \in \operatorname*{argmin}_{W,D} \|P_0 - DW\|^2 + \iota_{\mathcal{C}_{\mathrm{cols}}}(W) + \iota_{\mathcal{C}_{\mathrm{dict}}}(D) \tag{1}$$

where we enforce the coefficients to be S-sparse using

$$\mathcal{C}_{\text{cols}} = \left\{ W \in \mathbb{R}^{N \times K} \setminus \|w_k\|_0 \leqslant S \quad \forall k \right\} \tag{2}$$

and where the atoms of the dictionary are constrained to be normalized using $\mathcal{C}_{\text{dict}} = \left\{ D \in \mathbb{R}^{L \times N} \setminus \|d_n\| \leqslant 1 \quad \forall n \right\}$.

Several algorithms have been proposed to minimize approximately a non-convex energy of the form (1), see for instance the K-SVD method of [5].

3 Variational Formulation of the Synthesis Process

Once the dictionary D_0 has been learned from an input exemplar u_0, a texture u (and the associated coefficients W of $\Pi(u)$) is synthesized by minimizing a non-convex energy $E(u, W)$ equal to

$$\frac{1}{Z}\|\Pi(u) - D_0 W\|^2 + \iota_{\mathcal{C}_{\text{cols}}}(W) + \iota_{\mathcal{C}_{\text{rows}}}(W) + \alpha \mathcal{W}_2^2(\mu_u, \mu_{u_0}) + \beta \|h \star (u - u_0)\|^2. \tag{3}$$

Here $Z = \lceil \frac{\tau}{\Delta} \rceil^2$ is constant so that the ℓ^2 data fidelity is normalized with respect to the number of extracted patches. The two parameters $\alpha, \beta > 0$ are weighting the influence of their respective terms. The synthesized images are stationary points of E that are sampled at random with an iterative scheme, which is described in Sect. 4. We now give the precise definition and the rationale for each term of this energy.

Sparse Coding Constraint. The sparse coding energy $\frac{1}{Z}\|\Pi(u) - D_0 W\|^2 + \iota_{\mathcal{C}_{\text{cols}}}(W)$ is the same as the one used for the dictionary learning minimization (1). It requires that all the patches of u are well approximated by an S-sparse expansion in D_0.

Frequency Constraint. The constraint $\mathcal{C}_{\text{rows}}$ imposes that all the geometrical features of u_0 encoded in the dictionary are represented with the same respective proportions in u and u_0. It enforces that atoms of D_0 be used with the same frequencies of occurrence for the sparse expansion of both $\Pi(u_0)$ and $\Pi(u)$. It is defined as

$$\mathcal{C}_{\text{rows}} = \left\{ W \in \mathbb{R}^{N \times K} \setminus \forall n,\ \|w^n\|_0 \leqslant F_0^n \right\}.$$

The frequencies F_0^n are estimated from the input exemplar coefficients W_0 as

$$F_0^n = \frac{K}{K_0}\|w_0^n\|_0, \tag{4}$$

where K and K_0 are the number of patches extracted from u and u_0 respectively.

Histogram Constraint. Maintaining the gray-level or color histogram of a texture is perceptually important for texture synthesis. This is achieved by penalizing the deviation between the empirical gray-level or color distributions μ_u and μ_{u_0} of u and u_0.

An efficient and robust distance between distributions is the optimal transport distance, also known as the Wasserstein distance (see e.g. [13]). When u and u_0 have the same number of pixels, the L^2 Wasserstein distance is defined as

$$\mathcal{W}_2^2(\mu_u, \mu_{u_0}) = \min_{\sigma} \|u - u_0 \circ \sigma\|^2. \tag{5}$$

where σ runs over all the permutations of the pixels. This definition can be extended for images having a different number of pixels. For grayscale images, the optimal permutation is computed by simply sorting the pixel values. For color images, the Wasserstein distance is more involved to compute and to minimize. We approximate it as the sum of the grayscale distances along the three channels in a principal component orthogonal basis.

Low-Pass Constraint. Low frequency patterns, whose sizes exceed τ, are not controlled by the patch decomposition. To avoid the apparition of artifacts, we penalize the deviation of the low frequencies of u with respect to those of u_0 using the term $\|h \star (u - u_0)\|^2$, where $\star$ is the discrete convolution. We use a box filtering kernel $h = (\tau^{-2})_{1 \leqslant i,j \leqslant \tau}$ which performs an averaging over the spatial extension of a patch.

4 Synthesis Algorithm

The synthesis is obtained by randomly sampling the stationary points of $E(u, W)$ by a block-coordinate descent method that minimize E iteratively with respect to u and W. Pseudo-code 1 details the different steps of the method that are detailed in the remaining part of this section.

Algorithm 1: texture synthesis algorithm by minimization of (3).

Data: input texture u_0.
Input: parameters $\tau, \Delta, S, \alpha, \beta, N$.
Output: synthesized texture u.

1. **Dictionary learning:** compute (D_0, W_0) by minimizing (1).
2. **Frequency estimation:** compute $(F_0^n)_n$ using (4).
3. **Initialization:** set u to be a random white noise image.
4. **Block-coordinate minimization:** repeat until convergence
   - *image update:* $u \approx \underset{u}{\operatorname{argmin}}\, E(u, W)$, see Sect. 4.1.
   - *coefficient update:* $W \approx \underset{W}{\operatorname{argmin}}\, E(u, W)$, see Sect. 4.2.

4.1 Step 1: Minimization with Respect to u

Given a fixed set of coefficients W, we compute the minimization of $E(u, W)$ with respect to u alone

$$\min_u \tilde{E}_W(u) = \frac{1}{Z}\|\Pi(u) - P\|^2 + \alpha \mathcal{W}_2^2(\mu_u, \mu_{u_0}) + \beta \|h \star (u - u_0)\|^2 \qquad (6)$$

where $P = D_0 W$ is fixed.

Gradient Descent. The function $\tilde{E}_W$ is smooth almost everywhere since $\mathcal{W}_2^2$ is defined in (5) as the minimum among a set of paraboloids. It has a Lipschitz gradient. We thus use a gradient descent scheme to solve approximately (6)

$$u^{(\ell+1)} = u^{(\ell)} - \eta \nabla \tilde{E}_W(u^{(\ell)})$$

where $u^{(0)}$ is initialized from the previous iteration of the synthesis process.

The gradient of $\tilde{E}_W$ reads

$$\nabla \tilde{E}_W(u) = 2\mathcal{R}(u, P) + \alpha \nabla_u \mathcal{W}_2^2(\mu_u, \mu_{u_0}) + 2\beta \bar{h} \star h \star (u - u_0) \qquad (7)$$

$$\text{where} \quad \mathcal{R}(u, P) = \frac{1}{Z}\Pi^* \left(\Pi(u) - P\right)$$

and where $\bar{h}(x) = h(-x)$. The step sizes η must be smaller than twice the inverse of the Lipshitz constant of this gradient, $0 < \eta < 4 \times (1 + \alpha + \beta)^{-1}$.

Gradient of the Wasserstein Distance. When u and u_0 are grayscale images with the same number of pixels, the gradient of $u \mapsto \mathcal{W}_2^2(\mu_u, \mu_{u_0})$ reads

$$\nabla_u \mathcal{W}_2^2(\mu_u, \mu_{u_0}) = 2\left(u - u_0 \circ \sigma_{u_0} \circ \sigma_u^{-1}\right)$$

where σ_v is a permutation that order the pixel values $(v_i)_i$ of an image v,

$$v_{\sigma_v(1)} \leqslant \ldots \leqslant v_{\sigma_v(i)} \leqslant v_{\sigma_v(i+1)} \leqslant \ldots$$

The permutation σ_u is not unique when $u \mapsto \mathcal{W}_2^2(\mu_u, \mu_{u_0})$ is not differentiable. However, a descent direction is obtained by considering any valid ordering. When u and u_0 are color images, $\nabla_u \mathcal{W}_2^2(\mu_u, \mu_{u_0})$ is computed as the sum of the gradients over the three channels of the principal components of the distribution of the pixels of u_0.

Non-linear Improved Reconstruction. We note that $\frac{1}{Z}\Pi^*\Pi = \mathrm{diag}_i(\rho_i/Z)$ where $\rho_i \leqslant Z$ is the number of patches that overlap at a pixel location i. In the case of a perfect tiling, $\rho_i = Z$ is constant and $\frac{1}{Z}\Pi^*\Pi = \mathrm{Id}$: we can thus write

$$\mathrm{diag}_i(Z/\rho_i)\mathcal{R}(u, P) = u - \Pi^+ P$$

where $\Pi^+ = (\Pi^*\Pi)^{-1}\Pi^* = \mathrm{diag}_i(1/\rho_i)\Pi^*$ is the pseudo-inverse of Π. The term $\mathcal{R}(u, P)$ thus involves images that are reconstructed linearly by an averaging of patches. This step thus typically induces blur in the image u recovered at convergence. We improve this reconstruction by replacing the linear pseudo-inverse Π^+ by a Non-Linear (NL) reconstruction operator Π^+_{NL}, and replace, in the gradient expression (7), $\mathcal{R}(u, P)$ by $\mathcal{R}_{\mathrm{NL}}(u, P) = u - \Pi^+_{\mathrm{NL}}(P)$.

Graph-Cuts Reconstruction. As a particular example of non-linear, edge-preserving, reconstruction operator $\Pi^+_{\mathrm{NL}}(P)$, we use the graph-cut reconstruction introduced in [9] for texture synthesis. The idea is to sequentially blend each pair of adjacent patches along a cut. The patches are juxtaposed instead of being averaged. For a given patches collection, the resulting image is much sharper than the image obtained by linear reconstruction $\Pi^+(P)$.

A vertical cut γ between two consecutive patches (p_1, p_2) in P is a vertical path splitting the overlapping pixels into 2 groups. It is thus a subset of edges joining pairs of pixels (x_1, x_2). An optimal cut is computed by minimizing a functional measuring how well the two patches can be juxtaposed seamlessly along γ

$$J(\gamma, p_1, p_2) = \sum_{(x_1,x_2)\in\gamma} \frac{\|p_1(x_1) - p_2(x_1)\|^2 + \|p_1(x_2) - p_2(x_2)\|^2}{\|p_1(x_1) - p_1(x_2)\|^2 + \|p_2(x_1) - p_2(x_2)\|^2}. \tag{8}$$

The minimization of $J(\gamma, p_1, p_2)$ with respect to γ is done by linear programming. The full image reconstruction $\Pi^+_{\mathrm{NL}}(P)$ is performed in a greedy manner. Patches are first merged using vertical cuts resulting in complete rows. These rows are then merged together using large horizontal cuts.

Note that the resulting term $\mathcal{R}_{\mathrm{NL}}(u, P) = u - \Pi^+_{\mathrm{NL}}(P)$ does not correspond anymore to the true L^2 gradient. The non-linear behavior of the graph cut operator makes it difficult to analyze the convergence of the resulting process. Numerical simulations indicate that the process converges in practice, and that no blur is created by these iterations. An interesting question for future work is to understand whether the modification of the descent scheme can be re-casted as a minimization of some edge-preserving energy.

4.2 Step 2: Minimization with Respect to W

The minimization of E with respect to W when u is fixed corresponds to the following combinatorial optimization problem

$$\min_W \|P - D_0 W\|^2 + \iota_{\mathcal{C}_{\mathrm{cols}}}(W) + \iota_{\mathcal{C}_{\mathrm{rows}}}(W) \tag{9}$$

where $P = \Pi(u)$ is fixed. Even in the case where $\mathcal{C}_{\mathrm{rows}}$ is dropped (usual sparse coding), this problem is known to be NP-hard. We thus extend the Matching Pursuit (MP) greedy algorithm [10] to take into account the additional constraint $\mathcal{C}_{\mathrm{rows}}$ and compute an approximate solution of (9). Pseudo-code 2 describes the steps of this Constraint Matching Pursuit (CMP) algorithm, that are detailed in the remaining part of this section.

Index Selection Step. At step ℓ, the algorithm greedily updates the coefficients $W^{(\ell)}$ to reduce as much as possible the amplitude of the residual $R^{(\ell)} = P - D_0 W^{(\ell)}$ while staying within the constraint sets $\mathcal{C}_{\mathrm{rows}}$ and $\mathcal{C}_{\mathrm{cols}}$. This update only increases by at most one the number of non-zero coefficients

$$\varepsilon^\star = \operatorname*{argmin}_{\|\varepsilon\|_0=1} \|P - D_0(W^{(\ell)} + \varepsilon)\|^2 + \iota_{\mathcal{C}_{\mathrm{cols}}}(W^{(\ell)} + \varepsilon) + \iota_{\mathcal{C}_{\mathrm{rows}}}(W^{(\ell)} + \varepsilon).$$

Algorithm 2: constrained matching pursuit to approximately solve (9).

Data: patches P, dictionary D_0.
Input: sparsity S, frequencies F_0.
Output: coefficients W.
for $\ell = 0$ **to** $SK - 1$ **do**
- select the indices $(k^\star, n^\star)$ by solving (10).
- update the coefficients to obtain $W^{(\ell+1)}$ using (11).
- update the residual $R^{(\ell+1)} = P - D_0 W^{(\ell+1)}$ using (12).

Similarly as in the case of the MP algorithm, the optimal 1-sparse vector $\varepsilon^\star$ indexes an atom $d_{n^\star}$ and a patch $r_{k^\star}^{(\ell)}$ of the residual $R^{(\ell)} = (r_k^{(\ell)})_k$. These indices can also be shown to maximize the correlations

$$(k^\star, n^\star) = \operatorname*{argmax}_{(k,n)\in\mathcal{I}_\ell} |\langle r_k^{(\ell)}, d_n \rangle| \tag{10}$$

where $\mathcal{I}_\ell$ is the set of indices that are still available at step ℓ

$$\mathcal{I}_\ell = \left\{ (k,n) \setminus \sum\nolimits_{n' \neq n} \|(w^{(\ell)})_k^{n'}\|_0 < S \text{ and } \sum\nolimits_{k' \neq k} \|(w^{(\ell)})_{k'}^{n}\|_0 < F_0^n \right\}$$

where $W^{(\ell)} = \left((w^{(\ell)})_k^n\right)_{k,n}$ are the coefficient at step ℓ.

Coefficient Update Step. The coefficients are then updated according the MP rule

$$(w^{(\ell+1)})_k^n = \begin{cases} (w^{(\ell)})_k^n + \langle r_k^{(\ell)}, d_n \rangle & \text{if} \quad (k,n) = (k^\star, n^\star), \\ (w^{(\ell)})_k^n & \text{otherwise;} \end{cases} \tag{11}$$

and the residual $R^{(\ell+1)} = P - D_0 W^{(\ell+1)}$ becomes

$$r_k^{(\ell+1)} = \begin{cases} r_k^{(\ell)} - \langle r_k^{(\ell)}, d_n \rangle \cdot d_n & \text{if} \quad k = k^\star, \\ r_k^{(\ell)} & \text{otherwise.} \end{cases} \tag{12}$$

Computational Complexity. Under the assumption that $S \leqslant L, N \leqslant K$, the number of operations of the CMP algorithm is $O(KN(L + \log K))$ when precomputing the inner products and using a heap max-search. The computation of all inner products $\langle p_k, d_n \rangle$ provides a rough lower bound KNL for both our algorithm and the original version of MP [10].

4.3 Multi-scale Synthesis

The energy $E(u, W)$ is highly non-convex and the optimization process is likely to fall in bad local minima. Following several works on texture synthesis such as [11], we use a multi-scale strategy, that is particularly efficient when synthesizing

images with features having various scales, such as a quasi-periodic tiling of small scale features.

We first proceed by filtering and down-sampling the input exemplar u_0 to produce a multi-scale hierarchy of J images $(u_j)_{j=0}^{J-1}$, where u_j corresponds to a sub-sampling by a factor 2^j. Keeping a fixed patch size but a varying resolution allows the method to capture details of varying sizes. A dictionary D_j is learned for each u_j following the method described in Sect. 2. The synthesis algorithm detailed in pseudo-code 1 is then applied for $j = J-1, \ldots, 1, 0$ with (u_j, D_j) in place of (u_0, D_0). Between two scales j and $j-1$, the current texture u output at scale j is up-sampled by a factor 2 using bi-cubic interpolation to serve as the initialization for the synthesis step at scale j.

5 Synthesis Experiments

In this section, we provide comparisons between the proposed method and 3 classical synthesis algorithms. We also illustrate the contribution of each term in the energy (3).

Choice of the Parameters. For all numerical experiments in this section, we use patches of width $\tau = 12$ and a spacing $\Delta = \tau/2$. The synthesis is performed through $J = 3$ scales. We choose $S = 4$ non-zero values per patch and $N = 384$ elements in the dictionary. The parameters of the energy (3) are chosen as $\alpha = \beta = 1$; we observed that changing these values within reasonable proportions has little visual influence on the results.

Comparison. Our results are compared with 3 other decomposition-based texture synthesis algorithms [7,12,11]. Peyré's approach [11] is, to the best of our knowledge, the only synthesis model using sparse dictionary decomposition; our work is based on this approach. The method from Portilla and Simoncelli [12] is a state of the art method for generic texture synthesis. Let us here emphasize that we are interested in algorithms that truly *generate* a new texture from an exemplar. Copy-paste methods such as the classical Efros-Leung algorithm [4] and numerous related approaches (see e.g. [14]) produce visually striking results on a larger class of textures than [12]. However they merely proceed, either explicitly or not, by stitching together pieces from the exemplar, as illustrated in Fig. 1. These approaches are therefore not included in the present comparison. The method from Heeger and Bergen [7] is included for methodological reasons. It relies on the prescription of both the marginals of wavelet coefficients and the gray level (or color) distribution of images. Therefore, it is closely related to our method, albeit working in a prescribed, non-adaptive dictionary.

In Figure 2 are displayed several successful synthesis examples on textures from the Brodatz database [1]. On these, the proposed method performs significantly better than the method from Heeger and Bergen [7], especially for structured textures. This is mostly due to the fact that learned dictionary are more efficient than wavelet dictionary at capturing edges, corners or other geometric characteristics of these textures. Second, results on these examples are

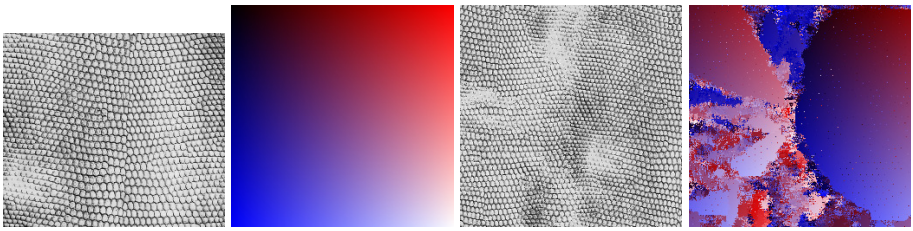


Fig. 1. A synthesis example using the method from [4]. From left to right: input, pixel coordinates visualized via a colormap, synthesis result, original position of the pixels used for the synthesis. Although pixels are synthesized one at a time, the texture is produced by stitching together pieces from the exemplar.

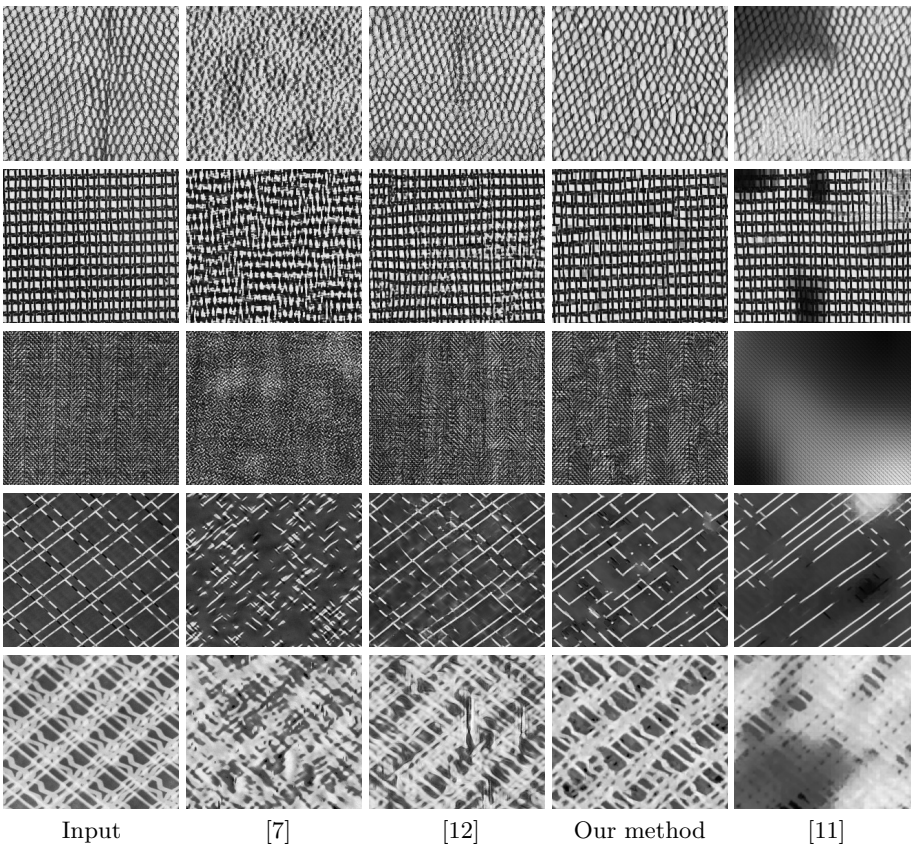


Fig. 2. From left to right: input texture, result using [7], result using [12], result from the proposed method, and result using the original framework [11]. The latter is often too smooth because of the multi-scale processing. All textures are from the Brodatz album [1].

comparable to those from [12]. Observe that this last method relies on second order statistics (correlations) between wavelet coefficients, while our approach only controls the proportion in which each dictionary atom is used. This indicates that the learned atoms could provide an interesting mathematical model of textons, as defined in [8]. Third, the importance of the penalty terms we introduced in energy (3) is evident through the comparison with the original method [11].

In Figure 3 are displayed two failure examples, a very large scale texture in the first row and a micro-texture in the second row. While the synthesis of very large scale textures without copy-pasting is still an open problem, micro-textures are successfully captured by relatively simple models such as the random phase model from [6]. A rough explanation of the inability of our method to synthesize such textures is that the sparse decomposition model is not adapted for noise-like patches.

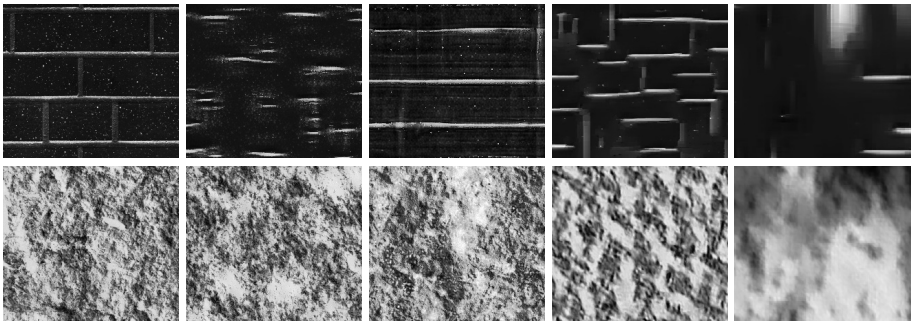


Fig. 3. Failure examples. Top: a large scale texture, bottom: a micro-texture for which the sparse hypothesis is not adapted.

Step-by-Step Analysis. In this second set of experiments, we illustrate the contributions of both the different components of energy (3) and the chosen minimization strategy. For each tested texture, we compare the following synthesis procedures:

- *basic*: only keep the first two terms (sparse coding constraint) and the fourth term (histogram constraint) of energy (3), which gives a method very similar to the initial framework of [11],
- *atom frequency*: add the atom frequency constraint $\mathcal{C}_{\text{rows}}$,
- *graph cut*: add the graph-cut reconstruction described in Sect. 4.1,
- *multi-scale*: add the multi-scale strategy described in Sect. 4.3,
- *low frequency*: add the low frequency constraint (last term of (3)), yielding the complete proposed procedure.

Several observations can be drawn from the results shown in Fig. 4. First, the atom frequency constraint is important for the generation of geometric structures and avoids an excessive use of smooth patches. Second, the non-linear image reconstruction procedure yields sharper results than the averaging of patches by the operator Π^*. Third, the multi-scale strategy introduces large scale coherence, without the computational cost of using larger patch sizes. Last, the low frequency constraint prevents from large scale variations due to the independence of patches.

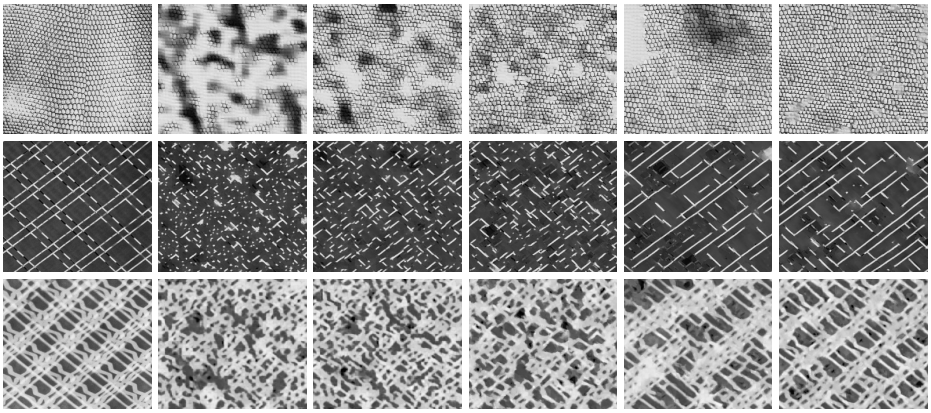


Fig. 4. Step-by-step examples. For each example, the result in the second column is obtained using a basic sparse synthesis scheme. Each column then shows the effect of adding a new constraint or of changing the minimization strategy. The last column is the complete proposed synthesis procedure.

6 Conclusions and Future Work

In this article we presented a variational approach to the texture synthesis problem. It extends significantly the initial sparsity-based framework of [11].

We identified a set of constraints to make the sparse approach suitable for texture synthesis. The first constraint controls the frequency of occurrence of each atom of the dictionary. The second constraint compensates the lack of coherence between adjacent patches. The third refinement is a cut-based reconstruction in the patch-based framework. The last and common refinement is the multi-scale processing.

The resulting model is well adapted to textures with sharp edges and small quasi-periodic patterns as shown in Fig. 2. It is less suitable for textures with high frequencies or structures at very large scale. Interesting perspectives include a better modeling of noisy textures, possibly through constraints on the power spectrum of images as in [6], as well as the use of a multi-scale learned dictionary.

Another perspective is to explore the variational approach which formulates the synthesis problem as a (highly non-convex) minimization problem. This paper uses a basic gradient descent but more efficient approaches may be used.

The solutions given by the minimization algorithm are (at most) local minima of the energy. Do they all look similar? If not, how to get a "good" solution?

Acknowledgement. This work has been partly supported by the ANR project MATAIM and by the ERC project SIGMA-Vision.

References

1. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. Dover, New York (1966)
2. Cross, G.R., Jain, A.K.: Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence (1), 25–39 (1983)
3. De Bonet, J.S.: Multiresolution sampling procedure for analysis and synthesis of texture images. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 361–368 (1997)
4. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1033–1038. IEEE (1999)
5. Elad, M., Aharon, M.: Image denoising via learned dictionaries and sparse representation. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 895–900. IEEE (2006)
6. Galerne, B., Gousseau, Y., Morel, J.-M.: Random phase textures: Theory and synthesis. IEEE Transactions on Image Processing 20(1), 257–267 (2011)
7. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: SIGGRAPH 1995, pp. 229–238 (1995)
8. Julesz, B.: A theory of preattentive texture discrimination based on first-order statistics of textons. Biological Cybernetics 41(2), 131–138 (1981)
9. Kwatra, V., Schodl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. ACM Transactions on Graphics 22(3), 277–286 (2003)
10. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing 41(12), 3397–3415 (1993)
11. Peyré, G.: Sparse modeling of textures. Journal of Mathematical Imaging and Vision 34(1), 17–31 (2009)
12. Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision 40(1), 49–70 (2000)
13. Villani, C.: Topics in optimal transportation, vol. 58. Amer Mathematical Society (2003)
14. Wei, L.-Y., Lefebvre, S., Kwatra, V., Turk, G.: State of the art in example-based texture synthesis. In: Eurographics 2009, State of the Art Report, EG-STAR. Eurographics Association (2009)
15. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: SIGGRAPH 2000, pp. 479–488. ACM Press/Addison-Wesley Publishing Co. (2000)
16. 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)

Outlier Removal Power of the L1-Norm Super-Resolution

Yann Traonmilin, Saïd Ladjal, and Andrés Almansa

Telecom Paristech LTCI
{yann.traonmilin,ladjal,andres.almansa}@telecom-paristech.fr

Abstract. Super-resolution combines several low resolution images having different sampling into a high resolution image. L1-norm data fit minimization has been proposed to solve this problem in a robust way. The outlier rejection capability of this methods has been shown experimentally for super-resolution. However, existing approaches add a regularization term to perform the minimization while it may not be necessary. In this paper, we recall the link between robustness to outliers and the sparse recovery framework. We use a slightly weaker Null Space Property to characterize this capability. Then, we apply these results to super resolution and show both theoretically and experimentally that we can quantify the robustness to outliers with respect to the number of images.

Keywords: super-resolution, interpolation, L1-norm.

1 Introduction

1.1 Problem Statement and State of the Art

The objective of super-resolution (SR) is to recover a high resolution (HR) image from several low resolution (LR) images. SR relies on the different sampling caused by motion between LR images acquisition. Several surveys of the subject exist in the literature [1–3]. The variational approach to super-resolution leads to the general form of a regularized minimization of the data-fit functional.

Most of the time, this data fit functional is an L^p-norm fit to the observed data. The L^2-norm (least squares data fit) has been the most frequent choice because of the optimality properties of the solution when data is contaminated by random noise [4]. Methods for least squares minimization such as the conjugate gradient are also well-known and efficient. More recently, L^1-norm minimization has been proposed to remove outliers from images [5] and as a robust way to perform super-resolution. It was shown that this method is robust to outliers in super-resolution [6–8]. Whatever norm is chosen, a regularization term is generally added to the variational problem.

Tychonov [4], bilateral total variation [6, 9], total variation [7, 8] or non-local regularization [10, 11] have been considered. In all these cases, an a priori hypothesis is made on the regularity of the HR image. However, when observation noise is random, it is likely that such regularization is not necessary when

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 198–209, 2013.

many LR images are available [12]. When there is an unnecessary regularization, high resolution features which could be recovered may be lost instead. In the case of unbounded outliers, results based on the least squares solution of super-resolution are not optimal because they are not well suited to the noise configuration.

In other areas of applied mathematics, it is known that the L^1-norm minimization has the ability to remove outliers. Candès and Tao showed in [13] that the outliers removal power of L^1-norm minimization is equivalent to a sparse recovery problem with sparsity having the cardinality of the support of outliers. They also showed that the observation matrix leads to the right result if it fulfills a restricted isometry property (RIP). Since this paper, the Null Space Property (NSP) has been shown to be an equivalent characterization of the capability to recover sparse vector from underdetermined observations [14].

1.2 Contributions

To our knowledge characterizations of L^1 norm minimization have not been used in the context of super-resolution. In **Section 2**, we set up the variational super-resolution problem. We then (**Section 3**) formulate the problem of forgiving outliers in the data in a slightly weaker way than in [13]. Vaswani [15] studied partially known support which is a stronger formulation of sparse recovery. Knowledge of the support is also used for structured sparsity where dedicated methods are designed [16]. [17] considered weaker formulation of the robustness of L^1-norm recovery by considering a fixed sparsity support. We consider arbitrary set of supports for outliers, which will allow an easy application to the super-resolution problem. This leads to an equivalent slightly weaker Null Space Property. In **Section 4**, we apply these results to the super-resolution interpolation problem. We find lower bounds on the number of images ensuring the robustness to a given number of outliers. We also show that allowing for arbitrary sets of supports for outliers can provide better practical results. Finally, we show experiments illustrating these results in **Section 5**.

2 Super-Resolution Interpolation Model

2.1 Low Resolution Image Generation

In a finite dimensional context, LR images are generated by a linear map A:

$$\begin{aligned} A : \mathbb{R}^{ML\times ML} &\to \left(\mathbb{R}^{L\times L}\right)^N \\ u &\to (A_i u)_{i=1,N} = (SQ_i u)_{i=1,N} \end{aligned} \tag{1}$$

where M is the super-resolution factor, N is the number of LR images, $L \times L$ is the size of LR images, u is a HR image of size $ML \times ML$, the A_i are linear maps generating LR images, S is the sub-sampling operator by a factor M and Q_i are the deformations associated with each LR image. SR is the process of recovering u_0 from $w = Au_0 + n$ (n is the observation noise). In this paper, we suppose that the Q_i are known. In this setting, the inversion of A is called super-resolution interpolation.

It has been shown in [12] that A is almost surely full rank when motions are random compositions of translations and rotations and $N \geq M^2$.

2.2 Variational Formulation

When A is full rank and $M^2 \leq N$, L^2-norm minimization guarantees that the energy of the reconstruction noise is bounded by the energy of observation noise times the operator norm of the pseudo-inverse $A^\dagger$ of A. This leads to useful results when observation noise is bounded. In the case of outliers, no assumption is made on the power of the noise and L^2 reconstruction does not guarantee a good result (unbounded reconstruction noise). In this paper, we study the efficiency of the L^1-norm minimization of the data-fit:

$$\text{argmin}_u \|Au - w\|_1 \tag{2}$$

with $w = Au_0 + n_0$. We look for conditions on A ensuring that u_0 is the unique solution of (2) when n_0 is an outlying noise. Outliers have the form : $n_0 = n.T$ with T a vector of 0 and 1 representing the support of the noise (the . represents the component-by-component vector product). We do not make any hypothesis on n. In Section 3, A will be a general full rank matrix of an over-determined system. In other sections, A will be an over-determined full rank SR operator of size $NL^2 \times (ML)^2$ with $N > M^2$.

3 Forgiving Matrices

3.1 Definitions

We introduce the concept of a $\mathcal{T}$-forgiving matrix A ($A : \mathbb{R}^m \to \mathbb{R}^p$) :

Definition 1. Forgiving Matrix *Let $\mathcal{T}$ be a set of supports in $\mathbb{R}^p$ (subset of $\{0,1\}^p$). A is called $\mathcal{T}$-forgiving if for all $T \in \mathcal{T}, n \in \mathbb{R}^p, u_0 \in \mathbb{R}^m$, we have:*

$$u_0 = \text{argmin}_u \|Au - (Au_0 + n.T)\|_1 \tag{3}$$

and u_0 is the unique minimizer.

When a matrix is $\mathcal{T}$-forgiving, the L^1 minimization recovers u_0 from any observation Au_0 contaminated by outliers whose support is in $\mathcal{T}$.

Definition 2. Sparse Capable Matrix *Let $\mathcal{T}$ be a set of supports in $\mathbb{R}^p$. B ($\mathbb{R}^p \to \mathbb{R}^q$) is called $\mathcal{T}$-sparse capable if for all $T \in \mathcal{T}, x_0 \in \mathbb{R}^p, y \in \mathbb{R}^q$, we have:*

$$x_0.T = \text{argmin}_x \|x\|_1 \text{ subject to } Bx = B(x_0.T) \tag{4}$$

and $x_0.T$ is the unique solution to problem (4).

The Null Space Property found in [14] only depends on the Null-Space of the matrix (and its interaction with supports). It is a non-concentration property which can be stated as follows:

Definition 3. Non-Concentration Property
Let $\mathcal{T}$ be a set of supports in $\mathbb{R}^p$ and V a subspace of $\mathbb{R}^p$. We say that V has the $\mathcal{T}$-Non-Concentration Property (NCP) if for all $v \in V \backslash \{0\}$ and all $T \in \mathcal{T}$

$$\|v.T\|_1 < \|v.T^c\|_1 \tag{5}$$

where T^c stands for the complement support of T.

We say that a matrix has the $\mathcal{T}$-Null Space Property ($\mathcal{T}$-NSP) if its null space has the $\mathcal{T}$-NCP.

Remark 1. Notice that, given the finite-dimensional setting, the NCP property implies the existence of a constant $\gamma < 1$ such that for all $v \in V$ and all $T \in \mathcal{T}$:

$$\|v.T\|_1 < \gamma \|v.T^c\|_1 \, . \tag{6}$$

This constant is called the NSP constant in the area of sparse recovery.

For the completeness of the paper, we now proceed with the direct proof of equivalence between the forgiveness of A and the Non-Concentration Property for the image of A (ImA). This equivalence can be obtained by combining [13] and [14] and slightly modifying the proofs to introduce arbitrary $\mathcal{T}$ instead of considering families of supports with fixed size. Indeed, [13] proves that forgiveness of a matrix (called linear coding capability) is equivalent to the sparse capability of any matrix whose kernel is the image of the original one and [14] proves that sparse capability is equivalent to the NCP (called there NSP).

3.2 Characterization of Forgiveness by the Non-Concentration Property

Theorem 1. *The two following propositions are equivalent:*

1. *A is $\mathcal{T}$-forgiving*
2. *ImA has the $\mathcal{T}$-Non Concentration Property.*

Proof. $1 \Rightarrow 2$: Let A be $\mathcal{T}$-forgiving, and $T \in \mathcal{T}$. Let $w \in \text{Im}A \backslash \{0\}$, there is u_0 such that $w = Au_0 \neq 0$. From the characterization of the L^1 minimizer in (3), we know that the following inequality holds

$$\|n.T\|_1 < \|Au - (w + n.T)\|_1 \tag{7}$$

for all $n \in \mathbb{R}^p$ and for every sub-optimal $u \neq u_0$. The strict inequality is a consequence of the uniqueness. In particular, for $n = w$ and $u = 2u_0$ ($u \neq u_0$ because $Au_0 \neq 0$), $Au = 2w$ and:

$$\|w.T\|_1 < \|w - w.T\|_1 = \|w.T^c\|_1 \, . \tag{8}$$

This shows that ImA satisfies the NCP on $\mathcal{T}$.

$2 \Rightarrow 1$: By hypothesis ImA has the NCP on $\mathcal{T}$. Let $u_0 \in \mathbb{R}^m$, $n \in \mathbb{R}^p$ and $T \in \mathcal{T}$. We have to show that u_0 is a minimizer of (3). Let $u \neq u_0$. The L^1-norm is the sum of L^1-norms taken on complementary supports:

$$\begin{aligned} f(u) &= \|Au - (Au_0 + n.T)\|_1 \\ &= \|(Au - (Au_0 + n.T)).T\|_1 + \|(Au - (Au_0 + n.T)).T^c\|_1 \\ &= \|A(u - u_0).T - n.T\|_1 + \|A(u - u_0).T^c\|_1 \ . \end{aligned} \tag{9}$$

We use the triangle inequality followed by the NCP :

$$\begin{aligned} &f(u) \geq \|n.T\|_1 - \|A(u - u_0).T\|_1 + \|A(u - u_0).T^c\|_1 \\ &f(u) > \|n.T\|_1 = f(u_0) \ . \end{aligned} \tag{10}$$

This strict inequality shows that u_0 is the unique minimizer of f. Consequently, A is $\mathcal{T}$-forgiving. □

With this slightly different result, the NCP can be checked on particular sets of supports, and not only those having a given cardinal as usually done in the sparse recovery framework. For example, in the context of image super-resolution, it is interesting to consider outliers contaminating a fixed number of LR images. This hypothesis models real situations like new object in the scene, light reflection...

Remark 2. The previous result implies the following already known result: the NSP of order K is equivalent to the K-sparse recovery capability. We just have to apply the result for $\mathcal{T}_K$ the set of all supports of cardinal K.

Remark 3. Note that in the context of outlier removal, the NCP could be called "Image Space Property" for A.

4 Application to Super-Resolution

4.1 Sufficient Condition for K-Forgiveness

In this section, we suppose that we only have the knowledge of the number of outliers K for the super-resolution problem. A is the super-resolution operator and $\mathcal{T}$ is the set of supports of cardinal K. We call this special case of $\mathcal{T}$-forgiveness the K-forgiveness. We first give sufficient conditions on the number of observed images for the NCP. Then we use the weaker Restricted Isometry Property (RIP) which is another sufficient condition for sparse capability. For any linear map A and support T, we call A_T, the operator $u \to (Au).T$.

Sufficient Condition for the NCP. Let T be a support with cardinal K. We look for a sufficient condition such that:

$$\frac{\|A_T u\|_1}{\|A_{T^c} u\|_1} < 1 \tag{11}$$

holds for all supports T of size K.

We start by bounding the L^1-operator norm of A_T. Let a_i be the lines of A:

$$\frac{\|A_T u\|_1}{\|u\|_1} = \frac{\sum_{i\in T} |< a_i, u >|}{\|u\|_1} \leq \frac{\sum_{i\in T}\sum_j |a_{i,j}u_j|}{\|u\|_1} \ . \tag{12}$$

Because each coefficient of A is a sample of a cardinal sine, we have $|a_{i,j}| \leq 1$. Therefore, we have

$$\begin{aligned} \frac{\|A_T u\|_1}{\|u\|_1} &\leq \frac{\sum_{i\in T}\sum_j |u_j|}{\|u\|_1} \\ &\leq K \ . \end{aligned} \tag{13}$$

Now we bound the ratio $\frac{\|A_T u\|_1}{\|A_{T^c} u\|_1}$. We use the L^1 conditioning $\kappa_{A_{T^c},1}$ of A_{T^c} . The L^p conditioning of an operator A is defined by:

$$\kappa_{A,p} = \frac{\sup_{\|u\|_p=1}\|Au\|_p}{\inf_{\|u\|_p=1}\|Au\|_p} \tag{14}$$

This leads to the following inequalities :

$$\begin{aligned} \frac{\|A_T u\|_1}{\|A_{T^c} u\|_1} &\leq \frac{K\|u\|_1}{\|A_{T^c} u\|_1} \\ &\leq K\left(\inf \frac{\|A_{T^c} u\|_1}{\|u\|_1}\right)^{-1} \\ &\leq K\frac{\kappa_{A_{T^c},1}}{\|A_{T^c}\|_1} \ . \end{aligned} \tag{15}$$

We use the fact that the L^1 operator norm $\|A_{T^c}\|_1$ can be bounded below the values taken on particular examples. The SR operator transforms constant HR images into constant LR images of same intensity. Consequently, $\|A_{T^c}\|_1 \geq (NL^2 - K)/(ML)^2$ and:

$$\frac{\|A_T u\|_1}{\|A_{T^c} u\|_1} \leq K(ML)^2 \frac{\kappa_{A_{T^c},1}}{NL^2 - K} \ . \tag{16}$$

We consider $\kappa^m_{A_{T^c},1}$ the maximum L^1 condition number of A restricted to the lines T^c. A condition for K-forgiveness is:

$$N > K\left(M^2\kappa^m_{A_{T^c},1} + 1\right) \ . \tag{17}$$

This inferior bound on N is linear with respect to K and is tight. Indeed, we can find a case where it is easy to see that N must be at least greater that a constant times K: Consider a 1D super-resolution problem with a sub-sampling factor of $M = 2$ and a number $N = 2P > 2$ observations with the corresponding translations being $0, 1, \ldots, 0, 1$ respectively (**i.e.** there are P observation with translation 0 and P with translation 1). In this case, the reconstruction according to equation (3) is the following HR signal: each sample is the median of the P values measured for each sample of the original signal. It is then clear that the L^1 variational setting can not resist to more than $P/2$ outliers. The worst case being that all outliers contaminate the same pixel (of the original signal) and have the same (unrelated to the signal) value.

Sufficient Condition for the RIP. A consequence of the equivalence between outlier resistance and sparse recovery is that we can use the Restricted Isometry Property [13] to find a sufficient condition for the K-forgiveness capability using the more convenient L^2 setting.

Definition 4. *B has the restricted isometry property of order J and constant $\delta \in]0,1[$ if for all $x \in \mathbb{R}^{N(L\times L)}$, for all supports T such that $|T| = J$*

$$(1-\delta)\|x.T\|_2 \leq \|B(x.T)\|_2 \leq (1+\delta)\|x.T\|_2 \,. \tag{18}$$

Given a matrix A, we set B as the orthogonal projection on $(ImA)^\perp$, that is $B = P_{(ImA)^\perp} = I - A(A^HA)^{-1}A^H$. Showing a RIP of order $J = K + K'$ with constant $\delta < \frac{\sqrt{K'}-\sqrt{K}}{\sqrt{K'}+\sqrt{K}}$ for B gives the K-sparse capability of B (See [18]). Consequently, $\ker B = \mathrm{Im}A$ has the NCP and A is K-forgiving. Moreover, if for all T of cardinal J:

$$\frac{\|A(A^HA)^{-1}A^H(x.T)\|_2}{\|x.T\|_2} \leq \sqrt{\delta} \tag{19}$$

then B has RIP of order J and constant δ (we square equation (18) and use the Pythagorean theorem). We can show using the same reasoning as in equation (13) that $\|A_T^H\|_2 = \|A_T\|_2 \leq \sqrt{J}$. Consequently, we bound the ratio:

$$\begin{aligned}\frac{\|A(A^HA)^{-1}A^H(x.T)\|_2}{\|x.T\|_2} \leq \sigma_{max}\frac{\|(A^HA)^{-1}A^H(x.T)\|_2}{\|x.T\|_2} &\leq \sigma_{max}\sigma_{min}^{-2}\|A_T\|_2 \\ &\leq \frac{\kappa_{A,2}^2\sqrt{J}}{\sigma_{max}}\end{aligned} \tag{20}$$

where $\sigma_?$ are the extremal singular values of A. Replacing with an admissible value of δ gives the condition:

$$\frac{\kappa_{A,2}^4(K+K')}{\sigma_{max}^2} \leq \frac{\sqrt{K'}-\sqrt{K}}{\sqrt{K}+\sqrt{K}} \,. \tag{21}$$

We take $K' = 3K$ (which we found is the optimal choice for the resulting constant) and get:

$$\frac{\kappa_{A,2}^4}{\sigma_{max}^2} \leq \frac{C_1}{\sqrt{K}} \tag{22}$$

where $C_1 = 0.0670$. $\sigma_{max} \geq \frac{\|Au\|_2}{\|u\|_2}$ because σ_{max} is the operator norm of A. Taking u as a constant image leads to: $\sigma_{max} \geq \sqrt{N/M^2}$. Finally,

$$N > M^2C_1^{-1}K\kappa_{A,2}^4 \tag{23}$$

is a sufficient condition for A to be K-forgiving. This bound uses the L^2 conditioning of the full operator. It has been shown [12, 19] that the conditioning

$\kappa_{A,2}$ converges to 1 for a large number of images and random motions. For a 1D signal and $M = 2$, this sufficient condition is roughly $N > 30K$ asymptotically. This bound has to be compared with the worst case scenario described in the precious section $N > 4K$ (which is a necessary condition).

4.2 Study of Particular Outlier Configurations

Here, the possibility to choose arbitrary sets of supports shows its benefit. Let $\mathcal{T}$ be the set of supports contaminating N_c LR images. In the same way as before, we want to find sufficient conditions for the NCP for $T \in \mathcal{T}$. More precisely, we allow for up to $K = N_c L^2$ outliers as long as they contaminate at most N_c images. We start by bounding operator norms with a tighter bound. Let $\mathcal{S}$ be the set of contaminated LR images indices ($|\mathcal{S}| = N_c$):

$$\begin{aligned} \frac{\|A_T u\|_1}{\|u\|_1} &\leq \sum_{i \in \mathcal{S}} \|A_i\|_1 \\ &\leq \sum_{i \in \mathcal{S}} \|SQ_i\|_1 \\ &\leq C_2 N_c \end{aligned} \tag{24}$$

where C_2 is an upper bound of $\|A_i\|_1$. C_2 is the maximum L^1-norm of the *sinc* used for interpolation. For 1D signals, the L^1 norm of the *sinc* is roughly bounded by the logarithm of the size of its support. We plot in Figure 1 a numerical evaluation of this constant for 2D SR. Figure 1 shows the max of the L^1 norms of the *sinc* for translational SR. This bound yields:

$$\begin{aligned} \frac{\|A_T u\|_1}{\|A_{T^c} u\|_1} &\leq \frac{\|A_T u\|_1 \|u\|_1}{\|u\|_1 \|A_{T^c} u\|_1} \\ &\leq \frac{\|u\|_1 C_2 N_c}{\|A_{T^c} u\|_1} \,. \end{aligned} \tag{25}$$

We introduce the pseudo-inverse $A^{\dagger}_{T^c} = (A^H_{T^c} A_{T^c})^{-1} A^H_{T^c}$ (recall that A_{T^c} has full column rank if $N - N_c < M^2$):

$$\begin{aligned} \sup_u \frac{\|u\|_1}{\|A_{T^c} u\|_1} &= \sup_{v \in \mathrm{Im} A_{T^c}} \frac{\|A^{\dagger}_{T^c} v\|_1}{\|v\|_1} \\ &\leq \|(A^H_{T^c} A_{T^c})^{-1}\|_1 \sup_{v \in \mathrm{Im} A_{T^c}} \frac{\|A^H_{T^c} v\|_1}{\|v\|_1} \\ &\leq \|(A^H_{T^c} A_{T^c})^{-1}\|_1 C_3 \end{aligned} \tag{26}$$

where C_3 is the maximum L^1 norm of the columns of $Q_i^H S^H$. This leads to the following sufficient condition :

Proposition 1. *If N_c images are contaminated, having N images with :*

$$N_c C_2 C_3 \| \left(A^H_{T^c} A_{T^c}\right)^{-1} \|_1 < 1 \tag{27}$$

guarantees a perfect reconstruction by L^1 minimization.

We evaluate in Figure 1, the constant C_2 and the product C_2C_3. We cannot bound $\|(A_{T^c}^H A_{T^c})^{-1}\|_1$ without knowledge of the motions because the its L^2 operator norm cannot be bounded (LR grids could be arbitrarily close). However, with random motions, we know that $(A_{T^c}^H A_{T^c})^1 \sim \frac{1}{N} I$ (see [12]) when T is fixed (on the first images for example) and $N \to \infty$. Asymptotically, the constraint is $N_c < C_4 N$ (for $L = 200$, $C_4 = 60$). This is much better than the previous result without hypothesis on the support, were the equivalent constant would have been $L^2 C_1^{-1} = 597000$ for $L = 200$. To have an idea of how robust the L^1 SR problem is, we can compare this result asymptotically with the case of random matrices [20] which have been studied in the context of sparse recovery. The equivalent condition would be: for outliers with sparsity K, with $NL^2 - M^2L^2$ observations and a signal of size NL^2, the constraint would be $K < (N - M^2)L^2/log(N/(N - M^2))$. We see that asymptotically, this constraint is much better because $log(N/(N - M^2)) \to 0$ when N grows.

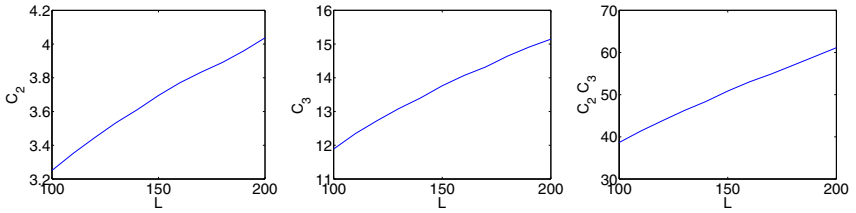


Fig. 1. Constants for translational SR (a) evaluation of C_2 with respect to L (b) evaluation of C_3 with respect to L (c) evaluation of C_2C_3 with respect to L

5 Experiments

5.1 Algorithm

The equivalence of the L^1 minimization with sparse recovery shown in Section 3 allows for the use of existing algorithms. Daubechies *et al.* [18] showed that iteratively reweighted least squares (IRWLS) convergence to the L^1 K-sparse solution is guaranteed when A is $K + \frac{2\gamma}{1-\gamma}$ sparse capable (with γ the NSP constant, see the remark in section 3.1) and when weights are carefully chosen (and the regularization of the weights $\epsilon_n \to 0$). We use this algorithm with the super-resolution L^2 data-fit functional. We construct iterations equivalent to [18]:

$$\begin{aligned} u_{n+1} &= \text{argmin}_u \|\Omega_n(Au - w)\|_2^2 \\ z_{n+1} &= Au_{n+1} - w \\ r_{n+1} &= \text{decreasing sort of abs}(z_{n+1}) \\ \epsilon_{n+1} &= \min(\epsilon_n, r_{n+1}(K+1)) \\ \Omega_{n+1} &= \text{diag}\left(\left[z_{n+1}^2 + \epsilon_{n+1}^2\right]^{-1/4}\right) \end{aligned} \tag{28}$$

We chose this algorithm because it converges quickly (a few iterations in practice) and convergence can be checked by looking at the variations of ϵ. Our aim is to give practical cases when outliers can be rejected.

5.2 Results

We show examples of outlier rejection using IRWLS. These practical results are better than our theoretical bounds which match the experience from compressed sensing. In Figure 2, we show an experimental evaluation of the number of images needed when N_c images are fully contaminated by outliers. For each N_c, N, the PSNR of the result of IRWLS is calculated for 30 experiments with different motion parameters. We plot the value of 10^{th} percentile (90% of the reconstructions have a better PSNR). Each line of this matrix can be interpreted as a phase transition diagram. In Figure 3, we contaminate one LR image with the absolute value of Gaussian random noise of variance 125 (pixels take values in

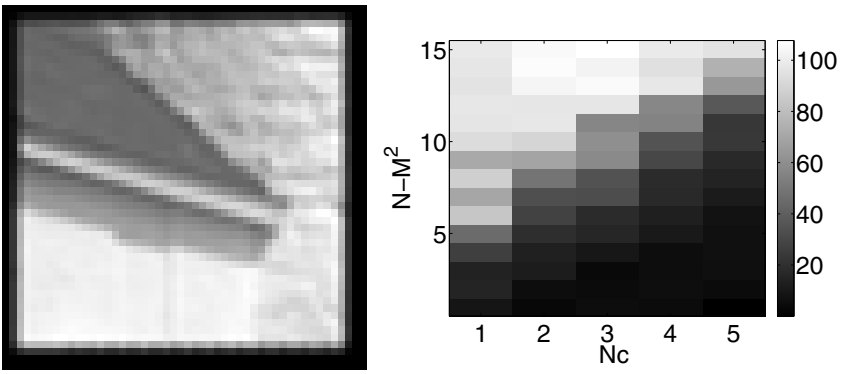


Fig. 2. Experimental outlier rejection (a) HR image used for all experiments (b) 10% percentile of the PSNR (in dB) with respect to the number of outliers N_c and number of images $N - M^2$

Fig. 3. L^1 SR interpolation outlier removal for $M = 2$ and $N = 7$ (a) Ideal HR image (b) Reconstructed image (c) LR images (outliers on the last image)

(a) (b) (c)

Fig. 4. L^1 SR interpolation outlier removal for $M = 2$ and $N = 8$ (a) Ideal HR image (b) Reconstructed image (c) LR images (outliers simulating saturated pixels (red squares) on the last 4 ones)

$[0, 255]$). In this case, 6 clean images give a perfect reconstruction of the HR image. In Figure 4, even with more contaminated images (4 noisy LR images on 8 LR images), if the location of outliers is different between LR images, L^1 minimization is still robust.

6 Conclusion

We have studied the outlier rejection capability of L^1 super-resolution in a quantitative way. The link between the outliers resistance problem and sparse recovery allows for the direct translation of the results of the literature of sparse recovery to over-determined super-resolution. We showed that if enough images are available, outlying noise can be completely removed from the observations. We gave theoretical bounds on the ratio between the number of images and outliers to ensure a perfect reconstruction without regularization. We showed that some conditions on the support of outliers allows for a robustness to more outliers. This result takes the form of much better theoretical bounds derived using these particular supports. Experiments show that fewer images are necessary to resist outliers in practice.

References

1. Farsiu, S., Robinson, D., Elad, M., Milanfar, P.: Advances and challenges in super-resolution. Int. J. Imaging Syst. Technol. 14(2), 47–57 (2004)
2. Milanfar, P.: Super-resolution imaging, vol. 1. CRC Press (2010)
3. Tian, J., Ma, K.K.: A survey on super-resolution imaging. Signal, Image and Video Processing 5(3), 329–342 (September 2011)
4. Hardie, R.C., Barnard, K.J., Armstrong, E.E.: Joint MAP registration and high-resolution image estimation using asequence of undersampled images. IEEE Transactions on Image Processing 6(12), 1621–1633 (1997)
5. Nikolova, M.: A Variational Approach to Remove Outliers and Impulse Noise 20(1-2), 99–120 (2004)

6. Farsiu, S., Robinson, M.D., Elad, M., Milanfar, P.: Fast and robust multiframe super resolution. IEEE Transactions on Image Processing 13(10), 1327–1344 (2004)
7. He, Y., Yap, K.H., Chen, L., Chau, L.P.: A Nonlinear Least Square Technique for Simultaneous Image Registration and Super-Resolution. IEEE Transactions on Image Processing 16(11), 2830–2841 (2007)
8. Yap, K.H., He, Y., Tian, Y., Chau, L.P.: A Nonlinear -Norm Approach for Joint Image Registration and Super-Resolution. IEEE Signal Processing Letters 16(11), 981–984 (2009)
9. Robinson, M.D., Toth, C.A., Lo, J.Y., Farsiu, S.: Efficient Fourier-Wavelet Super-Resolution. IEEE Transactions on Image Processing 19(10), 2669–2681 (2010)
10. Protter, M., Elad, M., Takeda, H., Milanfar, P.: Generalizing the nonlocal-means to super-resolution reconstruction. IEEE Transactions on Image Processing 18(1), 36–51 (2009)
11. 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)
12. Traonmilin, Y., Ladjal, S., Almansa, A.: On the amount of regularization for Super-Resolution interpolation. In: 20th European Signal Processing Conference (EUSIPCO 2012), Bucharest, Romania (August 2012)
13. Candes, E.J., Tao, T.: Decoding by linear programming. IEEE Transactions on Information Theory 51(12), 4203–4215 (2005)
14. Cohen, A., Dahmen, W., DeVore, R.: Compressed sensing and best k-term approximation. J. Amer. Math. Soc. 22(1) (2009)
15. Vaswani, N., Lu, W.: Modified-CS: Modifying Compressive Sensing for Problems With Partially Known Support. IEEE Transactions on Signal Processing 58(9), 4595–4607 (2010)
16. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Structured Sparsity through Convex Optimization. Statistical Science 27, 450–468 (2012)
17. Xu, W., Hassibi, B.: On sharp performance bounds for robust sparse signal recoveries. In: IEEE International Symposium on Information Theory, ISIT 2009, pp. 493–497. IEEE (June 2009)
18. Daubechies, I., DeVore, R., Fornasier, M., Güntürk, C.S.: Iteratively reweighted least squares minimization for sparse recovery. Comm. Pure Appl. Math. 63(1), 1–38 (2010)
19. Champagnat, F., Le Besnerais, G., Kulcsár, C.: Statistical performance modeling for superresolution: a discrete data-continuous reconstruction framework. J. Opt. Soc. Am. A 26(7), 1730–1746 (2009)
20. Dossal, C., Peyré, G., Fadili, J.: A numerical exploration of compressed sampling recovery. Linear Algebra and its Applications 432(7), 1663–1679 (2010)

Why Is the Census Transform Good for Robust Optic Flow Computation?

David Hafner, Oliver Demetz, and Joachim Weickert

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

Abstract. The census transform is becoming increasingly popular in the context of optic flow computation in image sequences. Since it is invariant under monotonically increasing grey value transformations, it forms the basis of an illumination-robust constancy assumption. However, its underlying mathematical concepts have not been studied so far. The goal of our paper is to provide this missing theoretical foundation. We study the continuous limit of the inherently discrete census transform and embed it into a variational setting. Our analysis shows two surprising results: The census-based technique enforces matchings of extrema, and it induces an anisotropy in the data term by acting along level lines. Last but not least, we establish links to the widely-used gradient constancy assumption and present experiments that confirm our findings.

Keywords: robust optic flow, census transform, illumination changes, anisotropy, variational method.

1 Introduction

In 1994, Zabih and Woodfill have proposed the so-called *census transform* [1]. It computes for every pixel a binary string (*census signature*) by comparing its grey value with the grey values in its neighbourhood. In particular, the signature encodes whether the neighbours are smaller than the reference pixel or not. For a 3×3 neighbourhood, the census signature has length 8 and can be represented efficiently via a single byte.

The census transform is becoming increasingly important: It provides an illumination-robust constancy assumption for solving correspondence problems in computer vision, e.g. computation of the displacement field (*optic flow*) in image sequences. The census signatures are by construction *morphologically invariant*, i.e. invariant under global monotonically increasing grey level rescalings. This can be an important advantage in modern applications such as driver assistant systems. Stein [2] uses the census signatures in an efficient feature matching approach. A hash table-based indexing scheme provides flow estimates in real-time and is well-suited for large displacements. Müller et al. [3] as well as Mohamed and Mertsching [4] exploit these sparse feature matches to handle large

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 210–221, 2013.

displacements and to recover image details lost in a coarse-to-fine minimisation technique, respectively. Furthermore, Müller et al. [5] embed the census transform as a data term into a variational optic flow framework. Tests in real-world scenarios show the desired morphological invariance of the resulting dense flow fields. Also in the context of stereo estimation, Ranftl et al. [6] have demonstrated the usefulness of the census transform under challenging lightning conditions.

In spite of its increasing popularity, however, the theoretical understanding of the successful census transform is still rather limited.

Our Contributions. The goal of our paper is to provide a thorough theoretical foundation of the census transform. Our contributions are threefold:

(i) We regard differences to neighbours as approximations of directional derivatives and study the continuous limit where all possible angles are taken into account.
(ii) We develop this concept into a constancy assumption, and we embed it as data term in a variational model for optic flow computation.
(iii) Most importantly, we analyse the energy functional and its minimisation in order to obtain a novel interpretation of census-based optic flow. We will see that this interpretation reveals many clever properties of the census transform which have not been used in other optic flow formulations.

We want to stress that the focus of our work is not on developing new competitive high-end optic flow methods: We are interested in a mathematical underpinning of census-based approaches. Once their properties are well-understood, these ideas can easily be embedded in any highly sophisticated optic flow method that ranks favourably in the Middlebury benchmark [7].

Related Work. Since 1994, the census idea has appeared under several names in the literature: Ojala et al. [8] developed almost the same concept independently but interpreted the resulting descriptor as a binary number (*local binary patterns*). Later, Calonder et al. [9] revisited this idea by introducing the feature point descriptor *BRIEF*.

There is also a long tradition of designing methods for illumination-robust optic flow computation. Inspired by Uras et al. [10], Brox et al. [11] achieve robustness w.r.t. additive brightness changes by considering the image gradients in addition to the intensity values. Chambolle and Pock [12] follow a different strategy to tackle these additive illumination changes and estimate the additive component explicitly. Another idea by Mileva et al. [13] is to make use of photometric invariants to design illumination-robust flow methods for colour images.

Paper Organisation. Starting with a continuous interpretation of the census transform, Section 2 presents our census-based variational optic flow method. The energy formulation and its minimisation yield new insights into census-based approaches. These results are presented in Section 3. After having sketched our numerical algorithm in Section 4, we evaluate the proposed method in Section 5. Finally, Section 6 concludes the paper with a summary and an outlook.

2 Census-Based Variational Optic Flow

In this section, we introduce our census-based optic flow method. To this end, we start with a formal definition of the original census transform and derive the corresponding constancy assumption in a continuous manner. This provides the basis of our energy functional and is the starting point of our analysis.

2.1 Census Transform

Let in a discrete setting $g_{i,j}$ denote the grey values of an image. Then, every digit of the census signature in pixel $(i,j)^\top$ is computed as

$$H\left(g_{i+d_1,j+d_2} - g_{i,j}\right), \tag{1}$$

where $(i+d_1, j+d_2)^\top$ is a neighbouring pixel, and $H\colon \mathbb{R} \to \{0,1\}$ denotes the Heaviside step function

$$H(z) := \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}. \tag{2}$$

2.2 Census-Based Constancy Assumption

Let us now transfer the census transform to the continuous setting and derive the associated constancy assumption. For this purpose, the three-dimensional function $f(x,y,t)$ represents a spatio-temporal image sequence, where $(x,y)^\top$ describes the location within the rectangular image domain $\Omega \subset \mathbb{R}^2$ and $t \in [0,T]$ denotes the time.

The argument of the step function in Equation (1) approximates a directional derivative. Consequently, one census digit can be interpreted as the discrete version of

$$H\left(\partial_{\boldsymbol{e}_\varphi} f(x,y,t)\right), \tag{3}$$

where the directional derivative operator $\partial_{\boldsymbol{e}_\varphi}$ only acts on the spatial domain. Here, the unit vector $\boldsymbol{e}_\varphi := (\cos\varphi, \sin\varphi)^\top$ specifies the direction.

We now derive the constancy assumption that corresponding points $(x,y,t)^\top$ and $(x+u, y+v, t+1)^\top$ in two consecutive frames have identical census signatures. In our notation, the functions $u, v\colon \Omega \to \mathbb{R}$ represent the sought optic flow. With the abbreviations $\boldsymbol{x} := (x,y,t)^\top$ and $\boldsymbol{w} := (u,v,1)^\top$, the constancy assumption of the census signature implies

$$H\left(\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x}+\boldsymbol{w})\right) - H\left(\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x})\right) \stackrel{!}{=} 0 \qquad \forall\, \varphi \in [0, 2\pi). \tag{4}$$

In order to embed this constraint as a data term in an energy functional, we consider a linearised version of it. To this end, we replace the Heaviside step function H by the smooth approximation

$$H_\varepsilon(z) := \frac{1}{2}\left(1 + \frac{z}{\sqrt{z^2+\varepsilon^2}}\right), \tag{5}$$

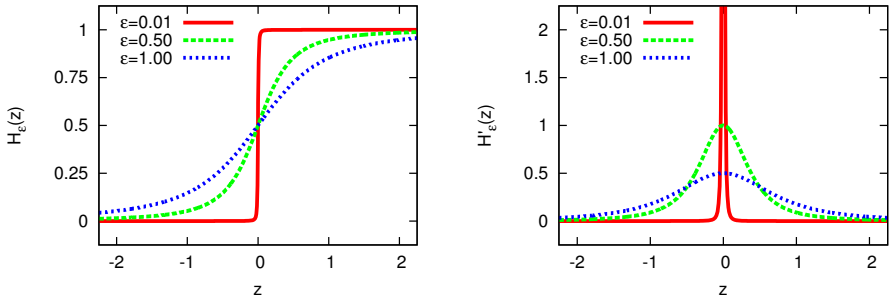


Fig. 1. Different approximations $H_\varepsilon(z)$ of the Heaviside step function (*left*) and corresponding derivatives $H'_\varepsilon(z)$ (*right*). Smaller choices of ε lead to closer approximations of the original sharp step function.

with a small positive regularisation parameter $\varepsilon > \varepsilon_0 > 0$ (cf. Figure 1). The numerical parameter ε_0 ensures that ε is also in the limit strictly larger than 0. Otherwise, the linearisation becomes invalid and the resulting data term would not be suitable for a typical variational optic flow framework [14].

Assuming small flow components u and v as well as a small change of the directional derivative $\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x})$ in time, we propose a twofold linearisation of the regularised version of constraint (4). For this purpose, let $\boldsymbol{\nabla}_3 := (\partial_x, \partial_y, \partial_t)^\top$ denote the spatio-temporal gradient and

$$H'_\varepsilon(z) = \frac{\varepsilon^2}{2\left(z^2 + \varepsilon^2\right)^{3/2}} \tag{6}$$

the derivative of $H_\varepsilon(z)$. At first, we linearise $\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x} + \boldsymbol{w})$ around $\boldsymbol{x}$ and obtain

$$H_\varepsilon\left(\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x}) + \boldsymbol{w}^\top \boldsymbol{\nabla}_3 \left(\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x})\right)\right) - H_\varepsilon\left(\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x})\right) \stackrel{!}{=} 0\,. \tag{7}$$

In the second step, the first H_ε term in (7) is linearised around $\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x})$:

$$H'_\varepsilon\left(\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x})\right) \cdot \boldsymbol{w}^\top \boldsymbol{\nabla}_3 \left(\partial_{\boldsymbol{e}_\varphi} f(\boldsymbol{x})\right) \stackrel{!}{=} 0\,. \tag{8}$$

2.3 Energy Formulation and Minimisation

Now, we embed the derived constancy assumption into a variational framework. To this end, let $\boldsymbol{\nabla} := \boldsymbol{\nabla}_2 := (\partial_x, \partial_y)^\top$ denote the spatial gradient operator. Furthermore, let $\alpha > 0$ be a regularisation parameter that allows to steer the impact of the data and smoothness term, respectively. Then, an energy incorporating the proposed linearised constancy assumption is given by

$$E(\boldsymbol{w}) := \int_\Omega \left(M(f, \boldsymbol{w}) + \alpha \cdot S(\boldsymbol{w})\right) \mathrm{d}\boldsymbol{x}\,, \tag{9}$$

with the census-based data term

$$M(f, \boldsymbol{w}) := \frac{1}{\pi} \int_0^{2\pi} {H'_\varepsilon}^2 \left(\partial_{\boldsymbol{e}_\varphi} f\right) \cdot \left(\boldsymbol{w}^\top \boldsymbol{\nabla}_3 \left(\partial_{\boldsymbol{e}_\varphi} f\right)\right)^2 \mathrm{d}\varphi \tag{10}$$

and the quadratic smoothness term

$$S(\boldsymbol{w}) := |\boldsymbol{\nabla} u|^2 + |\boldsymbol{\nabla} v|^2 \,. \tag{11}$$

For the sake of clarity, we omit the argument $\boldsymbol{x}$ of the functions f, u, and v. Following the calculus of variations, the minimiser of the energy in Equation (9) w.r.t. u and v has to fulfil the Euler-Lagrange equations

$$\frac{1}{\pi} \int_0^{2\pi} {H'_\varepsilon}^2 \left(\partial_{\boldsymbol{e}_\varphi} f\right) \cdot \partial_{\boldsymbol{e}_\varphi} f_x \cdot \boldsymbol{w}^\top \boldsymbol{\nabla}_3 \left(\partial_{\boldsymbol{e}_\varphi} f\right) \mathrm{d}\varphi \;-\; \alpha\, \Delta u = 0 \,, \tag{12}$$

$$\frac{1}{\pi} \int_0^{2\pi} {H'_\varepsilon}^2 \left(\partial_{\boldsymbol{e}_\varphi} f\right) \cdot \partial_{\boldsymbol{e}_\varphi} f_y \cdot \boldsymbol{w}^\top \boldsymbol{\nabla}_3 \left(\partial_{\boldsymbol{e}_\varphi} f\right) \mathrm{d}\varphi \;-\; \alpha\, \Delta v = 0 \,, \tag{13}$$

with reflecting Neumann boundary conditions $\boldsymbol{n}^\top \boldsymbol{\nabla} u = 0$ and $\boldsymbol{n}^\top \boldsymbol{\nabla} v = 0$. Here, $\boldsymbol{n}$ denotes the outer normal vector to the boundary of Ω.

3 Interpretation

To analyse the presented census-based data term in Equation (10), we exploit the symmetry of the integrand w.r.t. π and the equivalence $\partial_{\boldsymbol{e}_\varphi} f = \boldsymbol{e}_\varphi^\top \boldsymbol{\nabla} f$ for differentiable functions f:

$$M(f, \boldsymbol{w}) = \frac{2}{\pi} \int_0^{\pi} {H'_\varepsilon}^2 \left(\boldsymbol{e}_\varphi^\top \boldsymbol{\nabla} f\right) \cdot \left(\boldsymbol{w}^\top \boldsymbol{\nabla}_3 \left(\boldsymbol{e}_\varphi^\top \boldsymbol{\nabla} f\right)\right)^2 \mathrm{d}\varphi \,. \tag{14}$$

Further algebraic rearrangements allow to isolate the *census tensor* $\boldsymbol{C}$:

$$M(f, \boldsymbol{w}) = \frac{2}{\pi} \int_0^{\pi} {H'_\varepsilon}^2 \left(\boldsymbol{e}_\varphi^\top \boldsymbol{\nabla} f\right) \cdot \left(\boldsymbol{e}_\varphi^\top \begin{pmatrix} \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_x \\ \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_y \end{pmatrix}\right)^2 \mathrm{d}\varphi \tag{15}$$

$$= \begin{pmatrix} \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_x \\ \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_y \end{pmatrix}^\top \cdot \underbrace{\frac{2}{\pi} \int_0^{\pi} {H'_\varepsilon}^2 \left(\boldsymbol{e}_\varphi^\top \boldsymbol{\nabla} f\right) \boldsymbol{e}_\varphi \boldsymbol{e}_\varphi^\top \mathrm{d}\varphi}_{=:\boldsymbol{C}} \cdot \begin{pmatrix} \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_x \\ \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_y \end{pmatrix} . \tag{16}$$

A thorough analysis of this symmetric tensor $\boldsymbol{C} \in \mathbb{R}^{2\times 2}$ has already been performed by Weickert in the context of anisotropic diffusion filtering [15]. Here, we review the results that are relevant for us: Let $(r, \psi)^\top$ denote the polar coordinates of $\boldsymbol{\nabla} f \neq \boldsymbol{0}$. Then, Weickert has shown that the first and second eigenvector of $\boldsymbol{C}$ are parallel and perpendicular to isolines of f, respectively. They read

$$\boldsymbol{v}_\parallel(\psi) = \begin{pmatrix} -\sin\psi \\ \cos\psi \end{pmatrix} \quad \text{and} \quad \boldsymbol{v}_\perp(\psi) = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} , \tag{17}$$

and the corresponding eigenvalues are

$$\lambda_{\parallel}(r) = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} {H'_\varepsilon}^2 (r\cos\varphi) \cdot \sin^2\varphi \,\mathrm{d}\varphi\,, \tag{18}$$

$$\lambda_{\perp}(r) = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} {H'_\varepsilon}^2 (r\cos\varphi) \cdot \cos^2\varphi \,\mathrm{d}\varphi\,. \tag{19}$$

Let us now substitute the census tensor $\boldsymbol{C}$ in (16) by its eigendecomposition

$$\boldsymbol{C} = \lambda_{\parallel}(r) \cdot \boldsymbol{v}_{\parallel}(\psi)\, \boldsymbol{v}_{\parallel}^{\top}(\psi) + \lambda_{\perp}(r) \cdot \boldsymbol{v}_{\perp}(\psi)\, \boldsymbol{v}_{\perp}^{\top}(\psi)\ . \tag{20}$$

Thus, we obtain

$$M(f,\boldsymbol{w}) \;=\; \lambda_{\parallel}(r) \cdot \left(\boldsymbol{v}_{\parallel}^{\top}(\psi) \begin{pmatrix} \boldsymbol{w}^{\top}\boldsymbol{\nabla}_3 f_x \\ \boldsymbol{w}^{\top}\boldsymbol{\nabla}_3 f_y \end{pmatrix} \right)^2 \;+\; \lambda_{\perp}(r) \cdot \left(\boldsymbol{v}_{\perp}^{\top}(\psi) \begin{pmatrix} \boldsymbol{w}^{\top}\boldsymbol{\nabla}_3 f_x \\ \boldsymbol{w}^{\top}\boldsymbol{\nabla}_3 f_y \end{pmatrix} \right)^2, \tag{21}$$

where the original data term is explicitly split into two perpendicular constraints. In particular, this can be understood as a projection of the linearised gradient constancy assumption along and across isolines of f. Moreover, both terms are weighted with the corresponding eigenvalues $\lambda_{\parallel}(r)$ and $\lambda_{\perp}(r)$.

3.1 Anisotropic Data Term

Based on the formulation in Equation (21), the following two paragraphs discuss the behaviour of the data term at different image regions:

Vanishing Gradient. At extrema and homogeneous regions, where $|\boldsymbol{\nabla} f|$ vanishes ($r \to 0$), the eigenvalues of the census tensor $\boldsymbol{C}$ fulfil

$$\lim_{r\to 0} \lambda_{\parallel}(r) = \lim_{r\to 0} \frac{4}{\pi} \int_0^{\frac{\pi}{2}} {H'_\varepsilon}^2 (r\cos\varphi) \cdot \sin^2\varphi \,\mathrm{d}\varphi = {H'_\varepsilon}^2(0) \cdot \underbrace{\frac{4}{\pi} \int_0^{\frac{\pi}{2}} \sin^2\varphi \,\mathrm{d}\varphi}_{=1} \tag{22}$$

and accordingly

$$\lim_{r\to 0} \lambda_{\perp}(r) = \lim_{r\to 0} \frac{4}{\pi} \int_0^{\frac{\pi}{2}} {H'_\varepsilon}^2 (r\cos\varphi) \cdot \cos^2\varphi \,\mathrm{d}\varphi = {H'_\varepsilon}^2(0) \cdot \underbrace{\frac{4}{\pi} \int_0^{\frac{\pi}{2}} \cos^2\varphi \,\mathrm{d}\varphi}_{=1}\ . \tag{23}$$

Revisiting Equation (6), we see that ${H'_\varepsilon}^2(0) = \frac{1}{4\varepsilon^2}$. Hence, both eigenvalues $\lambda_{\parallel}$ and $\lambda_{\perp}$ exceed all bounds for close approximations of the Heaviside function. This means that the gradient constancy is assumed parallel as well as perpendicular to isolines of the image (cf. Equation (21)).

The occurring second order image derivatives $\partial_{\boldsymbol{e}_\varphi} f_x$ and $\partial_{\boldsymbol{e}_\varphi} f_y$ in the Euler-Lagrange Equations (12) and (13) behave differently in local extrema and homogeneous image regions. Consequently, our analysis of the constancy assumption has to differentiate these two cases:

Local Extrema. Here, the first order derivatives vanish, but the second order derivatives are in general non-zero. Since the reaction parts are weighted with the factor $\frac{1}{4\varepsilon^2}$, they dominate the diffusion terms entirely for small ε.

This reveals a surprising property of the discussed census-based model: The constancy assumption implicitly enforces a strong reliance on the local extrema, which contributes to the observed morphological invariance. On the one hand the positions of the minima and maxima remain constant under monotonically increasing grey level rescalings, and on the other hand the property $\boldsymbol{\nabla} f = \mathbf{0}$ at the extrema is not violated under those illumination changes. Thus, the imposed constancy assumption of the gradient holds here in all directions.

Homogeneous Regions. In contrast, the second order image derivatives $\boldsymbol{\nabla} f_x$ and $\boldsymbol{\nabla} f_y$ go to $\mathbf{0}$ in homogeneous regions. As a result, the terms

$$\partial_{\boldsymbol{e}_\varphi} f_x \;=\; \boldsymbol{e}_\varphi^\top \boldsymbol{\nabla} f_x \tag{24}$$

as well as

$$\partial_{\boldsymbol{e}_\varphi} f_y \;=\; \boldsymbol{e}_\varphi^\top \boldsymbol{\nabla} f_y \tag{25}$$

in the reaction parts of the Euler-Lagrange equations vanish. Hence, the solution at those regions is solely determined by filling-in the information from the neighbouring pixels:

$$\Delta u \;=\; 0\,, \tag{26}$$

$$\Delta v \;=\; 0\,. \tag{27}$$

High Contrast Edges. The previous paragraph was concerned with image regions where $r \to 0$. Let us now shed light on the opposite case ($r \to \infty$), which corresponds to high contrast edges of the image. Considering the eigenvalues of the census tensor $\boldsymbol{C}$ shows the strong anisotropic behaviour in those regions:

$$\lim_{r \to \infty} \frac{\lambda_\parallel(r)}{\lambda_\perp(r)} \;=\; \infty\,. \tag{28}$$

This ratio of the eigenvalues has already been analysed by Weickert for a family of monotonically decreasing functions including ${H'_\varepsilon}^2(z)$ [15].

Considering Equation (21), we see that the constancy of the gradient entries is here strongly imposed along isolines of f. In contrast, the constancy assumption across isolines is weighted down. This anisotropy is, besides the reliance on the local extrema, another reason for the morphological invariance of census-based methods. Under monotonically increasing grey level rescalings, the positions of the isophotes are invariant and additionally the directional derivatives along these isophotes remain zero. In other words, the gradient constancy assumption is valid in this direction.

3.2 Relation to the Gradient Constancy Assumption

Let us now illustrate the connection between the presented census-based constancy assumption and the widely-used gradient constancy assumption [10, 11]. The data term of the linearised gradient constancy assumption reads

$$\left(\boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_x\right)^2 + \left(\boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_y\right)^2 = \begin{pmatrix} \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_x \\ \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_y \end{pmatrix}^\top \boldsymbol{I} \begin{pmatrix} \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_x \\ \boldsymbol{w}^\top \boldsymbol{\nabla}_3 f_y \end{pmatrix}, \tag{29}$$

where $\boldsymbol{I}$ denotes the 2×2 identity matrix. This formulation inherently decouples the constancy assumptions of the gradient entries f_x and f_y. Comparing the data terms (16) and (29), we observe that the reason for the increased robustness of census-based methods (compared to gradient constancy) is hidden in the census tensor $\boldsymbol{C}$. This confirms our findings from Section 3.1: Coupling the constancy assumptions of f_x and f_y by $\boldsymbol{C}$, or rather by its eigenvectors $\boldsymbol{v}_\parallel(\psi)$ and $\boldsymbol{v}_\perp(\psi)$, induces an anisotropic behaviour which effects the proposed invariance.

Replacing the regularised step function H_ε in Equation (16) by the identity function, the matrix $\boldsymbol{C}$ comes down to

$$\frac{2}{\pi}\int_0^\pi 1 \cdot \boldsymbol{e}_\varphi \boldsymbol{e}_\varphi^\top \,\mathrm{d}\varphi = \frac{2}{\pi}\int_0^\pi \begin{pmatrix} \cos^2\varphi & \cos\varphi\sin\varphi \\ \sin\varphi\cos\varphi & \sin^2\varphi \end{pmatrix} \mathrm{d}\varphi = \frac{2}{\pi}\begin{pmatrix} \frac{\pi}{2} & 0 \\ 0 & \frac{\pi}{2} \end{pmatrix} = \boldsymbol{I}\,. \tag{30}$$

The resulting data term coincides with the gradient constancy assumption in Equation (29). Consequently, the census-based method may be regarded as a *censorisation* of the gradient constancy. On the one hand, this censorisation decreases the amount of extracted image information due to the binary quantisation of the directional derivative values. On the other hand, however, the induced anisotropy increases the robustness under illumination changes. While the original gradient constancy assumption is solely invariant w.r.t. global additive illumination changes, the *censored* gradient constancy assumption provides an invariance against any kind of monotonically increasing grey level rescalings.

4 Implementation

For the ease of implementation, we cast the linearised constancy assumption from (8) into the versatile motion tensor framework by Bruhn [16]. To this end, we exploit the equivalence

$$H'_\varepsilon\left(\partial_{\boldsymbol{e}_\varphi} f\right) \cdot \boldsymbol{w}^\top \boldsymbol{\nabla}_3 \left(\partial_{\boldsymbol{e}_\varphi} f\right) \;=\; \boldsymbol{w}^\top \boldsymbol{\nabla}_3 H_\varepsilon\left(\partial_{\boldsymbol{e}_\varphi} f\right)\,. \tag{31}$$

Furthermore, we approximate the periodic integral in Equation (10) by the Riemann sum and finally obtain

$$M\left(f, \boldsymbol{w}\right) \;=\; \boldsymbol{w}^\top \left(\frac{2}{N} \sum_{n=0}^{N-1} \boldsymbol{\nabla}_3 H_\varepsilon\left(\partial_{\boldsymbol{e}_{\varphi_n}} f\right) \cdot \boldsymbol{\nabla}_3^\top H_\varepsilon\left(\partial_{\boldsymbol{e}_{\varphi_n}} f\right) \right) \boldsymbol{w}\,, \tag{32}$$

where N denotes the number of considered neighbours and $\varphi_n := 2\pi\frac{n}{N}$. Choosing e.g. $N = 8$, the direct neighbours of each pixel are used to compute the census

signatures. Generally, we assume the images to be sampled on a regular grid with horizontal and vertical grid sizes h_1 and h_2, respectively. Accordingly, the directional derivative $\partial_{\boldsymbol{e}_{\varphi_n}} f$ at pixel $(i,j)^\top$ is approximated via the two point stencil

$$\left[\partial_{\boldsymbol{e}_{\varphi_n}} f\right]_{i,j} = \frac{[f]_{i+d_1,j+d_2} - [f]_{i,j}}{\sqrt{(h_1 d_1)^2 + (h_2 d_2)^2}}\,, \tag{33}$$

where the vector $\boldsymbol{d} := (d_1, d_2)^\top \neq \mathbf{0}$ represents, especially for diagonal neighbours, a scaled version of $\boldsymbol{e}_{\varphi_n}$ (cf. Section 2.1). All other spatial and temporal derivatives are computed by means of standard finite differences.

The resulting discrete versions of the Euler-Lagrange Equations (12) and (13) create a sparse linear system of equations, which we solve iteratively using a variant of the Gauß-Seidel method, namely successive over-relaxation [17].

5 Evaluation

Our experiments have been performed on the commonly available test image sequence *New Marble*[1]. We subjected the grey values $g \in [0, 255]$ of the second input image to the monotonically increasing transformation

$$g_{\text{out}} = 255 \cdot \left(\frac{m \cdot g_{\text{in}} + a}{255}\right)^{\gamma}\,, \tag{34}$$

where the constant a represents additive changes, $m > 0$ multiplicative changes and $\gamma > 0$ is used for gamma corrections.

The parameter ε of the regularised step function should be adapted to the noise level and is here fixed to 0.1. Furthermore, the input images are presmoothed with a Gaussian of standard deviation 0.8 and the census signatures are determined on a 3×3 neighbourhood ($N=8$).

Figure 2 demonstrates the increased robustness of the census-based method compared to the gradient constancy assumption. In the absence of artificial illumination changes (*first column*), the gradient constancy provides a better *average angular error* (AAE) [18]. It extracts more information form the input images. The resulting flow fields for additive changes (*second column*) are unaltered due to the inherent invariance of both methods. In contrast, the gradient constancy assumption is not invariant under multiplicative rescalings and gamma corrections (*third and fourth column*), while the censored version provides an increased robustness. The absolute invariance is lost due to the presmoothing and ε being unequal to zero.

In addition, the plots in Figure 3 confirm these observations. The gradient constancy is not able to compensate for the multiplicative changes and gamma corrections. Contrary, the census-based approach provides the proposed robustness. However, this increase of robustness is associated with a loss of accuracy in the presence of small illumination changes.

[1] Available from `http://i21www.ira.uka.de/image_sequences`

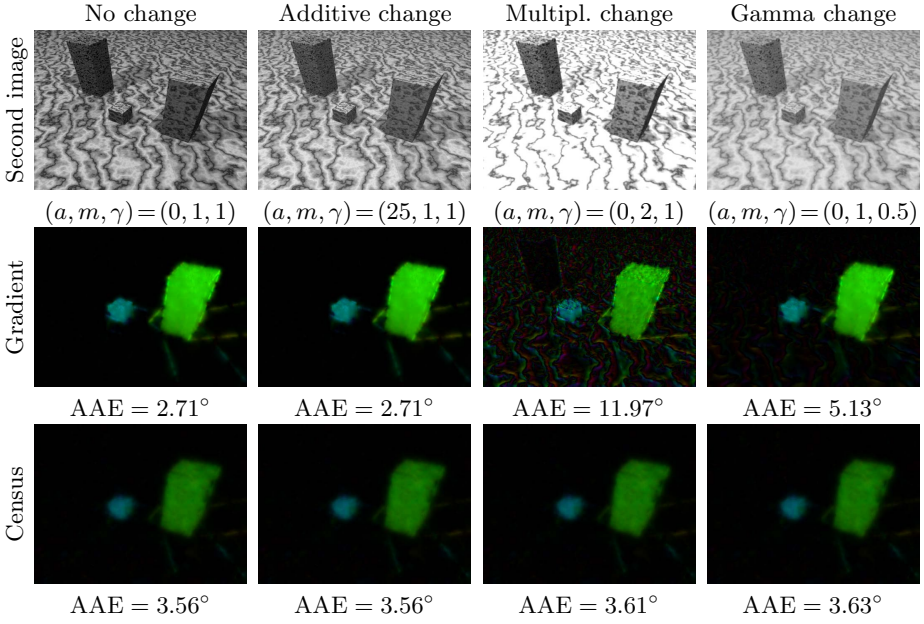


Fig. 2. Visual comparison of the gradient constancy assumption (*second row*, $\alpha = 430$) and its censored version (*third row*, $\alpha = 7$) under illumination changes. The second input image (*first row*) is manipulated by different grey level rescalings (cf. Equation (34)).

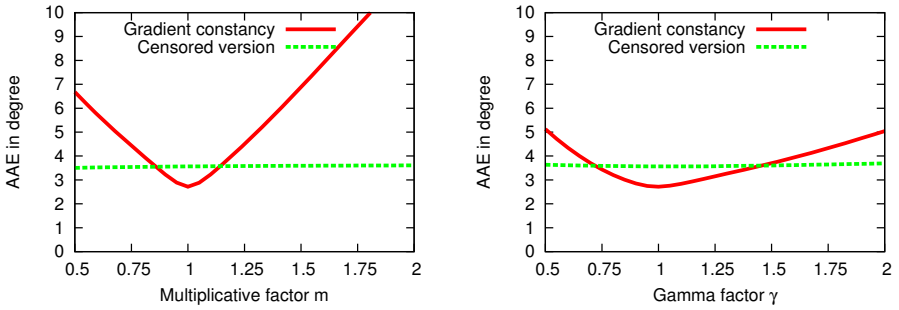


Fig. 3. Comparison of the gradient constancy assumption and its censored version under global multiplicative illumination changes (*left*) and gamma corrections (*right*). The parameter setting can be found in Figure 2.

6 Conclusions and Future Work

We have seen that interpreting the census transform in the continuous limit and embedding it into a variational framework reveals unexpected insights. The presented census-based technique shows two key properties: the strong reliance on local extrema as well as the restriction of the gradient constancy assumption

along level lines. These advanced features are efficiently realised by a very simple binary transform. They exploit the morphological invariance of the gradient direction in a clever way and yield the observed robustness under illumination changes. This builds the basis for the success of the census transform in the context of correspondence problems.

These promising insights motivate us to investigate also generalisations of the census transform that involve higher order constancy assumptions, e.g. constancy of the Hessian. The key properties of the census transform are of course not restricted to optic flow models. They have already proven to be equally beneficial for other computer vision tasks such as stereo reconstruction [6] or face detection [19].

Our findings confirm the general usefulness of studying continuous limits of inherently discrete morphological transforms. Other examples include e.g. continuous reinterpretations of median filters in terms of mean curvature motion [20] and morphological amoebae as self-snakes [21].

Acknowledgements. Our research has partly been funded by the Deutsche Forschungsgemeinschaft (DFG) through the Saarbrücken Graduate School of Computer Science and a Gottfried Wilhelm Leibniz Prize.

References

1. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994)
2. Stein, F.: Efficient computation of optical flow using the census transform. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 79–86. Springer, Heidelberg (2004)
3. Müller, T., Rannacher, J., Rabe, C., Franke, U.: Feature- and depth-supported modified total variation optical flow for 3D motion field estimation in real scenes. In: Proc. 24th IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, pp. 1193–1200. IEEE Computer Society Press (2011)
4. Mohamed, M.A., Mertsching, B.: TV-L1 optical flow estimation with image details recovering based on modified census transform. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Fowlkes, C., Wang, S., Choi, M.-H., Mantler, S., Schulze, J., Acevedo, D., Mueller, K., Papka, M. (eds.) ISVC 2012, Part I. LNCS, vol. 7431, pp. 482–491. Springer, Heidelberg (2012)
5. Müller, T., Rabe, C., Rannacher, J., Franke, U., Mester, R.: Illumination-robust dense optical flow using census signatures. In: Mester, R., Felsberg, M. (eds.) DAGM 2011. LNCS, vol. 6835, pp. 236–245. Springer, Heidelberg (2011)
6. Ranftl, R., Gehrig, S., Pock, T., Bischof, H.: Pushing the limits of stereo using variational stereo estimation. In: IEEE Intelligent Vehicles Symposium, Alcala de Henares, pp. 401–407. IEEE Computer Society Press (2012)
7. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision 92(1), 1–31 (2011)

8. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29(1), 51–59 (1996)
9. 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)
10. Uras, S., Girosi, F., Verri, A., Torre, V.: A computational approach to motion perception. Biological Cybernetics 60, 79–87 (1988)
11. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
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. Mileva, Y., Bruhn, A., Weickert, J.: Illumination-robust variational optical flow with photometric invariants. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 152–162. Springer, Heidelberg (2007)
14. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
15. 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)
16. Bruhn, A.: Variational Optic Flow Computation: Accurate Modelling and Efficient Numerics. PhD thesis, Dept. of Computer Science, Saarland University, Saarbrücken, Germany (2006)
17. Young, D.M.: Iterative Solution of Large Linear Systems. Academic Press, New York (1971)
18. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. International Journal of Computer Vision 12(1), 43–77 (1994)
19. Fröba, B., Ernst, A.: Face detection with the modified census transform. In: Proc. 6th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 91–96. IEEE Computer Society Press (2004)
20. Guichard, F., Morel, J.M.: Partial differential equations and image iterative filtering. In: Duff, I.S., Watson, G.A. (eds.) The State of the Art in Numerical Analysis. IMA Conference Series (New Series), vol. 63, pp. 525–562. Clarendon Press, Oxford (1997)
21. Welk, M., Breuß, M., Vogel, O.: Morphological amoebas are self-snakes. Journal of Mathematical Imaging and Vision 39(2), 87–99 (2011)

Generalised Perspective Shape from Shading in Spherical Coordinates

Silvano Galliani[1,2], Yong Chul Ju[1,3], Michael Breuß[1], and Andrés Bruhn[3]

[1] Institute for Applied Mathematics and Scientic Computing,
BTU Cottbus, 03046 Cottbus, Germany
{galliani,ju,breuss}@tu-cottbus.de
[2] Institute of Geodesy and Photogrammetry,
ETH Zürich, 8093 Zürich, Switzerland
silvano.galliani@geod.baug.ethz.ch
[3] Institute for Visualization and Interactive Systems,
University of Stuttgart, 70569 Stuttgart, Germany
bruhn@vis.uni-stuttgart.de

Abstract. In the last four decades there has been enormous progress in Shape from Shading (SfS) with respect to both modelling and numerics. In particular approaches based on advanced model assumptions such as perspective cameras and non-Lambertian surfaces have become very popular. However, regarding the positioning of the light source, almost all recent approaches still follow the simplest geometric configuration one can think of: The light source is assumed to be located exactly at the optical centre of the camera. In our paper, we refrain from this unrealistic and severe restriction. Instead we consider a much more general SfS scenario based on a perspective camera, where the light source can be positioned *anywhere* in the scene. To this end, we propose a novel SfS model that is based on a Hamilton-Jacobi equation (HJE) which in turn is formulated in terms of spherical coordinates. This particular choice of the modelling framework and the coordinate system comes along with two fundamental contributions: While on the modelling side, the spherical coordinate system allows us to derive a generalised brightness equation – a compact and elegant generalisation of the standard image irradiance equation to arbitrary configurations of the light source, on the numerical side, the formulation as Hamilton-Jacobi equation enables us to develop a specifically tailored variant of the fast marching (FM) method – one of the most efficient numerical solvers in the entire SfS literature. Results on synthetic and real-world data confirm our theoretical considerations. They clearly demonstrate the feasibility and efficiency of the generalised SfS approach.

Keywords: shape from shading, Hamilton-Jacobi equation, viscosity solution, fast marching, general light source configuration, spherical coordinates.

1 Introduction

Since the early works of Rindfleisch [1] and Horn [2] more than four decades ago, *Shape from Shading (SfS)* is considered one of the key problems in computer vision. Given the information about the reflectance of the surface and the position of the light

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 222–233, 2013.

source, its goal is to reconstruct the 3D depth of an object in a scene from a single 2D input image. In practice, SfS has a wide field of interesting applications. They range from classical large scale problems such as astronomy [1] or terrain reconstruction [3,4] to challenging small scale tasks such as dentistry [5] or endoscopy [6,7,8,9].

In order to make both the modelling and the computation tractable, early SfS approaches relied on strongly simplified assumptions. Since these approaches were first used for large scale problems such as astronomy, they were typically based on a camera model with orthographic projection, a light source that illuminates the scene from infinity, as well as a physically incorrect light transport that neglects attenuation [2]. Not surprisingly, these early methods worked reasonably well in the case of large scale problems – the scenario they were designed for. However, if applied to tasks, where the distance of the camera to the object is small, such methods revealed a consistently poor performance in terms of reconstruction quality [10,11]. Moreover, independent of the distance to the object, the corresponding mathematical models turned out to be severely ill-posed showing a strong dependency of the result on the initialisation [12]. Evidently, all these simplified assumptions did not make the SfS problem easier but actually rendered it more difficult from both a theoretical and a practical viewpoint.

This observation led to significant progress in the last few years. Nowadays, a perspective camera model [7,13,14] based on the inverse square law for light attenuation [12,15] has become *the* standard assumption of recent SfS methods including those techniques with more advanced, i.e. non-Lambertian, reflection models [16,17]. Moreover, with the consideration of the perspective camera model, the assumed position of the light source was moved from infinity to the optical centre of the camera [7]. While, this choice is very convenient from a computational viewpoint, it is obvious that a light source cannot be located at this place. This holds particularly for flash photography, where the position of the optical centre and the light source differ by construction.

In face of these considerations, it is surprising that there has hardly been any effort in the literature to model perspective SfS with an *arbitrary* position of the light source. In fact, there is only one work known to the authors, where a variational model for endoscopic SfS was proposed with a position of the light source different than the one in the optical centre [9]. This approach, however, suffers from two main drawbacks. On the one hand, the approach does not make use of proper discretisations of the hyperbolic terms such as e.g. upwind-type schemes [18]. This, however, would be necessary to ensure solutions in the viscosity sense [19]. On the other hand, the numerical algorithm proposed for this method is quite slow. In fact, the authors rely on a simple Jacobi-like scheme that is moreover explicit in the irradiance equation [20].

Contributions. In this paper we address all of the aforementioned problems. To this end, we formulate the perspective SfS problem in terms of a *Hamilton-Jacobi (HJE) equation* based on *spherical coordinates*. While such approaches have already been proposed for the standard case with the light source being located at the optical centre of the camera [15,21], we demonstrate that employing such a spherical coordinate system is perfectly suited for the general case where the position of the light source can be arbitrary. In this context, our contributions are twofold:

1. On the modelling side, we derive a novel mathematical model for SfS in spherical coordinates which we call the *generalised brightness equation*. While the model itself can handle the most general geometry with the light source not being located in the optical centre of the camera, its compact and elegant structure suggests that this approach is the natural and intuitive way to formulate this problem.
2. On the numerical side, we develop a *highly efficient numerical algorithm* for solving the resulting highly non-linear HJE. This specifically tailored algorithm not only extends the popular fast marching (FM) method [22] by an iterative correction step but also guarantees to find solutions in the viscosity sense at the same time.

Summarising, we propose a perspective SfS approach that combines the applicability of a general SfS method with the efficiency of FM based approaches.

Organisation. In Section 2 we start by discussing the general perspective SfS framework. In Section 3 we then describe the representation of a surface in both Cartesian and the spherical coordinates. This allows us to compute the corresponding surface normal in Section 4 and finally to derive a compact formulation of the brightness equation for the general case. in Section 5. After we discuss our extended variant of the FM scheme in to solve this equation efficiently in Section 6, we present the results of our method in Section 7. The paper is concluded by a summary in Section 8.

2 Perspective Shape from Shading

Let us consider the general setting for perspective SfS with focal length $\mathtt{f}$, where the position of the light source can be anywhere in the scene. Let us furthermore assume that the surface is Lambertian with uniform albedo and the light fall-off follows the inverse square law. Then, the brightness of the acquired image I is given by the so-called *brightness equation* [12]:

$$I(\mathbf{x}) = \frac{1}{r^2} \left(\frac{\mathbf{n}}{|\mathbf{n}|} \cdot \mathbf{L} \right) . \tag{1}$$

Here, $\mathbf{x} = (x_1, x_2)^\top \in \Omega \subset \mathbb{R}^2$ denotes a pixel position in the rectangular image plane, $\cdot$ the scalar product, $\mathbf{L}$ the normalised light direction, $\mathbf{n}$ the surface normal vector, $|\cdot|$ the Euclidean norm, and r the distance from the light source to the surface point.

Please note that in the literature this equation is typically parametrised such that the light source is located in the optical centre of the camera [12,16,17]. We will denote this specific variant of Eq. (1) as *restricted brightness equation*. In our approach, however, we follow a more general approach. We allow the light source to be everywhere in the image and parametrise the surface and thus the surface normal accordingly.

A sketch that illustrates the general scene geometry that comes with our model is depicted in Fig. 1. As one can see, w.l.o.g. we have chosen the origin of the coordinate system such that it coincides with the location of the light source. This decision will allow us to derive a mathematical model that is compact and elegant at the same time. As a first step towards this model, we have to parametrise the surface of the object

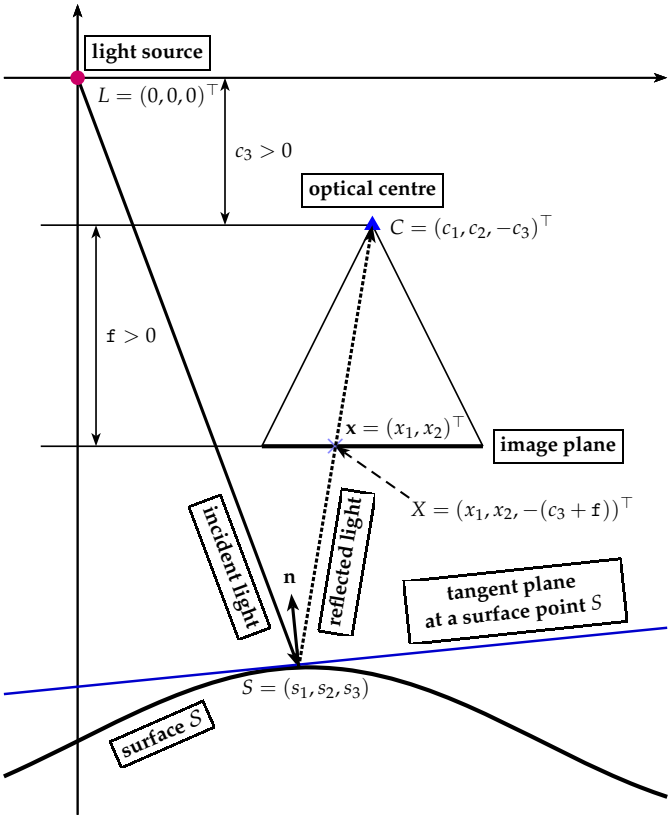


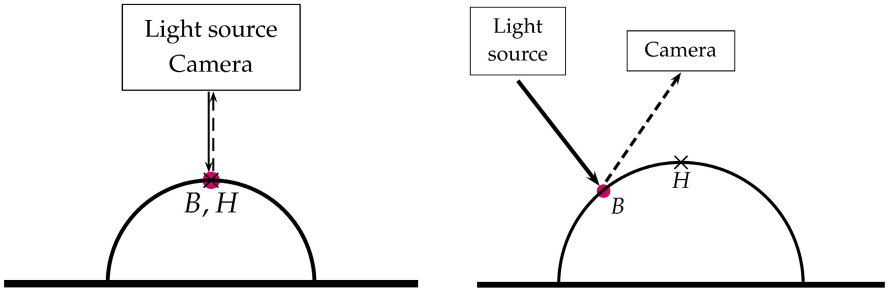
Fig. 1. Cross section of a 3-D model for perspective SfS with arbitrary position of the light source. The distance between the light source L and the point S on the surface is denoted by r in Eq. (1).

to compute the corresponding surface normal. This will be done in the next section. Afterwards we can derive the *general brightness equation* – the brightness equation that is parametrised such that it allows for an arbitrary position of the light source.

3 Parametrisation of the Surface

When it comes to the parametrisation of the object surface, recent SfS methods make use of standard Cartesian coordinates [9,16,17]. In the general case, however, such coordinates have one decisive drawback illustrated in Fig. 2(a) and Fig. 2(b): When the light source is not located in the optical centre of the camera, the *critical points*, i.e. the points of the object with largest local height, are not any longer the brightest points, i.e. the points that are closest to the light source. In short: Local intensity maxima do not identify critical points (local surface maxima). This is due to the fact that in SfS with Cartesian coordinates, the depth is measured along the x_3-axis (Fig. 2: vertical axis).

Since identifying critical points is required to apply efficient algorithms of fast marching (FM) type [22,23,24,25,26] to solve the brightness equation in (1), we propose the



(a) Light source located at optical centre. Brightest and highest point (B, H) coincide.

(b) Light source *not* located at optical centre. Brightest and highest point (B, H) differ(!).

Fig. 2. Relationship between the brightest point B (critical point) and the highest point H of the object depending on the scene geometry. The problem of differing H and B is inherent to Cartesian coordinates. In a spherical coordinates with origin at the light source, B is always H.

following solution to the problem: By considering a *spherical coordinate system* with the origin placed at the position of the light source, we measure the depth and thus the critical points from the viewpoint of the light source. Per construction, in such a coordinate system, the locally brightest points in the image coincide again with the critical points. Please recall in this context that (i) the albedo is assumed to be uniform, (ii) surface normals at local maxima are parallel to the direction of incoming light (per definition of local maxima in our new coordinate system) and (iii) remaining convex-concave ambiguities are resolved by the light fall-off factor $1/r^2$ in the brightness equation [12].

Let us now describe our parametrisation. To this end, we start with standard Cartesian coordinates and then derive the corresponding formulation for the spherical case.

3.1 Surface Representation in Cartesian Coordinates

Considering Cartesian coordinates and following the notation from Fig. 1, the vector from the camera position C to a point X in the image plane is given by

$$\overrightarrow{CX} = \overrightarrow{LX} - \overrightarrow{LC} = \begin{bmatrix} x_1 \\ x_2 \\ -(c_3 + \mathtt{f}) \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \\ -c_3 \end{bmatrix} = \begin{bmatrix} x_1 - c_1 \\ x_2 - c_2 \\ -\mathtt{f} \end{bmatrix}, \tag{2}$$

where $\overrightarrow{AB}$ stands for a vector with starting point A and endpoint B. Furthermore we can use (2) to express the vector between the light source L and the surface point S

$$\begin{aligned} \overrightarrow{LS} &= \overrightarrow{LC} + \overrightarrow{CS} = \overrightarrow{LC} + \lambda \overrightarrow{CX} \\ &= \begin{bmatrix} c_1 \\ c_2 \\ -c_3 \end{bmatrix} + \lambda \begin{bmatrix} x_1 - c_1 \\ x_2 - c_2 \\ -\mathtt{f} \end{bmatrix} = \begin{bmatrix} \lambda x_1 + (1-\lambda) c_1 \\ \lambda x_2 + (1-\lambda) c_2 \\ -(c_3 + \lambda \mathtt{f}) \end{bmatrix} =: \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}, \end{aligned} \tag{3}$$

where $\lambda \in \mathbb{R}$ is a scaling factor that we are looking for. In particular, it holds that

$$|\overrightarrow{LS}|^2 = s_1^2 + s_2^2 + s_3^2 = [c_1 + \lambda\,(x_1 - c_1)]^2 + [c_2 + \lambda\,(x_2 - c_2)]^2 + (c_3 + \lambda\,\mathtt{f})^2\,. \quad (4)$$

In the following, we refrain from estimating the scaling factor λ, but solve for the distance $s_1^2 + s_2^2 + s_3^2$ from the light source to the surface directly. To this end, it turns out once again, that is advantageous to consider the problem in spherical coordinates.

3.2 Surface Representation in Spherical Coordinates

In order to express the distance from the light source to the surface in spherical coordinates, we have to define a suitable basis. Following Fig. 3, we represent the Cartesian vector $\mathbf{r}$ via two angles θ and φ, respectively, as well as a radius r:

$$\mathbf{r} = R_{x_3}(\theta)\; R_{x_2}(\varphi) \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} = \begin{bmatrix} \cos\theta\,\cos\varphi & -\sin\theta & \cos\theta\,\sin\varphi \\ \sin\theta\,\cos\varphi & \cos\theta & \sin\theta\,\sin\varphi \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix}. \quad (5)$$

Here, the two matrices

$$R_{x_3}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R_{x_2}(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \quad (6)$$

represent rotations around the x_3- and x_2-axis. The corresponding orthonormal basis is

$$\mathbf{e}_\varphi = \begin{bmatrix} \cos\varphi\,\cos\theta \\ \cos\varphi\,\sin\theta \\ -\sin\varphi \end{bmatrix}, \qquad \mathbf{e}_\theta = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}, \qquad \mathbf{e}_r = \begin{bmatrix} \sin\varphi\,\cos\theta \\ \sin\varphi\,\sin\theta \\ \cos\varphi \end{bmatrix}, \quad (7)$$

where

$$\begin{bmatrix} \varphi \\ \theta \end{bmatrix} = \begin{bmatrix} \arccos \dfrac{s_3}{\sqrt{s_1^2 + s_2^2 + s_3^2}} \\ \arctan \dfrac{s_2}{s_1} \end{bmatrix}. \quad (8)$$

Thus, we can express the distance from the light source to the surface via the relation

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} =: \mathbf{s} := r\,\mathbf{e}_r \qquad \text{with} \qquad r = \sqrt{s_1^2 + s_2^2 + s_3^2}\,. \quad (9)$$

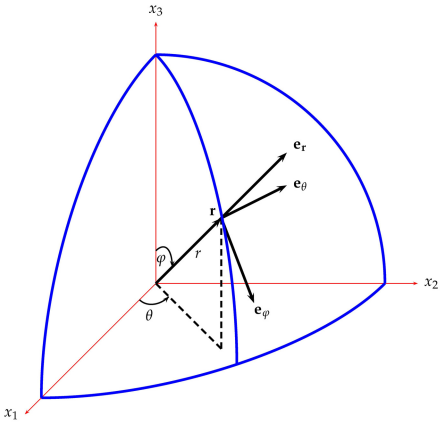


Fig. 3. The spherical system employed in this work. x_1, x_2 and x_3 are conventional Cartesian coordinate axes. $\mathbf{e}_r, \mathbf{e}_\varphi$ and $\mathbf{e}_\theta$ stand for basis vectors with respect to each direction in the spherical system. The distance between the light source L and the point S on the surface corresponds to r.

4 Computation of the Surface Normal

After we have parametrised the distance to the surface in spherical coordinates, we have to compute the *surface normal* for each pixel of the input image. This can be done by first determining the vectors defining the tangent plane – these vectors are given by the derivatives of the surface with respect to the two directions orthogonal to $\mathbf{r}$, namely θ and φ – and then by computing the cross product to obtain the corresponding normal vector. Using the definition of $\mathbf{r}$ from (9), the surface normal is then given by

$$
\begin{aligned}
\mathbf{n} &= \frac{\partial\,(r\mathbf{e}_r)}{\partial\theta} \times \frac{\partial\,(r\mathbf{e}_r)}{\partial\varphi} \\
&= \left(\frac{\partial\, r}{\partial\theta}\mathbf{e}_r + r\frac{\partial\,\mathbf{e}_r}{\partial\theta}\right) \times \left(\frac{\partial\, r}{\partial\varphi}\mathbf{e}_r + r\frac{\partial\mathbf{e}_r}{\partial\varphi}\right) \\
&= \left(\frac{\partial\, r}{\partial\theta}\mathbf{e}_r \times \frac{\partial\, r}{\partial\varphi}\mathbf{e}_r\right) + \left(\frac{\partial\, r}{\partial\theta}\mathbf{e}_r \times r\frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) + \left(r\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \frac{\partial\, r}{\partial\varphi}\mathbf{e}_r\right) + \left(r\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times r\frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) \\
&= \frac{\partial\, r}{\partial\theta}\frac{\partial\, r}{\partial\varphi}\underbrace{(\mathbf{e}_r \times \mathbf{e}_r)}_{=0} + r\frac{\partial\, r}{\partial\theta}\left(\mathbf{e}_r \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) + r\frac{\partial\, r}{\partial\varphi}\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \mathbf{e}_r\right) + r^2\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) \\
&= r\,\frac{\partial\, r}{\partial\theta}\left(\mathbf{e}_r \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) + r\,\frac{\partial\, r}{\partial\varphi}\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \mathbf{e}_r\right) + r^2\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) \\
&\overset{(11)}{=} r\,\frac{\partial\, r}{\partial\theta}\,(\mathbf{e}_r \times \mathbf{e}_\varphi) + r\,\frac{\partial\, r}{\partial\varphi}\,(\sin\varphi\,\mathbf{e}_\theta \times \mathbf{e}_r) + r^2\,(\sin\varphi\,\mathbf{e}_\theta \times \mathbf{e}_\varphi) \\
&\overset{(12)}{=} r\,\frac{\partial\, r}{\partial\theta}\,\mathbf{e}_\theta + r\,\sin\varphi\,\frac{\partial\, r}{\partial\varphi}\,\mathbf{e}_\varphi - r^2\,\sin\varphi\,\mathbf{e}_r\,.
\end{aligned}
\tag{10}
$$

Thereby we used the following relations that hold by definition

$$\frac{\partial\,\mathbf{e}_r}{\partial\varphi} = \begin{bmatrix} \cos\varphi\,\cos\theta \\ \cos\varphi\,\sin\theta \\ -\sin\varphi \end{bmatrix} \overset{(7)}{=} \mathbf{e}_\varphi\ , \qquad \frac{\partial\,\mathbf{e}_r}{\partial\theta} = \begin{bmatrix} -\sin\varphi\,\sin\theta \\ \sin\varphi\,\cos\theta \\ 0 \end{bmatrix} \overset{(7)}{=} \sin\varphi\,\mathbf{e}_\theta\,. \tag{11}$$

as well as the fact that $(\mathbf{e}_\varphi, \mathbf{e}_\theta, \mathbf{e}_r)$ constitutes a right-handed system, i.e. we have

$$\mathbf{e}_r \times \mathbf{e}_\varphi = \mathbf{e}_\theta\ , \qquad \mathbf{e}_\varphi \times \mathbf{e}_\theta = \mathbf{e}_r\ , \qquad \mathbf{e}_\theta \times \mathbf{e}_r = \mathbf{e}_\varphi\ . \tag{12}$$

5 Generalised Brightness Equation

After we have computed the surface normal, we are now in the position to set up the brightness equations for the general case. Following Eq. (1) this requires to evaluate the expressions $\mathbf{n}\cdot\mathbf{L}$ and $|\mathbf{n}|$. Based on Fig. 1, the light direction is given by

$$\mathbf{L} = -\,\mathbf{e}_r\,. \tag{13}$$

When plugging (10) and (13) into $\mathbf{n}$ and $\mathbf{L}$, respectively, the dot product $\mathbf{n}\cdot\mathbf{L}$ becomes

$$\begin{aligned} \mathbf{n}\cdot\mathbf{L} &= \left(r\,\frac{\partial\,r}{\partial\theta}\,\mathbf{e}_\theta + r\,\sin\varphi\,\frac{\partial\,r}{\partial\varphi}\,\mathbf{e}_\varphi - r^2\,\sin\varphi\,\mathbf{e}_r\right)\cdot(-\,\mathbf{e}_r) \\ &= r^2\,\sin\varphi \end{aligned} \tag{14}$$

while the (squared) magnitude of the surface normal is given by the expression

$$|\mathbf{n}|^2 = \mathbf{n}\cdot\mathbf{n} = r^2\,\left[\left(\frac{\partial\,r}{\partial\theta}\right)^2 + \sin^2\varphi\left(\frac{\partial\,r}{\partial\varphi}\right)^2 + r^2\,\sin^2\varphi\right]\,. \tag{15}$$

In both cases we have exploited the orthonormality of the basis vectors; see Eq. (12). Using our results from (14) and (15) in Eq. (1) then gives the brightness equation

$$\begin{aligned} & I = \frac{1}{r^2}\left(\frac{\mathbf{n}}{|\mathbf{n}|}\cdot\mathbf{L}\right) \\ \Rightarrow\ & r^2\,I\,|\mathbf{n}| - \mathbf{n}\cdot\mathbf{L} = 0 \\ \Rightarrow\ & r^3\,I\,\sqrt{\left(\frac{\partial\,r}{\partial\theta}\right)^2 + \sin^2\varphi\left(\frac{\partial\,r}{\partial\varphi}\right)^2 + r^2\,\sin^2\varphi} - r^2\,\sin\varphi = 0 \\ \Rightarrow\ & I\,\sqrt{\frac{1}{r^2\,\sin^2\varphi}\left(\frac{\partial\,r}{\partial\theta}\right)^2 + \frac{1}{r^2}\left(\frac{\partial\,r}{\partial\varphi}\right)^2 + 1} - \frac{1}{r^2} = 0 \end{aligned} \tag{16}$$

We can further simplify this equation using the following relation:

$$\nabla r = \frac{1}{r}\left(\frac{\partial\,r}{\partial\varphi}\right)\mathbf{e}_\varphi + \frac{1}{r\sin\varphi}\left(\frac{\partial\,r}{\partial\theta}\right)\mathbf{e}_\theta\,. \tag{17}$$

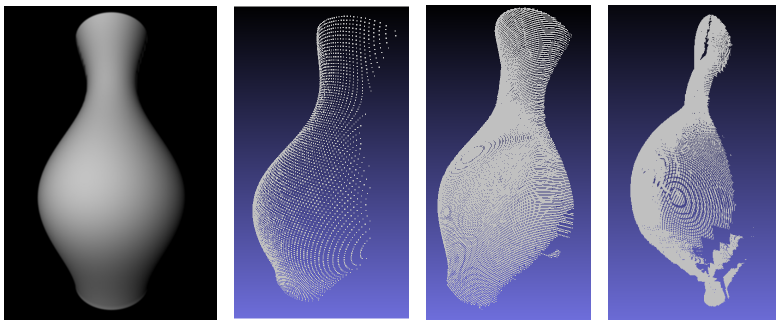
Thus we finally obtain a very compact and elegant formulation that we will denote as *generalised brightness equation*. It is given by the Hamilton-Jacobi equation (HJE):

$$\boxed{I\sqrt{|\nabla r|^2+1}-\frac{1}{r^2}=0\,.} \tag{18}$$

6 A Fast Marching Scheme for Spherical Coordinates

After we have derived the generalised brightness equation in the previous section, let us now discuss how it can be efficiently solved for the unknown radial distance field r (solved in the viscosity sense [24]). For this kind of HJEs so called fast marching (FM) schemes are among the fastest solvers in the literature [22,23,24,25,26]. Starting from critical points, such schemes are based on propagating the solution to the remaining points on the surface. Typically, each pixel is only visited once such that in the optimal case the performance is linear in the number of pixels [25]. Unfortunately, standard FM schemes cannot be applied in our case, since they have been designed for eikonal-type of the form $H(\cdot,\nabla r)$ without an explicit dependency on r. Moreover, our HJE is formulated in terms of spherical coordinates such that I actually depends on the solution r via the parametrised Cartesian pixel position $\mathbf{x}=(x(\theta,\varphi,r),y(\theta,\varphi,r))^\top$. Therefore, we propose the following specifically tailored variant of the FM scheme to solve our general HJE of type $H(\cdot,\nabla r,r)$ in spherical coordinates:

1. We identify critical points of the surface based on their brightness (cf. Section 3). Since their distance to the light source is minimal, we know that $\nabla r=0$, which in turn allows us to solve Eq. (18) directly for the radial depth r.
2. The main task of the FM update process is to spread information from the critical points to the other points on the surface and update them solving Eq. (18). Since our HJE is highly nonlinear and depends on both r and ∇r, we apply the iterative strategy proposed in [17] in the context of nonlinear HJEs for Euclidean coordinates and solve (18) using the classical *regula falsi* method. Thereby, spatial derivatives are discretised using the standard upwind scheme [18]. Since our algorithm works in spherical coordinates, we propagate the information to neighbouring locations in terms of θ and φ rather than x and y. This also requires to evaluate the brightness values of the input image at subpixel locations, which is realised in terms of bilinear interpolation. Moreover, we need an additional *correction step*: Since the location to evaluate the image brightness depends on r (see above), we have to update this brightness value each iteration within our iterative regular falsi framework. The iterations are stopped, if the residual of the equation drops below a certain threshold.
3. We proceed to the adjacent locations in terms of θ and φ and solve (18) there. Although the parametrisation is different, the order in which the locations are traversed, is analogue to the Euclidean case (see [17] for details).



(a) Vase input image. (b) Ground truth. (c) Our method. (d) Standard SfS.

Fig. 4. The Vase experiment

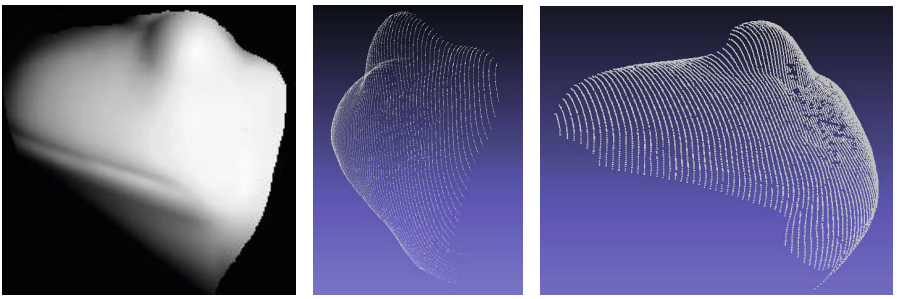
7 Experiments

Let us now evaluate our perspective SfS method for the general case. To this end, we have considered both a synthetic and a real-world image.

In our first experiment, we used our method to reconstruct the shape of a vase. The corresponding input image is depicted in Fig. 4(a). As one can see the light source is located in the upper left corner of the scene. The result obtained by our method as well as the ground truth are shown in Fig. 4(c) and Fig. 4(b), respectively. Evidently, the reconstructed shape looks very realistic. Moreover, in order to evaluate the impact of the general model on the reconstruction quality, we compared our result to the one of a perspective SfS method based on the restricted brightness equation, i.e. where the light source is assumed to be in the optical centre of the camera. As one can see from the corresponding result in Fig. 4(d), this method fails completely. This shows that the generalised brightness equation is essential for the success of SfS in practical applications – in particular if the assumption that the light source is close to the optical centre of the camera does not hold.

In our second experiment, we applied our perspective SfS technique for the general case to a real-world image. This image shows a sculpture that depicts the head of a frog (see Fig. 5(a)). This time, the light source is located in the right centre of the scene. Once again, we can see that the reconstruction looks reasonable. Thereby, we have to keep in mind that the head of the frog is reconstructed from the viewpoint of the light source. Moreover, this experiment shows nicely that the discrepancy between the position of the light source and the optical centre of the camera should not be too large. While the general model is capable of handling such situations, the overlap between both viewpoints may become quite small. In that case, the reconstruction will only show a very small part of the original object. In our experiment, however, the reconstructed part is still sufficiently large to give a good impression of the overall object surface.

The runtime of our approach is in the order of 40 seconds for a megapixel result, i.e. a reconstruction of size 1024×1024. Please note that this runtime is not related to the size of the input image, but the angular sampling of the radial depth (i.e. of θ, φ).



(a) Frog input image. (b) Our method. (c) Our method, rotated view.

Fig. 5. The Frog experiment

8 Summary

In this paper we proposed a novel model for perspective SfS for the general case. Unlike previous methods that restricted the position of the light source to be located in the optical centre of the camera, our model allows the light source to be placed anywhere in the scene. In this context, a formulation of the problem as Hamilton-Jacobi equation in terms of spherical coordinates turned out to be very useful: On the one hand, it allowed us to formulate the brightness equation for the complex general case in a very compact and elegant way. On other hand, it enabled us to determine critical points and thus to develop a specifically tailored variant of the highly efficient FM scheme as solver for our model. Experiments have shown that our method works well in practice and that it gives reconstructions of good quality. It even allows to obtain results in those cases where standard models based on the restricted brightness equation fail. This shows that considering alternative parametrisations can be worthwhile in many computer vision problems. They may turn an originally difficult problem into a simple one - from both a modelling and a numerical viewpoint.

Acknowledgements. This work has been partly funded by the Deutsche Forschungsgemeinschaft (BR 2245/3-1, BR 4372/1-1). Moreover, Silvano Galliani gratefully acknowledges funding by the Fraunhofer Institute for Industrial Mathematics (ITWM).

References

1. Rindfleisch, T.: Photometric method for lunar topography. Photogrammetric Engineering 32, 262–277 (1966)
2. Horn, B.K.P.: Shape from shading: a method for obtaining the shape of a smooth opaque object from one view. PhD thesis. Massachusetts Institute of Technology (1970)
3. Ostrov, D.N.: Viscosity solutions and convergence of monotone schemes for synthetic aperture radar shape-from-shading equations with discontinuous intensities. SIAM Journal on Applied Mathematics 59, 2060–2085 (1999)
4. Bors, A.G., Hancock, E.R., Wilson, R.C.: Terrain analysis using radar shape-from-shading. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 974–992 (2003)

5. Abdelrahim, A.S., Abdelrahman, M.A., Abdelmunim, H., Farag, A., Miller, M.: Novel image-based 3D reconstruction of the human jaw using shape from shading and feature descriptors. In: Proceedings of the British Machine Vision Conference, pp. 1–11 (2011)
6. Okatani, T., Deguchi, K.: Shape reconstruction from an endoscope image by shape from shading technique for a point light source at the projection center. Computer Vision and Image Understanding 66, 119–131 (1997)
7. Tankus, A., Sochen, N., Yeshurun, Y.: Perspective shape-from-shading by fast marching. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 43–49 (2004)
8. Wang, G.H., Han, J.Q., Zhang, X.M.: Three-dimensional reconstruction of endoscope images by a fast shape from shading method. Measurement Science and Technology 20 (2009)
9. Wu, C., Narasimhan, S., Jaramaz, B.: A multi-image shape-from-shading framework for near-lighting perspective endoscopes. International Journal of Computer Vision 86, 211–228 (2010)
10. Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape from shading: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 690–706 (1999)
11. Durou, J.D., Falcone, M., Sagona, M.: Numerical methods for shape-from-shading: A new survey with benchmarks. Computer Vision and Image Understanding 109, 22–43 (2008)
12. Prados, E., Faugeras, O.: Shape from shading: A well-posed problem? In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 870–877 (2005)
13. Courteille, F., Crouzil, A., Durou, J.D., Gurdjos, P.: Towards shape from shading under realistic photographic conditions. In: IEEE International Conference on Pattern Recognition, vol. 2, pp. 277–280 (2004)
14. Tankus, A., Sochen, N., Yeshurun, Y.: Shape-from-shading under perspective projection. International Journal of Computer Vision 63, 21–43 (2005)
15. Bruvoll, S., Reimers, M.: Spherical surface parameterization for perspective shape from shading. Pattern Recognition Letters 33, 33–40 (2012)
16. Ahmed, A., Farag, A.: A new formulation for shape from shading for non-Lambertian surfaces. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1817–1824 (2006)
17. Vogel, O., Breuß, M., Weickert, J.: Perspective shape from shading with non-lambertian reflectance. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 517–526. Springer, Heidelberg (2008)
18. Rouy, E., Tourin, A.: A viscosity solutions approach to shape-from-shading. SIAM Journal on Numerical Analysis 29, 867–884 (1992)
19. Crandall, M., Lions, P.L.: Viscosity solutions of Hamilton-Jacobi equations. Transactions of the American Mathematical Society 277, 1–42 (1983)
20. Horn, B., Brooks, M.: The variational approach to shape from shading. Computer Vision, Graphics, and Image Processing 33, 174–208 (1986)
21. Alkhalifah, T., Fomel, S.: Implementing the fast marching eikonal solver: spherical versus Cartesian coordinates. Geophysical Prospecting 49, 165–178 (2001)
22. Sethian, J.: A fast marching level set method for monotonically advancing fronts. Proc. of the National Academy of Sciences of the United States of America, 1591–1595 (1995)
23. Sethian, J.: Fast-marching level-set methods for three-dimensional photolithography development. In: Proc. SPIE, Optical Microlithography IX., vol. 2726, pp. 262–272 (1996)
24. Sethian, J.: Level set methods and fast marching methods. Cambridge University Press (1999)
25. Tsitsiklis, J.: Efficient algorithms for globally optimal trajectories. IEEE Transactions on Automatic Control 40, 1528–1538 (1995)
26. Helmsen, J., Puckett, E., Colella, P., Dorr, M.: Two new methods for simulating photolithography development in 3D. In: Proc. SPIE, Optical Microlithography IX, vol. 2726, pp. 253–261 (1996)

Weighted Patch-Based Reconstruction: Linking (Multi-view) Stereo to Scale Space

Ronny Klowsky, Arjan Kuijper, and Michael Goesele

Technische Universität Darmstadt

Abstract. Surface reconstruction using patch-based multi-view stereo commonly assumes that the underlying surface is locally planar. This is typically not true so that least-squares fitting of a planar patch leads to systematic errors which are of particular importance for multi-scale surface reconstruction. In a recent paper [12], we determined the modulation transfer function of a classical patch-based stereo system. Our key insight was that the reconstructed surface is a box-filtered version of the original surface. Since the box filter is not a true low-pass filter this causes high-frequency artifacts. In this paper, we propose an extended reconstruction model by weighting the least-squares fit of the 3D patch. We show that if the weighting function meets specified criteria the reconstructed surface is the convolution of the original surface with that weighting function. A choice of particular interest is the Gaussian which is commonly used in image and signal processing but left unexploited by many multi-view stereo algorithms. Finally, we demonstrate the effects of our theoretic findings using experiments on synthetic and real-world data sets.

Keywords: multi-view stereo, multi-scale surface reconstruction.

1 Introduction

The basis of virtually all multi-view stereo algorithms are correspondences found between images. Hereby, the de facto standard is to find a planar patch in 3D whose projected region in (some of) the images is photo-consistent, i.e., looks similar. There are many ways to measure photo-consistency including normalized cross-correlation (NCC) or the sum of squared differences (SSD, see Hu and Mordohai [10] for an overview and evaluation of different measures). Whatever measurement used, the underlying assumption is that the original surface is locally planar or even has constant depth in the patch area. This leads to a systematic error in reconstruction which becomes especially important when combining multi-scale data [1, 2]. Recently, Klowsky et al. [12] analyzed this systematic error and proposed a reconstruction model where the 3D patch is fitted to the original surface in a least-squares sense. In the resulting linear system they identified the modulation transfer function to be a sinc. In other words, the reconstructed surface is equal to a convolution of the original surface with a box filter. Since this is no true low-pass filter it causes high-frequency artifacts such as amplitude inversion for some frequencies.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 234–245, 2013.

In this paper, we develop an extended reconstruction model by weighting the fitting of the 3D patch. We derive constraints on the weighting function to ensure that the reconstructed surface is a convolution of the original surface with that weighting function. As a particular result, we will see that uniform weighting used in our previous work [12] causes the box filter effect. A much better choice for the weighting function fulfilling the derived constraints and allowing for true low-pass filtered reconstructions is the Gaussian, which is widely used in the imaging domain. When using different patch sizes (e.g., due to different image resolution or camera-object distances) the reconstructions reflect different levels of the scale space representation of the true surface. We show for one popular multi-view stereo algorithm [5] how to implement the weighting and discuss results on synthetic as well as real-world data sets. Our findings may influence a broad range of algorithms in multi-view stereo but also in the field of multi-scale surface reconstruction [2–4, 15] or geometry super-resolution [6, 20].

In summary the contributions of our paper are

- the generalization of a previously presented reconstruction model for (multi-view) stereo by introducing weights,
- the theoretical derivation of the (predicted) reconstructed surface without the detour in frequency space, and
- we show how a weighting, e.g., a Gaussian, can be implemented for a common multi-view stereo algorithm which expectably improves the frequency behavior of the reconstruction.

1.1 Related Work

While there is a large body of work on multi-view stereo (see, e.g., the survey paper and the constantly updated benchmark by Seitz at al. [14, 18]), the study of multi-scale depth reconstruction has long been neglected. In previous work we [12] introduced a theoretical reconstruction model and determined the modulation transfer function of patch-based stereo systems. We also discussed the (loosely) related work on multi-scale analysis of (multi-view) stereo to which we refer the reader for a more extensive discussion. Our current work builds upon this reconstruction model and demonstrates how more freedom in the reconstruction outcome is possible. As one particular result, we demonstrate that multi-view stereo can yield a scale space representation of the underlying geometry. In contrast to [12], we derive our results directly in geometry space without operation (at least in an intermediate step) in frequency space.

Our work is also related to existing work on patch-based photo-consistency measures. An overview and evaluation of confidence measures used in (multi-view) stereo is given by Hu and Mordohai [10]. In all their cost computations, however, a square patch of $N \times N$ pixels is used and all pixels are weighted uniformly. If we assume all measures aim at fitting a patch in 3D space, they all result in a box filter. Kanade and Okutomi [11] already tried to find optimal size and shape of the patch but still only used rectangular shapes. Habbecke and Kobbelt [8] propose a multi-view stereo system where matching is performed on

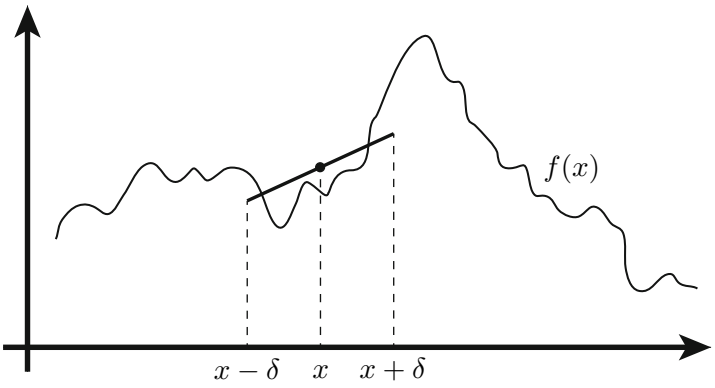


Fig. 1. Fitting a planar patch (line segment) to the geometry for each point x

circular disks in object space. The size of the disks is selected to achieve a minimum intensity variance on each disk. Totally different shapes are achieved by Micusik and Koseka [13] whose approach is suited for man-made environments with many planar surfaces. Here, the reference view is first segmented into superpixels, that are assumed to be planar in object space, and matching is then performed using those superpixels. Thus the shape of the matching window is adapted to the local scene structure and texture. Yoon and Kweon [21] were probably the first to compute weights for each pixel in the patch that steer the influence of that pixel in the matching process. Their weights are dependent on the color similarity and the spatial distance from the center pixel. Hosni et al. [9] improve on that by computing weights using the geodesic distance transform. In contrast to all these efforts, we investigate the influence of a specific weighting on the reconstructed geometry and derive the resulting (multi-scale) behavior of the resulting surface.

2 Theoretical Considerations

2.1 Extension of the Reconstruction Model

In this paper, we build upon our previously introduced reconstruction model [12]. We describe the process of photometric consistency optimization between images (e.g. using normalized cross-correlation (NCC), or sum of squared differences (SSD)) as a geometric least-squares fitting of a planar patch to the unknown geometry. Figure 1 visualizes this idea for a 2D geometry described as a height field $z = f(x)$. To obtain the reconstruction at some point x, a line segment (parameterized by slope m and offset n) with extent 2δ is fitted to the geometry in a least-squares sense minimizing the energy

$$E(m, n, x) = \int_{x-\delta}^{x+\delta} (mt + n - f(t))^2 dt. \tag{1}$$

The reconstructed surface is then represented by the central patch points. For this model we determined the modulation transfer function which turned out to be a sinc. Though not explicitly stated in our prior paper [12] this is equivalent to a convolution with a box filter. In the following we will show that the reason for this result is the uniform weighting of pixels during optimization. We suggest the following extension of the reconstruction model: Instead of considering each point in $[x-\delta, x+\delta]$ uniformly we introduce a weighting function g allowing for different areas of influence. Consequently, we alter the energy function to

$$E(m,n,x) = \int_{-\infty}^{\infty} g(x-t)(mt+n-f(t))^2\,dt \tag{2}$$

where $g(t)$ is a weighting function. Note that with $g(t) = \mathbf{1}_{[-\delta,\delta]}$ this is equal to the former energy in Eq. 1. This weighting function could be implemented as a weighting of the pixels during photo-consistency optimization. In Section 3 we will demonstrate this using a specific multi-view stereo algorithm. In the following subsection, we derive theoretically how this weighting function affects the reconstructed surface.

2.2 Reconstruction in 2D

For the sake of simplicity, we first look at a surface in 2D (a line) as illustrated in Figure 1. For now, we put no further constraints on $g(t)$ except for integrability. Later on, we will discuss further desirable properties. Minimizing E in Equation 2 requires taking the partial derivatives with respect to m and n:

$$\begin{aligned}\partial_m E &= 2\int_{-\infty}^{\infty} g(x-t)t(mt+n-f(t))\,dt \qquad (3)\\ &= 2m\int_{-\infty}^{\infty} g(x-t)t^2\,dt + 2n\int_{-\infty}^{\infty} g(x-t)t\,dt - 2\int_{-\infty}^{\infty} g(x-t)tf(t)\,dt\\ \partial_n E &= 2\int_{-\infty}^{\infty} g(x-t)(mt+n-f(t))\,dt \qquad (4)\\ &= 2m\int_{-\infty}^{\infty} g(x-t)t\,dt + 2n\int_{-\infty}^{\infty} g(x-t)\,dt - 2\int_{-\infty}^{\infty} g(x-t)f(t)\,dt\end{aligned}$$

We introduce a short notation for the zeroth, first and second moment of g

$$\mu_0 = \int_{-\infty}^{\infty} g(t)\,dt \qquad \mu_1(x) = \int_{-\infty}^{\infty} g(x-t)t\,dt \qquad \mu_2(x) = \int_{-\infty}^{\infty} g(x-t)t^2\,dt \tag{5}$$

and abbreviate the other convolution integrals using

$$(g*\cdot f)(x) = \int_{-\infty}^{\infty} g(x-t)tf(t)\,dt \tag{6}$$

$$(g*f)(x) = \int_{-\infty}^{\infty} g(x-t)f(t)\,dt. \tag{7}$$

W.l.o.g. we can assume that $\mu_0 = 1$ which corresponds to normalizing the weighting function g. Under the condition that $\mu_2(x) \neq 0$ we set the partial derivatives to zero and transpose the equations:

$$m = \frac{(g * \cdot f)(x) - n\mu_1(x)}{\mu_2(x)} \tag{8}$$

$$n = (g * f)(x) - m\mu_1(x) \tag{9}$$

We can now solve for m and n which leads to

$$\begin{aligned} m &= \frac{(g * \cdot f)(x) - ((g * f)(x) - m\mu_1(x))\mu_1(x)}{\mu_2(x)} \\ \Leftrightarrow\ m &= \left(1 - \frac{\mu_1(x)^2}{\mu_2(x)}\right)^{-1} \left(\frac{(g * \cdot f)(x)}{\mu_2(x)} - \frac{(g * f)(x)\mu_1(x)}{\mu_2(x)}\right) \\ &= \frac{(g * \cdot f)(x) - (g * f)(x)\mu_1(x)}{\mu_2(x) - \mu_1(x)^2} \end{aligned} \tag{10}$$

$$\begin{aligned} n &= (g * f)(x) - \frac{(g * \cdot f)(x) - (g * f)(x)\mu_1(x)}{\mu_2(x) - \mu_1(x)^2}\mu_1(x) \\ &= \frac{(g * f)(x)\mu_2(x) - (g * \cdot f)(x)\mu_1(x)}{\mu_2(x) - \mu_1(x)^2} \end{aligned} \tag{11}$$

Since the final surface is represented by the central patch points it can be written as

$$mx + n = \frac{(g * \cdot f)(x)(x - \mu_1(x)) + (g * f)(x)(\mu_2(x) - x\mu_1(x))}{\mu_2(x) - \mu_1(x)^2}. \tag{12}$$

Though valid for very general weighting functions g this result is not very satisfactory. On closer inspection we see that when $\mu_1(x) = x$, which is true for all normalized symmetric functions g, it can be easily simplified to

$$mx + n = (g * f)(x). \tag{13}$$

In other words, every function g with $\mu_0 = 1$, $\mu_1(x) = x$, $\mu_2(x) \neq 0$, and $\mu_2(x) \neq x^2$, used to weight the least-squares fitting results in a reconstruction that is the convolution of the true surface with g. Note, that a uniform weighting [12] naturally leads to the convolution with a box filter in this framework.

2.3 Building a Scale Space Representation

The derived constraints for the weighting function obviously allow for many different choices. One of particular interest is the Gaussian since convolutions with Gaussians are well studied and widely applied, e.g., in the image domain. If we set g to be a normalized Gaussian with standard deviation σ

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-t^2}{2\sigma^2}\right). \tag{14}$$

we obtain the following moments

$$\mu_0 = 1 \qquad \mu_1(x) = x \qquad \mu_2(x) = \sigma^2 + x^2. \tag{15}$$

That is, the normalized Gaussian fulfills our constraints and we can determine the slope m and offset n of the fitted patch at each point x by

$$m = \frac{(g * \cdot f)(x) - (g * f)(x)x}{\sigma^2} \tag{16}$$

$$n = \frac{(g * f)(x)(\sigma^2 + x^2) - (g * \cdot f)(x)x}{\sigma^2}. \tag{17}$$

In order to create a scale space representation of the underlying surface we need to use Gaussians with varying standard deviations σ. However, during reconstruction we can influence σ only to a limited extent because it depends on the scene depth, image resolution and focal length of the camera. In that sense, if we reconstruct depth maps of the same geometry using a variety of images results in a natural variation of the standard deviation σ in real-world space. The only parameter one can actively steer is the standard deviation σ_i (linked with the window size due to approximation and clamping of the Gaussian) in image space used for patch-based optimization. When selecting σ_i one often has a rough depth estimate and also the camera parameters are known from registration. With that it is possible to indirectly steer the standard deviation σ in world space at least to a limited extent, e.g., for parts of the scene with different depths. In Section 3 we will conduct some experiments with varying the standard deviation σ_i but we first transfer our results into 3D.

2.4 Reconstruction in 3D

For the reconstruction in 3D we assume the 2D geometry is described as a height field $z = f(x, y)$. To obtain the reconstruction at some point (x, y), we fit a patch (surface segment) that is parameterized by 2 slopes m_1 and m_2 and an offset n. Again, the weighting function g allows for different areas of influence. As a result we now have the following energy

$$E(m_1, m_2, n, x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x-t, y-s)(m_1 t + m_2 s + n - f(t,s))^2 \, dt \, ds. \tag{18}$$

Minimizing E requires taking the partial derivatives with respect to m_1, m_2, and n:

$$\partial_{m_1} E = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} 2tg(x-t, y-s)(m_1 t + m_2 s + n - f(t,s)) \, dt \, ds \stackrel{!}{=} 0 \tag{19}$$

$$\partial_{m_2} E = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} 2sg(x-t, y-s)(m_1 t + m_2 s + n - f(t,s)) \, dt \, ds \stackrel{!}{=} 0 \tag{20}$$

$$\partial_{n} E = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} 2g(x-t, y-s)(m_1 t + m_2 s + n - f(t,s)) \, dt \, ds \stackrel{!}{=} 0 \tag{21}$$

Similar to the reconstruction in 2D, we introduce the short notation μ_{00}, μ_{10}, μ_{01}, μ_{20}, μ_{11}, and μ_{02} for the moments of g with respect to x and y, respectively.

$$\mu_{00} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(t,s)\,dt\,ds, \qquad \mu_{10} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-t,y-s)t\,dt\,ds \tag{22}$$

$$\mu_{01} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-t,y-s)s\,dt\,ds, \quad \mu_{20} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-t,y-s)t^2\,dt\,ds \tag{23}$$

$$\mu_{11} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-t,y-s)st\,dt\,ds, \; \mu_{02} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-t,y-s)s^2\,dt\,ds \tag{24}$$

For the sake of clarity we chose an even shorter abbreviation for the other convolution integrals:

$$\mathrm{gtf} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} tg(x-t,y-s)f(t,s)\,dt\,ds \tag{25}$$

$$\mathrm{gsf} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} sg(x-t,y-s)f(t,s)\,dt\,ds \tag{26}$$

$$\mathrm{gf} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-t,y-s)f(t,s)\,dt\,ds. \tag{27}$$

Again, we can normalize g such that $\mu_{00} = 1$. With this notation we can rewrite Eqs. (19)-(21) as

$$\partial_{m_1} E = 2(m_1\mu_{20} + m_2\mu_{11} + n\mu_{10} - \mathrm{gtf}) \stackrel{!}{=} 0 \tag{28}$$

$$\partial_{m_2} E = 2(m_1\mu_{11} + m_2\mu_{02} + n\mu_{01} - \mathrm{gsf}) \stackrel{!}{=} 0 \tag{29}$$

$$\partial_{n} E = 2(m_1\mu_{10} + m_2\mu_{01} + n - \mathrm{gf}) \stackrel{!}{=} 0 \tag{30}$$

Solving these equations for m_1, m_2, and n yields

$$\alpha m_1 = \mathrm{gf}\,(\mu_{02}\mu_{10} - \mu_{01}\mu_{11}) + \mathrm{gsf}\,(\mu_{11} - \mu_{01}\mu_{10}) + \mathrm{gtf}\left(\mu_{01}^2 - \mu_{02}\right) \tag{31}$$

$$\alpha m_2 = \mathrm{gf}\,(\mu_{01}\mu_{20} - \mu_{10}\mu_{11}) + \mathrm{gsf}\left(\mu_{10}^2 - \mu_{20}\right) + \mathrm{gtf}\,(\mu_{11} - \mu_{01}\mu_{10}) \tag{32}$$

$$\alpha n = \mathrm{gf}\left(\mu_{11}^2 - \mu_{02}\mu_{20}\right) + \mathrm{gsf}\,(\mu_{01}\mu_{20} - \mu_{10}\mu_{11}) + \mathrm{gtf}\,(\mu_{02}\mu_{10} - \mu_{01}\mu_{11}) \tag{33}$$

where $\alpha = \mu_{20}\mu_{01}^2 - 2\mu_{10}\mu_{11}\mu_{01} + \mu_{02}\mu_{10}^2 + \mu_{11}^2 - \mu_{02}\mu_{20}$. Plugging in these expressions in the patch $P = m_1 x + m_2 y + n$, we obtain

$$P = \frac{1}{\alpha}\left(\mathrm{gf}\left(\mu_{11}^2 - \mu_{02}\mu_{20} - \mu_{01}\mu_{11}x + \mu_{02}\mu_{10}x - \mu_{10}\mu_{11}y + \mu_{01}\mu_{20}y\right) + \right. \tag{34}$$

$$\mathrm{gsf}\left(-\mu_{11}\mu_{10} + \mu_{01}\mu_{20} - \mu_{01}\mu_{10}x + \mu_{11}x + \mu_{10}^2 y - \mu_{20}y\right) + \tag{35}$$

$$\left.\mathrm{gtf}\left(-\mu_{11}\mu_1 + \mu_{02}\mu_{10} + \mu_{01}^2 x - \mu_{02}x - \mu_{10}\mu_{01}y + \mu_{11}y\right)\right). \tag{36}$$

Taking symmetric filters yields $\mu_{10} = x$ and $\mu_{01} = y$. Then immediately one gets

$$P = \mathrm{gf} \tag{37}$$

Of course we can use a classical anisotropic Gaussian characterized by σ and τ

$$g(t,s) = \frac{1}{2\pi\sigma\tau} \exp\left(\frac{-t^2}{2\tau^2} + \frac{-s^2}{2\sigma^2}\right) \tag{38}$$

because the moments are $\mu_{00} = 1$, $\mu_{10} = x$, $\mu_{01} = y$, $\mu_{02} = x^2 + \tau^2$, $\mu_{11} = xy$, $\mu_{02} = y^2 + \sigma^2$.

3 Experiments

In order to verify our theoretic findings in practice we now conduct some experiments. We hereby chose the depth map reconstruction method of Goesele et al. [5] because it does a pure photo-consistency optimization (going back to Gruen and Baltsavias [7]) to find depth and normal for a certain pixel and has no regularization force. For a small region around a pixel i, j in a reference view I_R the method aims to find depth d and normal $\boldsymbol{n}$ of the associated 3D patch such that it is photo-consistent with a set of neighboring views I_k. The algorithm minimizes (see [5, Sec. 6.2] ignoring the color scale)

$$\sum_{k,i,j} [I_R(s+i, t+j) - I_k(P_k^{d,\boldsymbol{n}}(s+i, t+j))]^2 \tag{39}$$

where P_k describes the projection of a pixel from the reference view in the neighbor view I_k according to some depth d and normal $\boldsymbol{n}$. We implement the weighting on the least-squares patch fit by weighting the pixels, i.e., we compute a weighted SSD:

$$\sum_{k,i,j} g(i,j)[I_R(s+i, t+j) - I_k(P_k^{d,\boldsymbol{n}}(s+i, t+j))]^2. \tag{40}$$

The remaining question is whether this weighted photo-consistency optimization still reflects the process of weighted least-squares fitting as described by Eq. 2. We test this using a synthetic data set because of two reasons: First, we can assure that our results are not affected by registration errors but solely reflect the photometric consistency optimization, and second, we know the ground truth surface and are able to compute the predicted reconstruction according to our model. Our ground truth surface is created as a random sum of one-dimensional B-Splines extruded into the third dimension. We then render five different views (one central view looking perpendicular onto the surface and four views distributed uniformly around it with a parallax of 35°) of this scene using the PBRT system [16] while a random texture is mapped onto the surface to guarantee matching success at all pixels (see Fig. 2). For the central view we now reconstruct a depth map by using the other four views as neighbors and minimizing

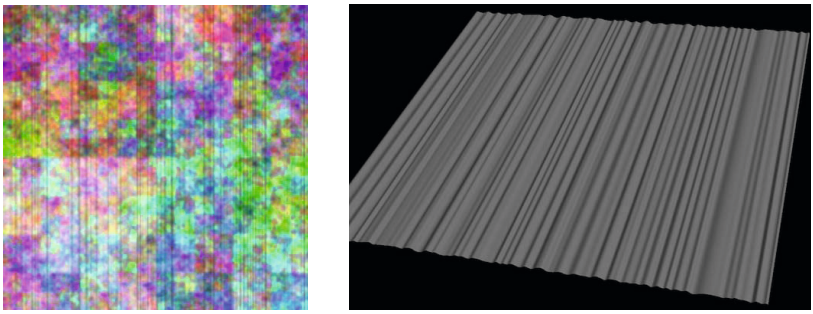


Fig. 2. Left: The central view of our synthetic data set. Right: The underlying mesh (shaded) used to render the views.

Table 1. Mean deviation of the reconstruction from the theoretical predicted surface (see Figs. 3&4)

Patch size	mean deviation (L_1-norm)	
in pixels	uniform weighting	Gaussian weighting
11×11	$1.9 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$
21×21	$4.1 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$
41×41	$6.9 \cdot 10^{-4}$	$5.8 \cdot 10^{-4}$
61×61	$6.3 \cdot 10^{-4}$	$7.0 \cdot 10^{-4}$

the weighted SSD from Eq. 40. We start the optimization for each pixel with the depth value obtained from PBRT and the normal representing a fronto-parallel patch. To reduce noise we average the reconstructed values along the constant dimension. Fig. 3 shows the reconstructions using a uniform weighting function. The quadratic windows in image space are 11 (blue), 21 (green), 41 (red), and 61 (cyan) pixels wide which corresponds to a patch size (2δ) of 0.06,0.12,0.24, and 0.36 in world coordinates, respectively. We also plotted the predicted reconstructions, i.e., convolutions of the original surface with box filters of the corresponding width. Overall, the reconstruction is close to the prediction although there is some local deviation. The best conformity is achieved for the small patch size which can also be seen in Table 1 where we computed the mean deviation. Note the occasional amplitude inversion visible in the prediction as well as the reconstruction, in particular for the largest filter at around -1.4.

In Fig. 4 we used Gaussian weighting with increasing standard deviation which leads to a scale space representation of the underlying surface. The window sizes are the same used for the uniform weighting and we always chose the standard deviation σ such that $\delta = 2.5\sigma$. That is, in world coordinates we used $\sigma = 0.012, 0.024, 0.048, 0.072$. We can see from the figure and also by studying the numbers in Table 1 that the deviation from the prediction again increases for larger σ.

Finally, we show reconstruction results on real world data. Figure 5 (top left) shows an input image of the Notre Dame data set consisting of 715 images

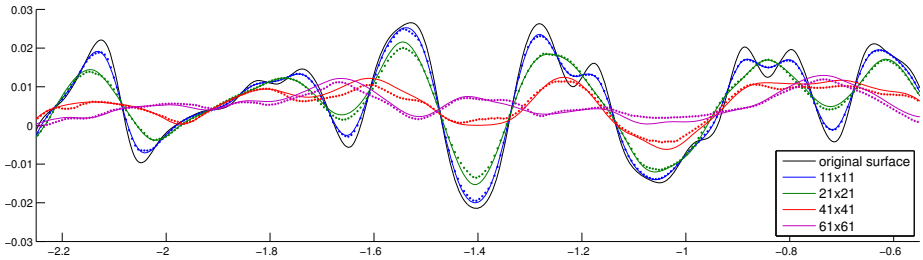


Fig. 3. Multi-view stereo reconstruction using a uniform weighting with increasing patch size. The black line denotes the original surface. The colored solid lines are the computed predictions while the corresponding dots are the reconstructed values.

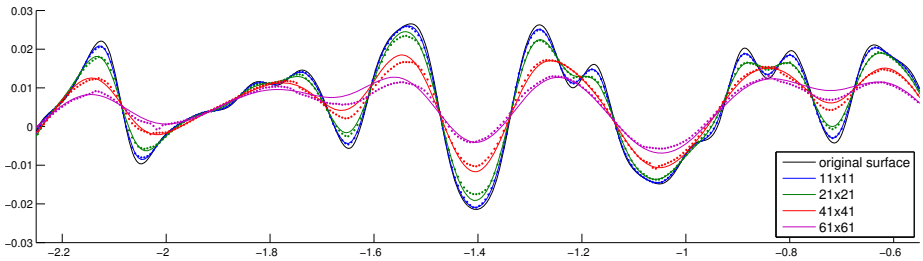


Fig. 4. Reconstructing a scale space representation using a Gaussian weighting with increasing standard deviation (see text). The black line denotes the original surface. The colored solid lines are the computed predictions while the according dots are the reconstructed values.

downloaded from the Internet. We use Snavely et al. [19] to register them and compute depth maps for the shown image using different weightings and window sizes. The middle and bottom row show reconstructions obtained using uniform and Gaussian weighting, respectively. Although hard to jugde, the Gaussian weighting seems to produce slightly more noise and less complete reconstructions. On the other hand it better preserves the low frequencies. One must consider though, that the algorithm [5] was tuned to work well with the uniform weighting and on a broad range of data sets. That is, playing with the parameters in the optimization or view selection might result in more favorable results for the Gaussian weighting.

4 Conclusion and Future Work

This paper extends a recently introduced model for patch-based depth reconstruction by adding a weighting function. We derive criteria on the weighting function such that we can predict the reconstructed surface as the convolution of the true surface with the applied weighting function. This includes using a Gaussian instead of a uniform weighting during reconstruction which corresponds to a

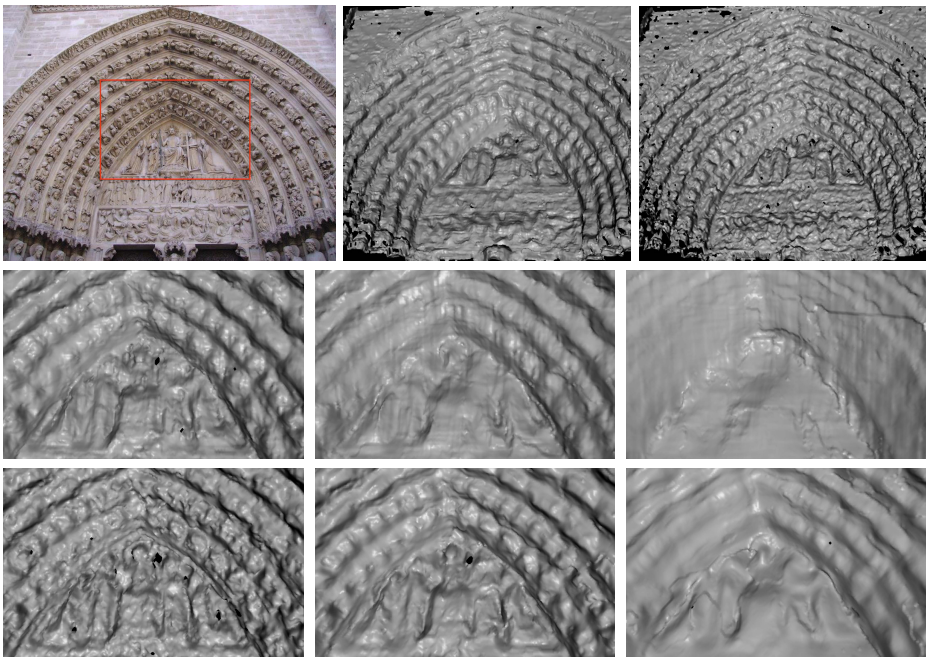


Fig. 5. *Top left*: Input image of the Notre Dame data set. The red box is roughly the area seen in the bottom rows. *Top middle,right*: Full rendered view of reconstructed depth map using uniform (middle) and Gaussian weighting (right) and a window size in images space of 7×7 pixels. *Middle+Bottom*: Enlarged area roughly corresponding to red box (top left) of the reconstructed depth map. We applied uniform (middle) and Gaussian weighting (bottom) using window sizes of 7×7, 11×11, and 21×21 pixels (from left to right) for reconstruction where the standard deviation of the Gaussian in image space is $\sigma_i = 1.2, 2.0, 4.0$.

Gaussian instead of a box filter in geometry space. In contrast to previous methods, we achieve a true low-pass filter avoiding the introduction of systematic high-frequency artifacts. Future work definitely includes to further investigate the correlation between weighted photo-consistency optimization and weighted least-squares fitting of a planar patch to the geometry.

Our findings are applicable in a broad range of applications. In contrast to [12], we give a local characterization of the reconstruction outcome at the same time offering more flexibility caused by the weighting. Multi-scale surface reconstruction methods like [2–4, 15] could take that knowledge into account when combining data from multiple depth maps. But also geometry super-resolution methods [6, 20] can benefit from our findings. Since we provide evidence for a generative model it is now possible to adapt well established methods from imaging, e.g., Bayesian super-resolution [17], to the geometry reconstruction context.

Acknowledgements. This work was supported in part by the DFG Emmy Noether fellowship GO 1752/3-1.

References

1. Bellocchio, F., Borghese, N.A., Ferrari, S., Piuri, V.: 3D Surface Reconstruction: Multi-Scale Hierarchical Approaches. Springer, New York (2013)
2. Fuhrmann, S., Goesele, M.: Fusion of depth maps with multiple scales. In: SIGGRAPH Asia (2011)
3. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multi-view stereo. In: CVPR (2010)
4. Gargallo, P., Sturm, P.: Bayesian 3D modeling from images using multiple depth maps. In: CVPR (2005)
5. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: ICCV (2007)
6. Goldlücke, B., Cremers, D.: A superresolution framework for high-accuracy multi-view reconstruction. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2009. LNCS, vol. 5748, pp. 342–351. Springer, Heidelberg (2009)
7. Gruen, A., Baltsavias, E.P.: Geometrically constrained multiphoto matching. Photogrammetric Engineering & Remote Sensing 54(5), 633–641 (1988)
8. Habbecke, M., Kobbelt, L.: A surface-growing approach to multi-view stereo reconstruction. In: CVPR (2007)
9. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local stereo matching using geodesic support weights. In: ICIP (2009)
10. Hu, X., Mordohai, P.: A quantitative evaluation of confidence measures for stereo vision. PAMI 34(11), 2121–2133 (2012)
11. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiment. PAMI 16(9), 920–932 (1994)
12. Klowsky, R., Kuijper, A., Goesele, M.: Modulation transfer function of patch-based stereo systems. In: CVPR (2012)
13. Micusik, B., Kosecka, J.: Multi-view superpixel stereo in man-made environments. Tech. rep., Dept. Computer Science, George Mason University (2008)
14. Middlebury multi-view stereo evaluation, `http://vision.middlebury.edu/mview/`
15. Mücke, P., Klowsky, R., Goesele, M.: Surface reconstruction from multi-resolution sample points. In: VMV (2011)
16. Physically based rendering, `http://www.pbrt.org`
17. Pickup, L., Capel, D., Roberts, S., Zisserman, A.: Bayesian methods for image super-resolution. The Computer Journal 52(1), 101–113 (2007)
18. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR (2006)
19. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal sets for efficient structure from motion. In: CVPR (2008)
20. Yang, Q., Yang, R., Davis, J., Nister, D.: Spatial-depth super resolution for range images. In: CVPR (2007)
21. Yoon, K.J., Kweon, I.S.: Locally adaptive support-weight approach for visual correspondence search. In: CVPR (2005)

Optical Flow on Evolving Surfaces with an Application to the Analysis of 4D Microscopy Data

Clemens Kirisits[1], Lukas F. Lang[1], and Otmar Scherzer[1,2]

[1] Computational Science Center, University of Vienna,
Nordbergstr. 15, 1090 Vienna, Austria
{clemens.kirisits,lukas.lang,otmar.scherzer}@univie.ac.at
[2] Radon Institute of Computational and Applied Mathematics,
Austrian Academy of Sciences, Altenberger Str. 69, 4040 Linz, Austria

Abstract. We extend the concept of optical flow to a dynamic non-Euclidean setting. Optical flow is traditionally computed from a sequence of flat images. It is the purpose of this paper to introduce variational motion estimation for images that are defined on an evolving surface. Volumetric microscopy images depicting a live zebrafish embryo serve as both biological motivation and test data.

Keywords: Computer Vision, biomedical imaging, optical flow, variational methods, evolving surfaces, zebrafish, laser-scanning microscopy.

1 Introduction

Advances in laser-scanning microscopy and fluorescent protein technology have increased resolution of microscopy imaging up to a single cell level [11]. They allow for four-dimensional (volumetric time-lapse) imaging of living organisms and shed light on cellular processes during early embryonic development. Understanding cellular development often requires estimation and analysis of cell motion. However, the amount of data captured is tremendous and therefore manual analysis is not an option.

The specific biological motivation for this work is to understand the motion and division behaviour of fluorescently labelled endodermal cells of a zebrafish embryo. The marked cells develop on the surface of the embryo's yolk, where they form a non-contiguous monolayer [17]. Loosely speaking, they only sit next to each other but not on top of each other. Moreover, the yolk deforms over time; see Fig. 1.

We take these biological facts into account and restrict our attention to the analysis of cell motion on the yolk's surface. With this approach it is possible to reduce the amount of data by one space dimension. The resulting problem consists in the estimation of motion of brightness patterns that are restricted to an itself moving surface. We approach this problem by adapting the classical concept of optical flow to the present setting, where the image domain is

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 246–257, 2013.

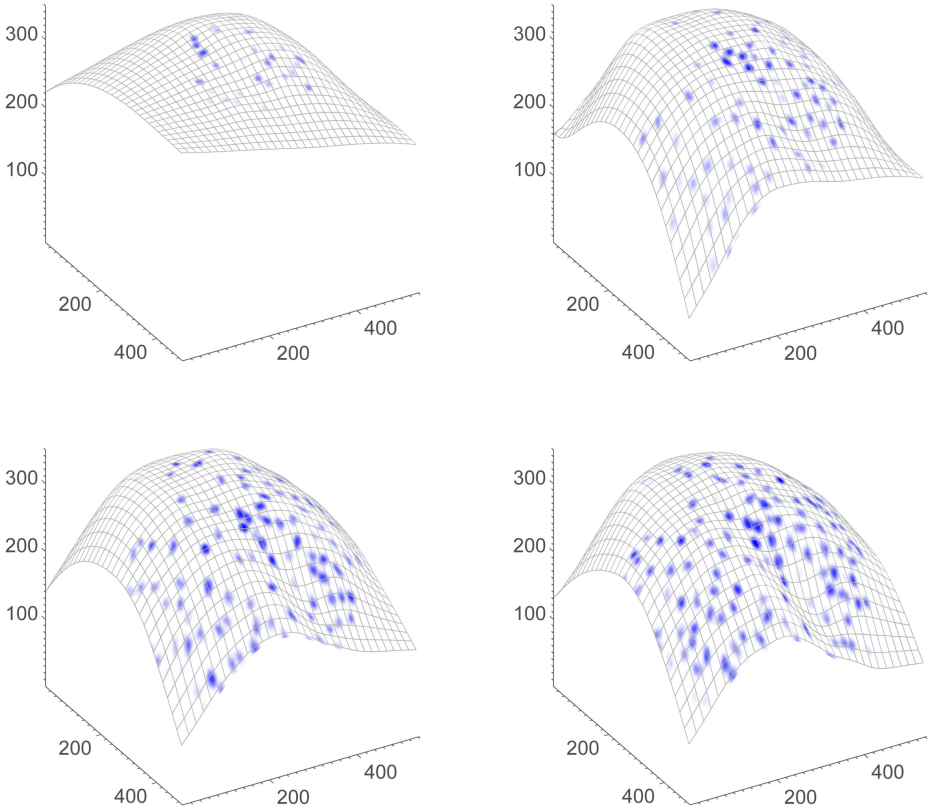


Fig. 1. Sequence of embryonic zebrafish images. The curved mesh represents a section of the yolk's surface. Depicted are frames no. 30, 45, 55, and 60 of the entire sequence. All dimensions are in micrometer (μm). See Sec. 4.1 for more details on the microscopy data.

both non-Euclidean and dynamic. Note that due to the monolayer structure cell occlusions cannot occur. This makes the optical flow field a more reliable approximation to the true motion field.

Our contributions in the field of optical flow are as follows. First, we formulate the optical flow problem on an evolving two-dimensional manifold and give two equivalent ways of linearising the brightness constancy assumption (Secs. 2.1 and 2.2). One uses a parametrisation of the evolving surface, the other one is parameter-independent. Second, we use a generalisation of the Horn-Schunck model to regularise the optical flow field (Sec. 2.3). For a given global parametrisation of the evolving surface, we solve the associated Euler-Lagrange equations in the parameter domain with a finite difference scheme (Sec. 3). Finally, we

apply this technique to obtain qualitative results from the afore-mentioned zebrafish data (Sec. 4). Our experiments show that the optical flow is an appropriate tool for analysing these data. It is capable of estimating global trends as well as individual cell movements and, in particular, it is able to indicate cell division events.

Related Work. Optical flow is the apparent motion in a sequence of images. Its estimation is a key problem in Computer Vision. Horn and Schunck [5] were the first to propose a variational approach assuming constant brightness of moving points and spatial smoothness of the velocity field. Since then, a vast number of modifications has been developed. See [1] for a recent survey.

Miura [13] observed that until 2005 optical flow has been mostly disregarded as a method for motion extraction in cell biological data. Since then, a few articles have explored this direction: Melani et al. [12] and Hubený et al. [6] extended variational optical flow methods to volumetric images to obtain 3D displacement fields. In the former article, the resulting algorithm is also applied to zebrafish microscopy data. Quelhas et al. [15] use optical flow to detect cell divisions in a live plant root. However, they work with 2D (plus time) data only. Therefore, their approach suffers from errors caused by 3D off-plane motion.

Clearly, certain natural scenarios are more accurately described by a velocity field on a non-flat surface rather than on a flat domain. With applications to robot vision, Imiya et al. [7,16] considered optical flow for spherical images. In a more general setting, Lefèvre and Baillet [10] extended the Horn-Schunck method to 2-Riemannian manifolds and showed well-posedness. They solve the numerical problem with finite elements on a surface triangulation. In all of the above works the underlying imaging surface is fixed over time, while in this paper it is not.

2 Optical Flow on Evolving Surfaces

2.1 Brightness Constancy

Let $\mathcal{M}_t \subset \mathbb{R}^3$, $t \in I = [0, T)$, be a compact smooth two-dimensional manifold evolving smoothly over time. We assume the velocity to be unknown. Moreover, denote by $\tilde{f}$ a scalar time-dependent quantity defined on the surface

$$\tilde{f} \colon \bigcup_{t \in I} (\mathcal{M}_t \times \{t\}) \to \mathbb{R}.$$

We begin with a Lagrangian specification of the optical flow field. That is, for every starting point $\boldsymbol{x}_0 \in \mathcal{M}_0$ we seek a trajectory where the data $\tilde{f}$ are conserved. More precisely, we want to find a function

$$\gamma \colon \mathcal{M}_0 \times I \to \bigcup_{t \in I} \mathcal{M}_t,$$

such that

1. $\gamma(\boldsymbol{x}_0, t) \in \mathcal{M}_t$ for all $t \in I$, for all $\boldsymbol{x}_0 \in \mathcal{M}_0$,
2. $\gamma(\cdot, t)$ is a diffeomorphism between $\mathcal{M}_0$ and $\mathcal{M}_t$ for all $t \in I$,
3. $\gamma(\cdot, 0) = \mathrm{Id}_{\mathcal{M}_0}$,

is fulfilled and which satisfies a "brightness" constancy assumption (BCA)

$$\tilde{f}(\boldsymbol{x}_0, 0) = \tilde{f}(\gamma(\boldsymbol{x}_0, t), t), \quad \text{for all } (\boldsymbol{x}_0, t) \in \mathcal{M}_0 \times I. \tag{1}$$

In classical optical flow computations it is common practice to linearise the BCA by taking its time derivative and to solve the resulting equation for the Eulerian unknown $\dot{\gamma}$.[1] We also take this route, but differentiation of $\tilde{f}$ is more involved. Observe, for example, that for an arbitrary $t_0 \in I$ and $\boldsymbol{x} \in \mathcal{M}_{t_0}$ the usual partial derivative

$$\partial_t \tilde{f}(\boldsymbol{x}, t_0) = \lim_{h \to 0} \frac{1}{h} \left(\tilde{f}(\boldsymbol{x}, t_0 + h) - \tilde{f}(\boldsymbol{x}, t_0) \right)$$

is not well-defined, simply because, in general, $\boldsymbol{x}$ is not an element of $\mathcal{M}_{t_0+h}$ for all $h \neq 0$.

In the next section we linearise (1) in two different ways. First, we use a global parametrisation to pull the data back to a fixed reference domain and linearise afterwards. In our second approach we borrow some notions from continuum mechanics [2] to directly linearise (1).

2.2 Linearisation

Linearisation after Pull-Back. Let $\Omega \in \mathbb{R}^2$ be a compact domain and

$$\boldsymbol{x} \colon \Omega \times I \to \mathbb{R}^3, \quad (x_1, x_2, t) = (x, t) \mapsto \boldsymbol{x}(x, t) \in \mathcal{M}_t$$

be a parametrisation of the evolving surface. Denote by f the coordinate representation of $\tilde{f}$, that is,

$$f(x, t) = \tilde{f}(\boldsymbol{x}(x, t), t) \tag{2}$$

and let

$$\beta \colon \Omega \times I \to \Omega$$

be the coordinate counterpart of γ. This means, if we let $\boldsymbol{x}_0 = \boldsymbol{x}(x_0, 0)$, then $\beta(x_0, t)$ gives the coordinates of $\gamma(\boldsymbol{x}_0, t) \in \mathcal{M}_t$ in Ω (see Fig. 2). In other words, we have the identity

$$\gamma(\boldsymbol{x}(x_0, 0), t) = \boldsymbol{x}(\beta(x_0, t), t), \quad \text{for all } (x_0, t) \in \Omega \times I. \tag{3}$$

Now, from (1), (2) and (3) we get

$$\begin{aligned} f(x_0, 0) &= \tilde{f}(\boldsymbol{x}_0, 0) \\ &= \tilde{f}(\gamma(\boldsymbol{x}_0, t), t) \\ &= \tilde{f}(\boldsymbol{x}(\beta(x_0, t), t), t) \\ &= f(\beta(x_0, t), t), \end{aligned}$$

[1] To simplify expressions we use Newton's notation for those time derivatives that correspond to actual velocities, for example $\dot{\gamma} = \partial_t \gamma$.

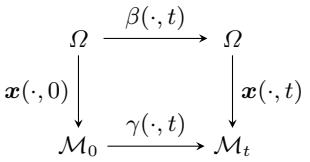


Fig. 2. Commutative diagram describing the relation between unknowns β and γ.

which is a coordinate version of the BCA. After differentiation with respect to t it becomes

$$\nabla^2 f \cdot \dot{\beta} + \partial_t f = 0, \tag{4}$$

where $\nabla^2 = (\partial_1, \partial_2)^\top$ is the two-dimensional spatial gradient. Note that the last equation is nothing but the classical optical flow constraint (OFC) for Euclidean data f and a displacement field $\dot{\beta}$.

Direct Linearisation. We turn to our second derivation. While, as pointed out above, the partial derivative $\partial_t \tilde{f}$ is undefined in general, it does make sense to differentiate $\tilde{f}$ following the surface movement. Let $\boldsymbol{y}$ be a point on $\mathcal{M}_{t_0}$ and $\xi\colon t \mapsto \xi(t) \in \mathcal{M}_t$ an arbitrary smooth trajectory through the evolving surface satisfying $\xi(t_0) = \boldsymbol{y}$. Now we can compute

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{f}(\xi(t),t)\bigg|_{t=t_0} = \lim_{h\to 0}\frac{1}{h}\left(\tilde{f}(\xi(t_0+h),t_0+h) - \tilde{f}(\boldsymbol{y},t_0)\right)$$

to obtain a valid derivative of $\tilde{f}$. Since this time derivative only depends on the vector $\boldsymbol{v} = \dot{\xi}(t_0)$, we denote it by $\mathrm{d}_t^{\boldsymbol{v}}\tilde{f}$. A natural candidate for a trajectory along which to differentiate is given by the parametrisation $\xi(t) = \boldsymbol{x}(x,t)$. Another possible choice would be a trajectory that is normal to $\mathcal{M}_{t_0}$. The resulting normal time derivative is accordingly denoted by $\mathrm{d}_t^{\boldsymbol{n}}\tilde{f}$.

Finally, we also need the surface gradient $\nabla_{\mathcal{M}}\tilde{f}$. If F is a smooth extension of $\tilde{f}$ to an open neighbourhood of $\boldsymbol{y} \in \mathcal{M}_{t_0}$ in $\mathbb{R}^3$, then the surface gradient of F at $\boldsymbol{y}$ is defined as the projection of the three-dimensional spatial gradient $\nabla^3 F$ onto the tangent plane to $\mathcal{M}_{t_0}$

$$\nabla_{\mathcal{M}} F = \nabla^3 F - (\nabla^3 F \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}},$$

where $\hat{\boldsymbol{n}}$ is the unit normal to $\mathcal{M}_{t_0}$. The surface gradient only depends on the values of F on the surface; see e.g. [4, p. 389]. Thus, $\nabla_{\mathcal{M}}\tilde{f} = \nabla_{\mathcal{M}} F$ is well-defined.

The spatial and temporal derivatives of $\tilde{f}$ introduced above are related in a simple way. As shown in [2], they satisfy the equality

$$\begin{aligned} \mathrm{d}_t^{\dot{\boldsymbol{x}}}\tilde{f} &= \nabla_{\mathcal{M}}\tilde{f}\cdot\dot{\boldsymbol{x}} + \mathrm{d}_t^{\boldsymbol{n}}\tilde{f} \\ &= \nabla_{\mathcal{M}}\tilde{f}\cdot\dot{\boldsymbol{x}}_{\tan} + \mathrm{d}_t^{\boldsymbol{n}}\tilde{f}, \end{aligned} \tag{5}$$

where $\dot{\boldsymbol{x}}_{\tan}$ is the tangential surface velocity, that is, the projection of $\dot{\boldsymbol{x}}$ onto the tangent plane to $\mathcal{M}_{t_0}$. This decomposition of $\mathrm{d}_t^{\dot{\boldsymbol{x}}}\tilde{f}$ into normal and tangential components is clearly valid for any trajectory in place of $\boldsymbol{x}$, and therefore in particular for the unknown γ. This means we can use (5) in order to differentiate the BCA (1) with respect to t. The resulting OFC reads

$$\nabla_{\mathcal{M}}\tilde{f}\cdot\dot{\gamma}_{\tan}+\mathrm{d}_t^{\boldsymbol{n}}\tilde{f}=0. \tag{6}$$

Discussion. We conclude this section with a brief comparison of the two OFCs derived above. We start by showing how to obtain (4) from (6) and vice versa. To this end we again assume the existence of a global parametrisation and rewrite all quantities in (6) in terms of $\boldsymbol{x}$. First observe that, by (3), the velocity of γ equals the surface velocity $\dot{\boldsymbol{x}}$ plus a purely tangential component

$$\dot{\gamma}=\dot{\boldsymbol{x}}+J\dot{\beta},$$

where $J=(\partial_1\boldsymbol{x}\ \partial_2\boldsymbol{x})$ is the Jacobian matrix of $\boldsymbol{x}$ with respect to x. On the other hand, by (5), the normal time derivative is equal to the time derivative of $\tilde{f}$ following $\boldsymbol{x}$ minus its tangential component

$$\mathrm{d}_t^{\boldsymbol{n}}\tilde{f}=\mathrm{d}_t^{\dot{\boldsymbol{x}}}\tilde{f}-\nabla_{\mathcal{M}}\tilde{f}\cdot\dot{\boldsymbol{x}}.$$

Using the last two equations to rewrite the left-hand side of (6) yields

$$\begin{aligned}\nabla_{\mathcal{M}}\tilde{f}\cdot\dot{\gamma}+\mathrm{d}_t^{\boldsymbol{n}}\tilde{f}&=\nabla_{\mathcal{M}}\tilde{f}\cdot\left(\dot{\boldsymbol{x}}+J\dot{\beta}\right)+\mathrm{d}_t^{\dot{\boldsymbol{x}}}\tilde{f}-\nabla_{\mathcal{M}}\tilde{f}\cdot\dot{\boldsymbol{x}}\\&=\nabla_{\mathcal{M}}\tilde{f}\cdot J\dot{\beta}+\mathrm{d}_t^{\dot{\boldsymbol{x}}}\tilde{f},\end{aligned}$$

which is already the left-hand side of (4) in terms of $\tilde{f}$. It only remains to observe that $\mathrm{d}_t^{\dot{\boldsymbol{x}}}\tilde{f}=\partial_t f$ and to replace the surface gradient $\nabla_{\mathcal{M}}\tilde{f}$ by its coordinate expression $Jg^{-1}\nabla^2 f$, where $g=J^\top J$ is the coefficient matrix of the Riemannian metric; see e.g. [9].

We highlight the qualitative difference between the constraints (4) and (6). Note that in the former the unknown is $\dot{\beta}$, while in the latter it is $\dot{\gamma}_{\tan}=\dot{\boldsymbol{x}}_{\tan}+J\dot{\beta}$. This means that (4) constrains the motion relative to the tangential surface velocity $\dot{\boldsymbol{x}}_{\tan}$, while (6) constrains the absolute tangential motion.

The nature of our microscopy data suggests a simple global parametrisation (see Sec. 3). We therefore pull the data back to the Euclidean plane and solve (4). However, equation (6) is independent of any parametrisation. It can thus serve as a starting point for alternative numerical approaches.

2.3 Regularisation

From now on we fix an arbitrary $t_0\in I$ and turn to the actual solution of the parametrised OFC for $(u^1(x),u^2(x))^\top=u(x)=\dot{\beta}(x,t_0)$. Recall that with this notation u contains the coefficients of the tangential vector field $\boldsymbol{u}=J\dot{\beta}$ with respect to the tangential basis $(\partial_1\boldsymbol{x},\partial_2\boldsymbol{x})$ of $\mathcal{M}_{t_0}$. Note also that, by fixing t_0,

there is no more time-dependence in our problem which makes it effectively an optical flow problem on a static surface. Hence we omit any reference to t_0 from now on and write $\mathcal{M}$ instead of $\mathcal{M}_{t_0}$.

The sought vector field is underdetermined by the OFC alone. We overcome this by minimising a functional that penalises violation of the OFC while imposing an additional smoothness restriction on $\boldsymbol{u}$. More precisely, we adopt a recent extension of the original quadratic Horn-Schunck regularisation to a Riemannian setting [10]. Basically, they propose to minimise

$$\mathcal{E}(u) = \frac{\alpha}{2}\big\|\nabla^2 f \cdot u + \partial_t f\big\|^2_{L^2(\mathcal{M})} + \frac{1}{2}\big\|Du\big\|^2_{L^2(\mathcal{M})}. \tag{7}$$

Here, $\alpha > 0$ is the regularisation parameter and $Du = (D_j u^i)$ is the 2×2 matrix containing the coefficient functions of the covariant derivatives

$$\nabla_j \boldsymbol{u} = \sum_{i=1}^{2} D_j u^i \partial_i \boldsymbol{x}, \;\; j = 1, 2,$$

of $\boldsymbol{u}$. Using the Christoffel symbols Γ^i_{jk} (see Sec. 3) associated to the parametrisation $\boldsymbol{x}$ the coefficients are given by

$$D_j u^i = \partial_j u^i + \sum_{k=1}^{2} \Gamma^i_{jk} u^k, \;\; i, j = 1, 2.$$

Rewriting (7) as an integral over the coordinate domain, we arrive at the functional

$$\mathcal{E}(u) = \frac{1}{2}\int_\Omega \Big[\alpha\left(\nabla^2 f \cdot u + \partial_t f\right)^2 + \|Du\|_F^2\Big]\sqrt{\det g}\,\mathrm{d}x, \tag{8}$$

where $\|\cdot\|_F$ is the Frobenius norm.

3 Numerical Solution

We solve the problem of minimising functional $\mathcal{E}$ via its associated Euler-Lagrange equations. Regarding the integrand of $\mathcal{E}$ as a function $G(x, u, \nabla^2 u^1, \nabla^2 u^2)$, they read

$$\begin{aligned} G_{u^1} &= \partial_1 G_{\partial_1 u^1} + \partial_2 G_{\partial_2 u^1} \\ G_{u^2} &= \partial_1 G_{\partial_1 u^2} + \partial_2 G_{\partial_2 u^2}, \end{aligned}$$

where subscripts of G denote partial derivatives. The resulting pair of linear PDEs is of the form

$$\begin{aligned} \Delta u^1 &= \nabla^2 u^1 \cdot c + \nabla^2 u^2 \cdot d + u \cdot b_1 + a_1 \\ \Delta u^2 &= \nabla^2 u^2 \cdot c + \nabla^2 u^1 \cdot d + u \cdot b_2 + a_2. \end{aligned} \tag{9}$$

The coefficient vectors a, b_1, b_2, c, d are rather lengthy functions of the data f and metric tensor g, which is why we do not write them out in full here. Letting

$\Omega = (0,1)^2$ for simplicity, the natural boundary conditions of the variational problem are

$$\partial_j u^i + \sum_k \Gamma^i_{jk} u^k = 0, \quad \text{for } x_j \in \{0,1\}, \tag{10}$$

where $i,j \in \{1,2\}$. In case of a flat manifold, e.g. $\mathcal{M} = \Omega$, the Euler-Lagrange equations (9) reduce to those of the original Horn-Schunck functional and the boundary conditions become the usual homogeneous Neumann ones. For more details on the calculus of variations we refer to [3].

Due to the nature of the microscopy data (see Sec. 4.1 and Fig. 1), the manifold $\mathcal{M}_t$ modelling the deforming yolk is a surface with boundary that is most easily parametrised as the graph of a function $z : \Omega \times I \to \mathbb{R}$. Hence, we set $\boldsymbol{x}(x_1, x_2, t) = (x_1, x_2, z(x_1, x_2, t))^\top$. Accordingly, for the metric we get

$$g = I_2 + \nabla^2 z \nabla^2 z^\top, \qquad \det g = 1 + |\nabla^2 z|^2,$$

where $I_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix. The Christoffel symbols turn out to be

$$\Gamma^i_{jk} = \frac{1}{2} \sum_{m=1}^{2} g^{mi} \left(\partial_j g_{km} + \partial_k g_{mj} - \partial_m g_{jk}\right) = \frac{\partial_i z \, \partial_{jk} z}{\det g}.$$

Partial derivatives of z and of the projected data f were approximated by central differences. The system (9) with boundary conditions (10) was then solved with a standard finite difference scheme. In the following section numerical results are presented.

4 Experiments

4.1 Data

As mentioned before, the biological motivation for this work are cellular image data of a zebrafish embryo. Endoderm cells expressing green fluorescent protein were recorded via confocal laser-scanning microscopy resulting in time-lapse volumetric (4D) images; see [11] for the imaging techniques. This type of image shows a high contrast at cell boundaries and a low signal-to-noise ratio in general. Our videos were obtained during the gastrula period, which is an early stage in the animal's developmental process and takes place approximately five to ten hours post fertilisation. In short, the fish forms on the surface of a spherical-shaped yolk; see e.g. [8] for many illustrations and detailed explanations. For the biological methods such as the fluorescence marker and the embryos used in this work we refer to [14]. The important aspect about endodermal cells is that they are known to form a monolayer during gastrulation [17], meaning that the radial extent is only a single cell. This crucial fact allows for the straightforward extraction of a surface together with a two-dimensional image of the stained cells. Since only a cuboid region of approximately $860 \times 860 \times 340\,\mu\text{m}^3$ of the pole region is captured by the microscope, this surface can easily be parametrised; cf. Sec. 3.

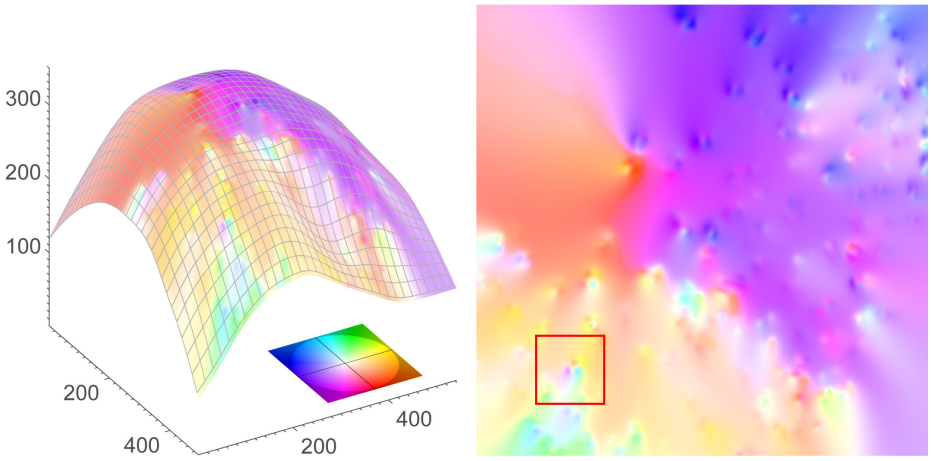


Fig. 3. Optical flow field between frames 57 and 58 of the sequence. Colours indicate direction whereas darkness of a colour indicates the length of the vector. Note that the colour circle has been enlarged for better visibility.

The spatial resolution of the Gaussian filtered images is 512×512 pixels and all intensities are given in the interval $[0, 1]$. Our sequence contains 77 frames recorded in intervals of 240 s with clearly visible cellular movements and cell divisions.

4.2 Numerical Results

In the following we present qualitative results and demonstrate the feasibility of our approach. For every subsequent pair of frames we minimised the functional (8) as outlined in Sec. 3. We chose grid size as well as temporal displacement as $h = 1$ and the regularisation parameter was set to $\alpha = 10$. For demonstration purpose we make use of the standard flow colour-coding [1], which maps (normalised) flow vectors to a colour space defined inside the unit circle. It is easy to see that the same colours are valid all over the manifold due to the parametrisation.

As representative candidates for this discussion we chose the displacement field between frames 57 and 58 for the following reasons. First, the surface is distinctly developed. Second, a considerable number of cells is present in the image, and third, the interval contains cell divisions. Figure 3, left, shows the colour-coded tangential vector field and the colour space whereas Fig. 3, right, displays the same motion field as computed in the parameter space.[2] A visual inspection of the dataset shows that cells tend to move towards the embryo's body axis, which roughly runs along the main diagonal in Fig. 3, right. Clearly, the velocity field is sufficiently smooth and suggests this behaviour in an adequate manner on a large

[2] Some figures may appear in colour only in the online version.

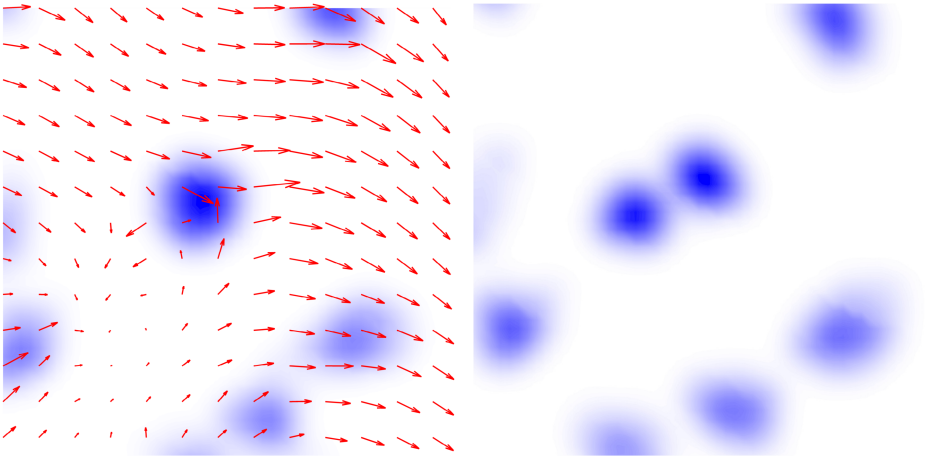


Fig. 4. Detailed view of a cell division occurring between frames 57 (left) and 58 (right). All vectors are scaled and only every fourth vector is shown. Intensities are interpolated for smooth illustration.

scale. The expected change in orientation along the body axis is well represented by the colour shift from orange-yellow below the main diagonal to purplish blue in the region above. On the contrary, the choice of the regularisation parameter ensures that individual movements are well preserved as can be observed from the image.

Figure 4 gives a detailed view of the section outlined by a (red) rectangle in Fig. 3, right. This section was chosen because it depicts a cell division. Figure 4, left, and Fig. 4, right, display the frames before and after the event, respectively. Moreover, in Fig. 4, left, the velocity field is shown. From the raw data we observed that when a cell actually splits, the two daughter cells drift apart in a 180 ° angle with respect to the mother cell. The displacement field clearly shows the anticipated pattern caused by the diverging daughter cells. In Fig. 3, right, the event is point up by two areas which are coloured mutually opposite with respect to the colour space. Our results suggest that cell division can be indicated reasonably well by our model. Both implementation and data are available on our website.[3]

5 Conclusion

Aiming at efficient motion analysis of 4D cellular microscopy data, we generalised the Horn-Schunck method to videos defined on evolving surfaces. The biological fact that the observed cells move along an itself deforming surface allows for motion estimation in 2D (plus time). In the course of this work, we presented two ways to linearise the brightness constancy assumption and showed that one could

[3] `http://www.csc.univie.ac.at`

be obtained from the other and vice versa. The resulting optical flow constraint was solved by means of quadratic regularisation and verified on the basis of the afore-mentioned data. Our qualitative results suggest that both global trends as well as individual movements including cell division are well shown in the surface velocity field. However, so far we only laid the basic groundwork in terms of a mathematical model.

Acknowledgements. We thank Pia Aanstad from the University of Innsbruck for sharing her biological insight and for kindly providing the microscopy data. This work has been supported by the Vienna Graduate School in Computational Science (IK I059-N) funded by the University of Vienna. In addition, we acknowledge the support by the Austrian Science Fund (FWF) within the national research networks "Photoacoustic Imaging in Biology and Medicine" (project S10505-N20, Reconstruction Algorithms for PAI) and "Geometry + Simulation" (project S11704, Variational Methods for Imaging on Manifolds).

References

1. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A Database and Evaluation Methodology for Optical Flow. Int. J. Comput. Vision 92(1), 1–31 (2011)
2. Cermelli, P., Fried, E., Gurtin, M.E.: Transport relations for surface integrals arising in the formulation of balance laws for evolving fluid interfaces. J. Fluid Mech. 544, 339–351 (2005)
3. Courant, R., Hilbert, D.: Methods of mathematical physics, vol. I. Interscience Publishers, Inc., New York (1953)
4. Gilbarg, D., Trudinger, N.: Elliptic Partial Differential Equations of Second Order. Classics in Mathematics. Springer, Berlin (2001), Reprint of the 1998 edition
5. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
6. Hubený, J., Ulman, V., Matula, P.: Estimating large local motion in live-cell imaging using variational optical flow. In: VISAPP: Proc. of the Second International Conference on Computer Vision Theory and Applications, pp. 542–548. INSTICC (2007)
7. Imiya, A., Sugaya, H., Torii, A., Mochizuki, Y.: Variational analysis of spherical images. In: Gagalowicz, A., Philips, W. (eds.) CAIP 2005. LNCS, vol. 3691, pp. 104–111. Springer, Heidelberg (2005)
8. Kimmel, C.B., Ballard, W.W., Kimmel, S.R., Ullmann, B., Schilling, T.F.: Stages of embryonic development of the zebrafish. Devel. Dyn. 203(3), 253–310 (1995)
9. Lee, J.M.: Riemannian Manifolds. An Introduction to Curvature. Graduate Texts in Mathematics, vol. 176. Springer, New York (1997)
10. Lefèvre, J., Baillet, S.: Optical flow and advection on 2-Riemannian manifolds: A common framework. IEEE Trans. Pattern Anal. Mach. Intell. 30(6), 1081–1092 (2008)
11. Megason, S.G., Fraser, S.E.: Digitizing life at the level of the cell: high-performance laser-scanning microscopy and image analysis for in toto imaging of development. Mech. Dev. 120(11), 1407–1420 (2003)

12. Melani, C., Campana, M., Lombardot, B., Rizzi, B., Veronesi, F., Zanella, C., Bourgine, P., Mikula, K., Peyriéras, N., Sarti, A.: Cells tracking in a live zebrafish embryo. In: Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007), pp. 1631–1634 (2007)
13. Miura, K.: Tracking Movement in Cell Biology. In: Rietdorf, J. (ed.) Microscopy Techniques. Advances in Biochemical Engineering/Biotechnology, vol. 95, pp. 267–295. Springer (2005)
14. Mizoguchi, T., Verkade, H., Heath, J.K., Kuroiwa, A., Kikuchi, Y.: Sdf1/Cxcr4 signaling controls the dorsal migration of endodermal cells during zebrafish gastrulation. Development 135(15), 2521–2529 (2008)
15. Quelhas, P., Mendonça, A.M., Campilho, A.: Optical flow based arabidopsis thaliana root meristem cell division detection. In: Campilho, A., Kamel, M. (eds.) ICIAR 2010, Part II. LNCS, vol. 6112, pp. 217–226. Springer, Heidelberg (2010)
16. Torii, A., Imiya, A., Sugaya, H., Mochizuki, Y.: Optical Flow Computation for Compound Eyes: Variational Analysis of Omni-Directional Views. In: De Gregorio, M., Di Maio, V., Frucci, M., Musio, C. (eds.) BVAI 2005. LNCS, vol. 3704, pp. 527–536. Springer, Heidelberg (2005)
17. Warga, R.M., Nüsslein-Volhard, C.: Origin and development of the zebrafish endoderm. Development 126(4), 827–838 (1999)

Perspective Photometric Stereo with Shadows*

Roberto Mecca, Guy Rosman, Ron Kimmel, and Alfred M. Bruckstein

Technion - Israel Institute of Technology,
Department of Computer Science

Abstract. High resolution reconstruction of 3D surfaces from images remains an active area of research since most of the methods in use are based on practical assumptions that limit their applicability. Furthermore, an additional complication in all active illumination 3D reconstruction methods is the presence of shadows, whose presence cause loss of information in the image data. We present an approach for the reconstruction of surfaces via Photometric Stereo, based on the perspective formulation of the Shape from Shading problem, solved via partial differential equations. Unlike many photometric stereo solvers that use computationally costly variational methods or a two-step approach, we use a novel, well-posed, differential formulation of the problem that enables us to solve a first order partial differential equation directly via an alternating directions raster scanning scheme. The resulting formulation enables surface computation for very large images and allows reconstruction in the presence of shadows.

Keywords: Photometric Stereo, Perspective Shape from Shading, Shadows, up-wind scheme, semi-Lagrangian scheme.

1 Introduction

The classical computer vision topic of Shape from Shading (SfS) was recently revitalized by a series of research contributions driven in part by some interesting new applications [1–3]. The technique based on the shape recovery from several pictures of the same scene taken under different illuminations, namely Photometric Stereo (PS), has gained some popularity, due to the feasibility of implementing controlled light systems. In this context, quite a few multi-image depth recovery techniques based on inverting shading models have been addressed in the literature [4, 5]. Utilizing multiple images in order to remove both the nonlinearities in the image irradiance equation and the generally unknown albedo, new ideas have been introduced in order to solve the PS problems more efficiently, see [6–9].

Most of the works which addressed the PS problem, for example [1, 5, 10, 11], reconstruct the surface in two steps:

* This research was partly supported by European Community's FP7- ERC program, grant agreement no. 267414 and by Broadcom foundation.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 258–269, 2013.

1. the estimation of the gradient of the surface (usually via some local minimization algorithms);
2. the recovery of the height from the gradient field all over the domain (by integration or by functional minimization).

In the framework of classical PDEs for a single input image and known albedo there exists a well known direct approach to SfS which uses level sets [12]. Its drawback, among others, is the need to know a-priori the albedo, which limits the scope of applications where this method can be employed. Here, we present a new model for a direct recovery of the surface considering Perspective Photometric Stereo with n images (PPS_n) with shadows. In Section 2 we recall the differential formulation for the PPS_2 introduced in [9] with only two images. Section 3 contains the construction of the proposed differential problem taking into account multiple images containing shadows. Note that our hypotheses are weaker than the ones assumed in [4] which addressed the same problem without the perspective transformation and considered a two step procedure with regularization terms for smooth surfaces. We will focus here on a surface recovery based on the direct computation of the unique weak (Lipschitz) solution of a linear PDEs.

The theoretical formulation of the new differential approach can be easily extended when more than three images are considered. The mathematical proof of the existence and uniqueness of a weak solution for this new formulation is sketched in Section 4. The numerical schemes are presented in Section 5 where both up-wind and semi-Lagrangian methods are implemented considering the *Fast Sweeping* technique. In Section 6 we show some numerical tests in order to demonstrate the order of consistency of the numerical schemes and the fast reconstruction of the surface respectively. In particular, in these tests we consider images of several megapixels with a significant portion of shadow areas. Section 7 concludes the paper.

2 Perspective Photometric Stereo Technique

In this section we briefly recall the model for the PSfS, and the direct solution method described for this case, as presented in [9]. Let us define the observed surface as $h(x, y) = (x, y, \hat{z}(x, y))$. We define a far light source by its unit vector ω. The associated reflectance equation is given by the Lambertian illumination model [13]:

$$I = \rho(\omega \cdot n), \quad \omega = (\omega_1, \omega_2, \omega_3), \quad \omega_3 < 0 \tag{1}$$

where ρ is the unknown albedo function, I is the image and n is the incoming unit normal to the surface. There are several ways to describe the perspective transformation of the surface [1, 2, 14]. Here we consider the one introduced in [15], based on the following transformation

$$k(\xi, \eta) = (\xi, \eta, z(\xi, \eta)) = \Big(-\frac{x}{\hat{z}(x, y)} f, -\frac{y}{\hat{z}(x, y)} f, \hat{z}(x, y) \Big) \tag{2}$$

where $\hat{z}(x,y) = z(\xi,\eta)$ and the positive quantity f is the focal length and the point (ξ,η) belongs to the perspective domain in the focal plane (in blue in Fig. 1), namely $\overline{\Omega}^p = \Omega^p \cup \partial\Omega^p$.

The differential formulation for the PSfS problem

$$\rho(\xi,\eta)\frac{-fz_\xi\omega_1 - fz_\eta\omega_2 - (z+\xi z_\xi+\eta z_\eta)\omega_3}{\sqrt{f^2(z_\xi^2+z_\eta^2)+(z+\xi z_\xi+\eta z_\eta)^2}} = I(\xi,\eta), \quad \text{for} \quad (\xi,\eta)\in\Omega^p \tag{3}$$

is not well-posed even if the Dirichlet boundary condition, i.e. $z(\xi,\eta) = g(\xi,\eta)$ is given [15].

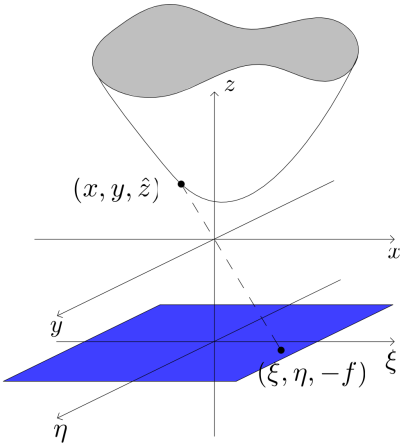


Fig. 1. Schematic representation of a surface taken under perspective view. The point of the real surface $(x,y,\hat{z})$ is projected in the perspective domain in the point (ξ,η) of the focal plane (in blue), parallel to the optical one (xy-plane) at a focal distance f.

In [9] the classical PS technique has been modified to a well posed formulation of the PPS_2 problem involving surface recovery via a direct differential approach. Two light sources given by ω' and ω'' are considered resulting in the following non-linear system of PDEs,

$$\begin{cases} \rho(\xi,\eta)\dfrac{-z_\xi(f\omega_1'+\xi\omega_3')-z_\eta(f\omega_2'+\eta\omega_3')-z\omega_3'}{\sqrt{f^2(z_\xi^2+z_\eta^2)+(z+\xi z_\xi+\eta z_\eta)^2}} = I_1(\xi,\eta), & \text{on } \Omega^p \\ \rho(\xi,\eta)\dfrac{-z_\xi(f\omega_1''+\xi\omega_3'')-z_\eta(f\omega_2''+\eta\omega_3'')-z\omega_3''}{\sqrt{f^2(z_\xi^2+z_\eta^2)+(z+\xi z_\xi+\eta z_\eta)^2}} = I_2(\xi,\eta), & \text{on } \Omega^p \\ z(\xi,\eta) = g(\xi,\eta) & \text{on } \partial\Omega^p. \end{cases} \tag{4}$$

Simplifying the common quantity $\frac{\rho(\xi,\eta)}{\sqrt{f^2(z_\xi^2+z_\eta^2)+(z+\xi z_\xi+\eta z_\eta)^2}}$, the system can be written as the following well-posed problem,

$$\begin{cases} b(\xi,\eta)\cdot\nabla z(\xi,\eta) + s(\xi,\eta)z(\xi,\eta) = 0, & \text{on } \Omega^p \\ z(\xi,\eta) = g(\xi,\eta) & \text{on } \partial\Omega^p \end{cases} \tag{5}$$

where

$$b(\xi,\eta) = \begin{pmatrix} (f\omega_1' + \xi\omega_3')I_2(\xi,\eta) - (f\omega_1'' + \xi\omega_3'')I_1(\xi,\eta) \\ (f\omega_2' + \eta\omega_3')I_2(\xi,\eta) - (f\omega_2'' + \eta\omega_3'')I_1(\xi,\eta)) \end{pmatrix} \tag{6}$$

and

$$s(\xi,\eta) = \omega_3' I_2(\xi,\eta) - \omega_3'' I_1(\xi,\eta). \tag{7}$$

We recall that (5) admits only one weak (i.e. Lipschitz) solution, under the main assumptions that are the absence of shadows and the knowledge of the Dirichlet boundary condition $g(\xi,\eta)$. To overcome these limitations in real applications we consider the PSfS problem with more than two images as the basic model. In this case, an additional potential advantage beyond computational efficiency is the possible noise robustness of the direct method. Furthermore, we exploit the extra image data not only to address noise – but rather to allow reconstruction in the presence of shadows. In the numerical tests we shall demonstrate that significant portions of the image can be missing (see the black patches in Fig. 3) without impeding the correct surface recovery.

3 Direct Surface Reconstruction Using Multiple Images and Shadows

We now generalize the model shown in Section 2 for more than two images. We start with 3 images, which is the minimal number of images that allow a computation of the boundary condition without a-priori knowing the albedo [11]. We start by writing the PDEs (5) resulting from each image pair

$$\begin{cases} b^{(1,2)}(\xi,\eta)\cdot\nabla z(\xi,\eta) + s^{(1,2)}(\xi,\eta)z(\xi,\eta) = 0, & \text{a.e. } (\xi,\eta)\in\Omega^p \\ b^{(1,3)}(\xi,\eta)\cdot\nabla z(\xi,\eta) + s^{(1,3)}(\xi,\eta)z(\xi,\eta) = 0, & \text{a.e. } (\xi,\eta)\in\Omega^p \\ b^{(2,3)}(\xi,\eta)\cdot\nabla z(\xi,\eta) + s^{(2,3)}(\xi,\eta)z(\xi,\eta) = 0, & \text{a.e. } (\xi,\eta)\in\Omega^p \end{cases} \tag{8}$$

where

$$b^{(h,k)}(\xi,\eta) = \begin{pmatrix} (f\omega_1^h + \xi\omega_3^h)I_k(\xi,\eta) - (f\omega_1^k + \xi\omega_3^k)I_h(\xi,\eta) \\ (f\omega_2^h + \eta\omega_3^h)I_k(\xi,\eta) - (f\omega_2^k + \eta\omega_3^k)I_h(\xi,\eta) \end{pmatrix} \tag{9}$$

and

$$s(\xi,\eta)^{(h,k)} = I_k(\xi,\eta)\omega_3^h - I_h(\xi,\eta)\omega_3^k. \tag{10}$$

A similar formulation has been given in [11], however in that paper the authors propose a two step procedure, computing explicitly the partial derivatives in the perspective variables (ξ,η). In other words, they do not treat the system (8) as a PDE system, but rather as a linear system, where the unknowns i.e. the entries of ∇z (namely $p = \frac{\partial z}{\partial \xi}$ and $q = \frac{\partial z}{\partial \eta}$), are computed *locally*.

Let us start by taking into account the differential formulation (8). We exploit the linearity of the hyperbolic equations in (8) by simply summing them, resulting in the single differential equation

$$\begin{cases} \left(b^{(1,2)} + b^{(1,3)} + b^{(2,3)}\right)\cdot\nabla z(\xi,\eta) + \left(s^{(1,2)} + s^{(1,3)} + s^{(2,3)}\right)z(\xi,\eta) = 0 \\ z(\xi,\eta) = g(\xi,\eta). \end{cases} \tag{11}$$

It is clear that, since the solution of each equation in (8) is the same (i.e. the differential problem (5) has a unique solution [9]), the problem (11) will be also satisfied by the same solution. However, a proof of uniqueness can not be obtained as a consequence of this sum. In fact, it is easy to prove that by subtracting some terms instead of summing all the addends, the problem becomes ill-posed. That is why we shall prove the existence of a unique weak solution for a problem such as (11) which also takes into account shadows and occlusions. In order to have a well-posed problem the boundary condition $g(\xi, \eta)$ is needed which can be readily obtained using the three available images and a two step procedure applied only on the boundary pixels assuming no occlusions on $\partial\Omega^p$.

Note that, if more than three images are available, we can easily generalize this reasoning. In the general case, defining the functions

$$b_n(\xi,\eta) = \sum_{r\in\binom{[n]}{2}} b^r(\xi,\eta) \quad \text{and} \quad s_n(\xi,\eta) = \sum_{r\in\binom{[n]}{2}} s^r(\xi,\eta) \tag{12}$$

the extension of the PDE-based approach for the PPS_n problem can be readily stated as

$$\begin{cases} b_n(\xi,\eta)\cdot\nabla z(\xi,\eta) + s_n(\xi,\eta)z(\xi,\eta) = 0, \text{ a.e. } (\xi,\eta)\in\Omega^p \\ z(\xi,\eta) = g(\xi,\eta) \qquad\qquad\qquad\qquad\qquad \forall(\xi,\eta)\in\partial\Omega^p \end{cases} \tag{13}$$

where with $\binom{[n]}{2}$ we call the set that contains the couple of integer indexes with no repetition. For example, if $n = 3$ we have $\binom{[n]}{2} = \{(1,2),(1,3),(2,3)\}$.

3.1 Weighted Perspective Photometric Stereo for Multiple Images with Shadows

The main idea of this paper is based on the possibility of ensuring the well-posedness of the PPS_n problem formulation (13) by exploiting the linearity of the operation involved in the basic differential formulation (5). It is also clear that (5) still does not lose the well-posedness if we multiply both sides (i.e. $b(\xi,\eta)$ and $s(\xi,\eta)$) by a function $q(\xi,\eta)$. That is:

$$\begin{cases} q(\xi,\eta)b(\xi,\eta)\cdot\nabla z(\xi,\eta) + q(\xi,\eta)s(\xi,\eta)z(\xi,\eta) = 0 \\ z(\xi,\eta) = g(\xi,\eta) \end{cases} \tag{14}$$

still has a unique Lipschitz solution. We do not go deeper with the discussion on the weak regularity of q. Here, we merely consider it as a smooth function.

We are now able to define the weighted PPS_n equation (W-PPS_n) by replacing b_n, s_n in (12) with

$$b_n^w(\xi,\eta) = \sum_{r\in\binom{[n]}{2}} q_r(\xi,\eta)b^r(\xi,\eta) \quad \text{and} \quad s_n^w(\xi,\eta) = \sum_{r\in\binom{[n]}{2}} q_r(\xi,\eta)s^r(\xi,\eta) \tag{15}$$

where the index r is used here only to make clear that we are now considering $\binom{n}{2}$ continuous functions.

We have now completed the set-up of the W-PPS$_n$ formulation with

$$\begin{cases} b_n^w(\xi,\eta)\cdot\nabla z(\xi,\eta)+s_n^w(\xi,\eta)z(\xi,\eta)=0, \text{ a.e. } (\xi,\eta)\in\Omega^p \\ z(\xi,\eta)=g(\xi,\eta) \qquad\qquad\qquad\qquad \forall(\xi,\eta)\in\partial\Omega^p. \end{cases} \tag{16}$$

We will explain in the next part how shadows will influence the definition of the vector field b_n^w and the scalar field s_n^w.

A key point is the possibility to use weights q_r that are not only positive. It is possible to maintain the well-posedness of the problem also by considering non-negative weights q_r that vanish at some points for some image pairs.

It can be shown that for the set of well-posed differential equations, maintaining a non-negative weight for at least one image pair suffices to give us a well-posed problem.

This allows us to adapt the W-PPS$_n$ equations for the case of shadows in some of the images. Specifically, let $\mathcal{S}^r$ define the areas that are shaded in either of the images in pair r. We define $\widetilde{q}_r$ as the indicator function,

$$\widetilde{q}_r(\xi,\eta)=\mathbb{1}_{[\overline{\Omega}^p\setminus\mathcal{S}^r]}(\xi,\eta). \tag{17}$$

In other words we consider the weights as switches able to locally put out go the global sums in (15) the functions b_r and s_r that do not contain relevant information due to the presence of shadows in the involved images. Finally we construct the weights q_r as smooth cutoff functions based on $\widetilde{q}_r$.

4 Uniqueness of the Weak Solution of W-PPS$_3$

In order to complete the theoretical analysis we will extend the uniqueness results of the differential problem (16) in the case of a weak solution. Discussion of depth-discontinuities and multiple objects is beyond the scope of this paper. Our purpose is to prove the uniqueness of solution of (16) in the Lipschitz function space via characteristics method. The meaning of weak solution here is intended as combination of classical solutions, each defined on a different domain. These domains are then going to be patched together in such a way that, across the boundaries between domains on which there are discontinuities in some derivatives, the equation (16) is satisfied. Let us recall that the points where the surface z is not differentiable are the same where the functions b_n^w and s_n^w are discontinuous (jump discontinuity) [8]. We assume the discontinuity points as the family of regular curves $(\gamma_1(t),\ldots,\gamma_k(t))$ where t is the argument of the parametric representation.

A complete proof of the well-posedness of our model can be given in a manner similar to [16]. It is based on the following two features of our model:

1. the absence of critical points for the projected characteristic field, i.e. $b_3^w(\xi,\eta)\neq(0,0)$;
2. the propagation of the information from the boundary is not prevented between two sets separated by discontinuity curves $(\gamma_1(t),\ldots,\gamma_k(t))$, see Fig. 2.

The following result is very important since it guarantees the absence of critical points that would prevent the method to work.

Lemma 1. *Assume that* $\bigcap_r \mathcal{S}^r = \emptyset$. *Then* $|b_3^w(\xi,\eta)| \neq 0$, $\forall(\xi,\eta) \in \Omega^p$.

This last result is not only important for the proof of uniqueness of weak solution. We use it also for the well-posedness of the numerical schemes introduced in Section 5.

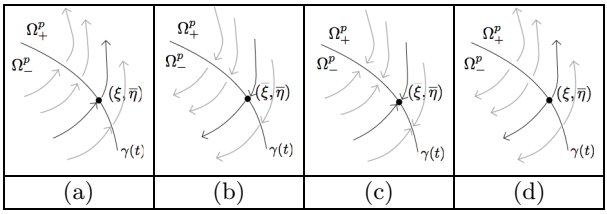


Fig. 2. Among the four possibilities shown for b_3^w, only the cases (a) and (b) allow the information to cross the discontinuity curve γ without needing additional data as required in (c) and (d)

The next results ensure that the characteristic method can actually be applied since the discontinuity on $\gamma(t)$ is not an obstacle for propagating a solution of the PDE.

Theorem 1. *Let* $\gamma(t)$ *be a regular curve of discontinuity for the function* $b_3^w(\xi,\eta)$ *(and* $s_3^w(\xi,\eta)$*) and let* $(\overline{\xi},\overline{\eta})$ *be a point along* $\gamma(t)$. *Let* $n(\overline{\xi},\overline{\eta})$ *be the outgoing normal with respect to the set* Ω^p_+, *then we have*

$$\left[\lim_{\substack{(\xi,\eta)\to(\overline{\xi},\overline{\eta})\\(\xi,\eta)\in\Omega^p_+}} b_3^w(\xi,\eta)\cdot n(\overline{\xi},\overline{\eta})\right]\left[\lim_{\substack{(\xi,\eta)\to(\overline{\xi},\overline{\eta})\\(\xi,\eta)\in\Omega^p_-}} b_3^w(\xi,\eta)\cdot n(\overline{\xi},\overline{\eta})\right] \geq 0 \qquad (18)$$

The result that permits to prove the uniqueness of weak solution is readily proved now. With Lemma 1 and Theorem 1 it is possible to show that the uniqueness can be reached using the characteristic strip method. In order to understand the idea behind the proofs of Lemma 1 and Theorem 1 we refer to [16] where the same results are proved in the case with only two images.

We emphasize once more the advantages of this new formulation with respect to [11]. The first is obviously the direct computation of the height of the surface, without passing through the preliminary computation of the partial derivatives. This would result in a slower computation of the 3D surface and also needs the condition that the 3D surface has to be smooth. That is the surface should be at least C^1. The second and much more important point is that, since our W-PPS$_3$ model is based on the differential problem (5) for two images (which admits a unique Lipschitz solution), even if we have three images with disjoint shadows we can still reconstruct the surface.

5 Numerical Schemes

Next, we consider the numerical methods used to obtain the solution. The difference among those presented in [9] and our is related to the different implementation. It allows to speed up the convergence of the four numerical schemes we will discuss in the following sections. The algorithms we implemented use the *fast sweeping* technique [17–20] which exploits the regularity of the vector field b_3^w.

For the numerical schemes we consider the domain $\overline{\Omega}^p = [a^p, b^p] \times [c^p, d^p] = [-1, 1]^2$ with a uniform discretization space step $\Delta_\xi = (b^p - a^p)/n$ and $\Delta_\eta = (c^p - d^p)/m$ where n and m are the number of intervals divide the sides of the rectangular domain (that is $\xi_i = a^p + i\Delta_\xi$, $\eta_j = c^p + j\Delta_\eta$ with $i = 0, \ldots, n$ and $j = 0, \ldots, m$). We will denote by $\overline{\Omega}_d^p$ all the points of the lattice belonging to $\overline{\Omega}^p$, by Ω_d^p all the internal points and by $\partial\Omega_d^p$ all the boundary points.

5.1 Forward Numerical Schemes

We want to recall now the numerical schemes used for the forward approximation of (16) where the propagation of the information is considered starting from the inflow part of the boundary

$$\Gamma_{in} = \left\{ (\widetilde{\xi}, \widetilde{\eta}) \in \partial\Omega^p : \nu(\widetilde{\xi}, \widetilde{\eta}) \cdot \lim_{\substack{(\xi.\eta)\to(\widetilde{\xi},\widetilde{\eta}) \\ (\xi,\eta)\in\Omega^p}} b_3^w(\xi, \eta) \leq 0 \right\} \tag{19}$$

where $\nu(\xi, \eta)$ represents the outgoing normal to the boundary $\partial\Omega^p$. It is clear that in the previous definition the limit is taken since it can happen that a discontinuity curve can coincide with the boundary. Now we can formulate the differential problem solved by the forward schemes as follow:

$$\begin{cases} b_3^w(\xi, \eta) \cdot \nabla z(\xi, \eta) + s_3^w(\xi, \eta) z(\xi, \eta) = 0, \text{ a.e. } (\xi, \eta) \in \Omega^p \\ z(\xi, \eta) = g(\xi, \eta) \qquad\qquad \forall (\xi, \eta) \in \Gamma_{in}. \end{cases} \tag{20}$$

In order to simplify the notation we will call $b_3^w(\xi_i, \eta_j)$ as $b_{i,j} = (b_{i,j}^1, b_{i,j}^2)$ and $s_3^w(\xi_i, \eta_j)$ as $s_{i,j}$.

Forward Up-Wind Scheme:

$$Z_{i,j}^F = \frac{\Delta_\eta |b_{i,j}^1| Z_{i-\mathrm{sgn}(b_{i,j}^1),j}^F + \Delta_\xi |b_{i,j}^2| Z_{i,j-\mathrm{sgn}(b_{i,j}^2)}^F}{|b_{i,j}^1|\Delta_\eta + |b_{i,j}^2|\Delta_\xi + \Delta_\xi \Delta_\eta s_{i,j}}. \tag{21}$$

In our case the numerical schemes are applied to digital images where clearly $\Delta_\xi = \Delta_\eta = \Delta$.

Forward Semi-Lagrangian Scheme:

$$z_{i,j}^F = z^F(\xi_i - h\alpha_{i,j}^1, \eta_j - h\alpha_{i,j}^2) \frac{|b_{i,j}|}{|b_{i,j}| + h s_{i,j}} \tag{22}$$

where $\alpha_{i,j} = \frac{b_{i,j}}{|b_{i,j}|}$ and the parameter $h > 0$ is assumed equal to the size of the grid Δ in order to reach the highest order of convergence equal to one ([9]).

5.2 Backward Numerical Schemes

The backward numerical schemes are based on the approximation of the surface propagating the information stored on the outflow part of the boundary

$$\Gamma_{out} = \partial\Omega^p \setminus \Gamma_{in}. \tag{23}$$

The formulation of these schemes can be easily obtained considering the following equivalent problem

$$\begin{cases} -b_3^w(\xi,\eta)\cdot\nabla z(\xi,\eta) - s_3^w(\xi,\eta)z(\xi,\eta) = 0, \text{ a.e. } (\xi,\eta)\in\Omega^p \\ z(\xi,\eta) = g(\xi,\eta) \qquad\qquad\qquad\qquad \forall(\xi,\eta)\in\Gamma_{out} \end{cases} \tag{24}$$

and repeating always the same passages for the forward ones.

Backward Up-Wind Scheme:

$$Z^B_{i,j} = \frac{\Delta_\eta |b^1_{i,j}| Z^B_{i+\mathrm{sgn}(b^1_{i,j}),j} + \Delta_\xi |b^2_{i,j}| Z^B_{i,j+\mathrm{sgn}(b^2_{i,j})}}{|b^1_{i,j}|\Delta_\eta + |b^2_{i,j}|\Delta_\xi + \Delta_\xi\Delta_\eta s_{i,j}}. \tag{25}$$

Backward Semi-Lagrangian Scheme:

$$z^B_{i,j} = z^B(\xi_i + h\alpha^1_{i,j}, \eta_j + h\alpha^2_{i,j})\frac{|b_{i,j}|}{|b_{i,j}| - hs_{i,j}}. \tag{26}$$

Let us emphasize that, in order to have all these schemes well defined, we have to take the parameter Δ (equal to h) small enough to have:

$$\begin{aligned} |b^1_{i,j}| + |b^2_{i,j}| + \Delta s_{i,j} \neq 0 \quad & \text{for both forward and backward u-w schemes} \\ |b_{i,j}| + hs_{i,j} \neq 0 \quad & \text{for the forward s-L scheme} \\ |b_{i,j}| - hs_{i,j} \neq 0 \quad & \text{for the backward s-L scheme} \end{aligned} \tag{27}$$

always possible since $|b_3^w(\xi,\eta)| \neq 0$, $\forall(\xi,\eta)\in\Omega^p$, Lemma 1 ([9]). Due to lack of space it is not possible to give theoretical results regarding these numerical schemes. An exhaustive discussion about the consistency, proof of convergences and estimation of the error with perturbed data, can be found in [16] where the case with only two images is taken into account.

6 Numerical Tests

We now present several results of our method. We will consider the W-PPS$_3$ problem with some *artificial shadow* regions defined in the images. The smooth surface v_{reg} exhibits three high slopes. The second one v_{lip} is a Lipschitz surface with a very high Lipschitz constant (i.e. the gradient changes sharply its direction across the point where the surface is not differentiable). Note also that the boundary condition is not constant for either of them. In particular, v_{lip} has a boundary condition differentiable almost everywhere. As mentioned at the

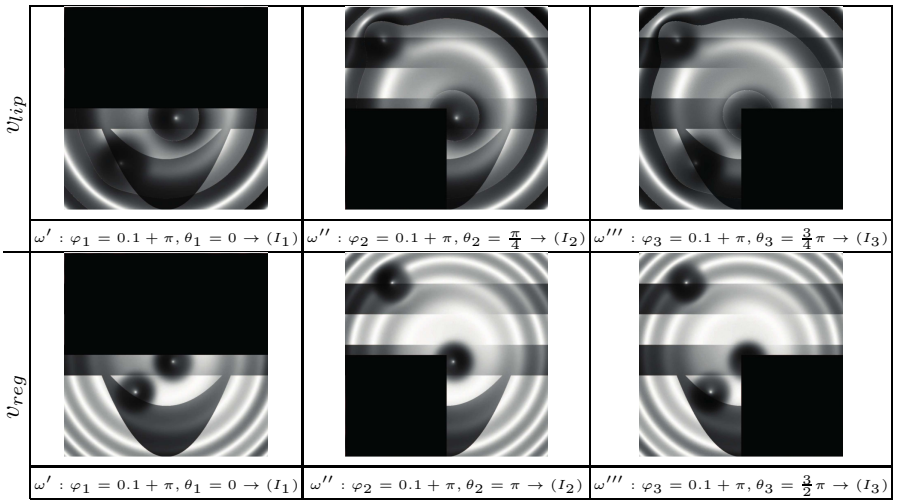


Fig. 3. Set of images used with the respective light sources described by their spherical coordinates. In this case the albedo mask and Gaussian noise (10%) is added for all the images.

beginning of the paper, well-posedness holds even if the albedo is not known. In order to exploit this advantage of our model we consider the initial images shown in Fig. 3. In order to reconstruct the surface in the worst situation we set artificial shadows for which the union of the shadow sets almost completely covers the image domain. In Fig. 3 are shown the starting data, images and light sources directions, used in the numerical tests.

Table 1. The values of this table explain how (in precision and in time) the semi-Lagrangian and the up-wind schemes converge for the v_{lip} case

v_{lip}		Forward schemes				Backward schemes			
	Δ	L^∞ s-L	time (sec)	L^∞ u-w	time (sec)	L^∞ s-L	time (sec)	L^∞ u-w	time (sec)
10 %	500	1.552×10^{-1}	0.259	2.613×10^{-1}	0.500	1.586×10^{-1}	0.026	3.014×10^{-1}	0.023
	1000	9.586×10^{-2}	1.313	1.651×10^{-1}	2.257	9.968×10^{-2}	0.135	1.818×10^{-1}	0.078
	2000	5.956×10^{-2}	5.676	1.020×10^{-1}	8.314	6.068×10^{-2}	0.483	1.090×10^{-1}	0.338
	4000	3.957×10^{-2}	21.372	6.366×10^{-2}	32.089	3.856×10^{-2}	1.650	6.650×10^{-2}	1.247
	500	1.980×10^{-1}	0.273	2.650×10^{-1}	0.492	2.587×10^{-1}	0.031	3.065×10^{-1}	0.021
	1000	1.247×10^{-1}	1.516	1.832×10^{-1}	2.431	1.237×10^{-1}	0.109	2.001×10^{-1}	0.080
	2000	8.742×10^{-2}	5.601	1.127×10^{-1}	8.786	8.805×10^{-2}	0.418	1.194×10^{-1}	0.325
	4000	9.098×10^{-2}	21.687	1.127×10^{-1}	8.786	9.080×10^{-2}	1.642	1.024×10^{-1}	1.258

The size of the images take into account varies from 500×500 pixels (with $\Delta = 0.004$) to 4000×4000 pixels, that is 16 megapixels (for a spacial step $\Delta = 0.0005$). The running times quoted are for a 2.4 Ghz Core i5 computer with 8 GB (1333Mhz) of RAM. Tables 1 and 2 show that the convergence of the schemes is not prevented by the presence of noise even if the consistency order is not even one like for the images without noise. The computational time is very small even for the largest size images. The difference between the forward and

Table 2. The values of this table explain how (in precision and in time) the semi-Lagrangian and the up-wind schemes converge for the v_{reg} case

v_{reg}		Forward schemes				Backward schemes			
	Δ	L^∞ s-L	time (sec)	L^∞ u-w	time (sec)	L^∞ s-L	time (sec)	L^∞ u-w	time (sec)
10 %	500	6.152×10^{-2}	0.077	1.916×10^{-1}	0.062	6.152×10^{-2}	0.031	2.671×10^{-1}	0.019
	1000	3.237×10^{-2}	0.319	1.263×10^{-1}	0.252	3.234×10^{-2}	0.104	1.390×10^{-1}	0.098
	2000	1.672×10^{-2}	1.416	8.065×10^{-2}	1.098	1.671×10^{-2}	0.415	8.167×10^{-2}	0.331
	4000	8.518×10^{-3}	5.024	5.141×10^{-2}	3.954	8.515×10^{-3}	1.642	5.178×10^{-2}	1.233
	500	1.019×10^{-1}	0.077	2.186×10^{-1}	0.156	1.024×10^{-1}	0.026	2.395×10^{-1}	0.019
	1000	1.303×10^{-1}	0.324	1.894×10^{-1}	0.737	1.299×10^{-1}	0.103	1.913×10^{-1}	0.106
	2000	1.048×10^{-1}	1.462	1.193×10^{-1}	3.681	1.052×10^{-1}	0.492	1.202×10^{-1}	0.327
	4000	4.698×10^{-2}	5.096	7.186×10^{-2}	16.935	4.691×10^{-2}	1.649	7.228×10^{-2}	1.255

the backward time of convergences is due to the direction of the vector field b_3^w which results for both cases much more easy passable from the backward than the forward.

Fig. 4 demonstrates the results obtained with the semi-Lagrangian and up-wind fast-sweeping approach.

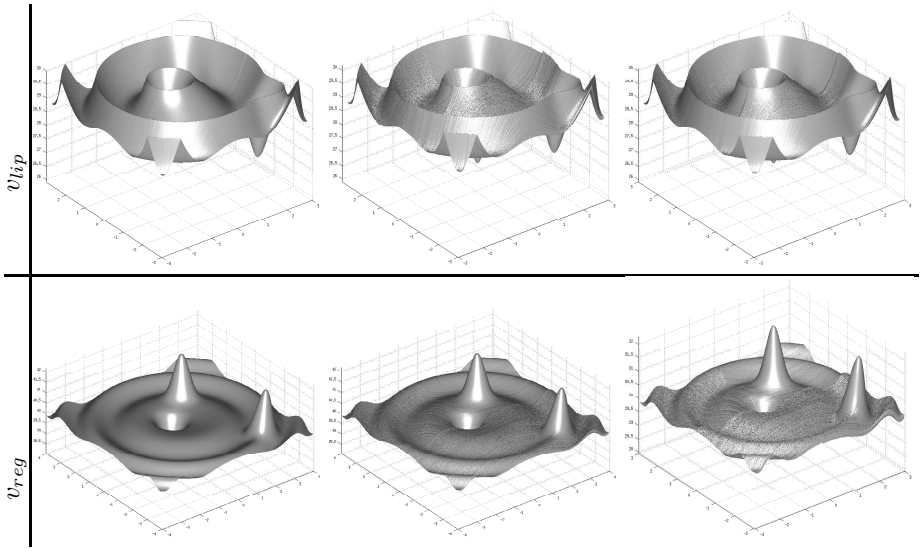


Fig. 4. Left-to-right: groundtruth surface, reconstruction via the semi-Lagrangian scheme, reconstruction via the up-wind scheme

7 Conclusion and Perspective

In this paper we have presented a new direct method for Photometric Stereo in the case of perspective viewing geometry in the case of multiple images and shadows. Using a fast-sweeping update, we are able to update the solution along characteristic lines in an efficient an accurate manner. The resulting algorithm is highly parallelizable and efficient to compute also on a single CPU, and seems promising for real-time implementation.

References

1. Wu, C., Narasimhan, S.G., Jaramaz, B.: A multi-image shape-from-shading framework for near-lighting perspective endoscopes. IJCV, 211–228 (2009)
2. Deguchi, K., Okatani, T.: Shape reconstruction from an endoscope image shape-from-shading technique for a point light source at the projection center. In: Workshop on MMBIA, pp. 290–298. IEEE Computer Society (1996)
3. Yeung, S.Y., Tsui, H.T., Yim, A.: Global shape from shading for an endoscope image. In: Taylor, C., Colchester, A. (eds.) MICCAI 1999. LNCS, vol. 1679, pp. 318–327. Springer, Heidelberg (1999)
4. Hernández, C., Vogiatzis, G., Cipolla, R.: Shadows in three-source photometric stereo. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 290–303. Springer, Heidelberg (2008)
5. Onn, R., Bruckstein, A.M.: Integrability Disambiguates Surface Recovery in Two-Image Photometric Stereo. IJCV 5, 105–113 (1990)
6. Lee, S., Brady, M.: Integrating stereo and photometric stereo to monitor the development of glaucoma. Image and Vision Computing 9, 39–44 (1991)
7. Mecca, R.: Uniqueness for shape from shading via photometric stereo technique. In: ICIP, pp. 2933–2936 (2011)
8. Mecca, R., Falcone, M.: Uniqueness and approximation of a photometric shape-from-shading model. Accepted to SIAM Journal on Imaging Sciences (2012)
9. Mecca, R., Tankus, A., Bruckstein, A.M.: Two-Image Perspective Photometric Stereo Using Shape-from-Shading. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part IV. LNCS, vol. 7727, pp. 110–121. Springer, Heidelberg (2013)
10. Kimmel, R., Yavneh, I.: An algebraic multigrid approach for image analysis. SIAM Journal on Scientific Computing 24, 1218–1231 (2003)
11. Tankus, A., Kiryati, N.: Photometric stereo under perspective projection. In: ICCV, pp. 611–616 (2005)
12. Kimmel, R., Bruckstein, A.M.: Tracking level sets by level sets: A method for solving the shape from shading problem. CVIU 62, 47–58 (1995)
13. Horn, B.K.P., Brooks, M.J.: Shape from Shading. The MIT Press (1989)
14. Prados, E., Faugeras, O.D.: Shape from shading: A well-posed problem? In: CVPR, vol. 2, pp. 870–877. IEEE Computer Society (2005)
15. Tankus, A., Sochen, N.A., Yeshurun, Y.: Shape-from-shading under perspective projection. IJCV 63, 21–43 (2005)
16. R.Mecca, Tankus, A., Bruckstein, A.: Two-image perspective photometric stereo: Analytical and numerical analysis of a new differential model. Technical report, Technion - Israel Institute of Technology, Computer Science Department (2013)
17. Danielsson, P.E.: Euclidean distance mapping. Computer Graphics And image Processing 14, 227–248 (1980)
18. Qian, J., Zhang, Y.T., Zhao, H.K.: Fast sweeping methods for eikonal equations on triangular meshes. SIAM J. Numer. Anal. 45, 83–107 (2007)
19. Weber, O., Devir, Y.S., Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Parallel algorithms for approximation of distance maps on parametric surfaces. ACM Trans. Graph. 27, 104:1–104:16 (2008)
20. Chacon, A., Vladimirsky, A.: Fast two-scale methods for eikonal equations. SIAM J. Scientific Computing 34 (2012)

Solving the Uncalibrated Photometric Stereo Problem Using Total Variation

Yvain Quéau[1], François Lauze[2], and Jean-Denis Durou[1]

[1] IRIT, UMR CNRS 5505, Toulouse, France
yvain.queau@enseeiht.fr, durou@irit.fr
[2] Dept. of Computer Science, Univ. of Copenhagen, Denmark
francois@diku.dk

Abstract. In this paper we propose a new method to solve the problem of uncalibrated photometric stereo, making very weak assumptions on the properties of the scene to be reconstructed. Our goal is to solve the generalized bas-relief ambiguity (GBR) by performing a total variation regularization of both the estimated normal field and albedo. Unlike most of the previous attempts to solve this ambiguity, our approach does not rely on any prior information about the shape or the albedo, apart from its piecewise smoothness. We test our method on real images and obtain results comparable to the state-of-the-art algorithms.

1 Introduction

Photometric stereo has first been introduced by Woodham in [28] in the early 80's, using the commonly used Lambertian model to recover both the surface shape and its albedo, given $m \geqslant 3$ pictures of a scene taken from the same viewpoint but under different illumination conditions. In this pioneering work, the lights are supposed to be known. When these lighting conditions are unknown, the problem is much harder, as one has to estimate both the normal field and the lights. Assuming the *integrability* of the surface [30], this can be done up to a GBR transformation [4]. This transformation is a 3-parameters (here they will be called μ, ν and λ) transformation which affects the normal field (and thus the lights) without changing the images the scene can create or the ability of the field to be "integrated" into a shape. Despite the few number of parameters induced by this transformation, estimating them without any assumption on the scene we want to reconstruct appears to be everything but an easy task, and most methods addressing this issue assume prior knowledge of shape properties [14], albedo distribution [3] or presence of outliers which violate the Lambertian assumption [8].

We introduce a new method for estimating these parameters, which does not rely on any of those assumptions, except that we say the surface parameters (albedo and normal field) should "vary few apart from the edges", which seems quite a reasonable assumption, and appears to work as well as the state-of-the-art methods like [10]. Total variation minimization was introduced by [23] for

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 270–281, 2013.

noise removal because of its edge preserving property, and has become an ubiquitous tool in image analysis. This regularization method has the very interesting property to preserve edges of an object, depending on its scale [25]. This property has, to our knowledge, never been used for photometric stereo, and appears to be a good way to propose both a new model for uncalibrated photometric stereo and a novel method for solving the generalized bas-relief ambiguity.

Our paper is organized as follows: after recalling the basic equations to solve the uncalibrated photometric stereo (UPS) problem (see Sec. 3) up to a GBR, we propose in Sec. 4 a new method to recover the GBR parameters, before introducing a new model for the UPS problem (Sec. 5) and a 3-step solution to it (Sec. 5.2). Finally, we present some results on synthetic and real images in Sec. 6.

2 Prior Work

Photometric stereo (PS) [28] was introduced in the early 80's for dealing with 3D-reconstruction, allowing the user to recover both the surface normal field and the albedo at the same time. In this technique, m images of a scene are taken from the same viewpoint but under variable lighting conditions. Unlike traditional stereo 3D-reconstruction, only the visible face can be reconstructed (PS is thus a 2.5D-reconstruction method), although it has been shown in [15] that it could be coupled with multi-view techniques to acquire the full shape.

It is an extension of the shape-from-shading problem [17], which is known for being ill-posed. The use of additional images with different lighting conditions allows one to solve for the ambiguities of shape-from-shading. Under the condition of a Lambertian reflectance model and knowledge about the light sources positions, a very efficient reconstruction can be achieved, recovering up to the tiniest details of the scene normal field (see Eq. 2 and Eq. 3). An additional step is then needed to "integrate" the field into a surface. This step, which can be very tricky, will not be presented in this paper, but the reader could learn about it in [9,11,18].

The usual assumptions about the number of images (*i.e.* the number of light sources) is that $m \geqslant 3$ and the sources should be non-coplanar. Some attempts have been made to solve the problem with $m = 2$ in [22], and a very interesting recent application of the coplanar sources configuration is the 3D-reconstruction from an outdoor webcam (as the Sun moves within a plane), which is dealt with by Abrams *et al.* in [1] and by Ackermann *et al.* in [2]. In those papers, the lighting configuration is deduced from GPS coordinates and time.

In this paper, we will focus on the traditional problem with $m \geqslant 3$ non-coplanar distant light sources, which are supposed to be unknown. The problem becomes the so-called *uncalibrated photometric stereo* problem, which was first addressed by Hayakawa in [14]. Using a SVD, Hayakawa shows that one can solve the problem up to a linear transformation (see Eq. 4), and given some (strong) prior knowledge on the normals distribution he gives a way to recover the lights, normals and albedo. Assuming the estimated normal field should derive from

a surface (the one we want to reconstruct), Yuille and Snow showed in [30] that one could impose the "integrability constraint" (see Eq. 5) to reduce the linear ambiguity to a special type of transformation, called *generalized bas-relief* (GBR) and presented by Belhumeur *et al.* in [4] (see Eq. 6).

In order to solve this last ambiguity, several approaches have been presented. Alldrin proposed in [3] to choose the parameters which minimize the entropy of the albedo distribution, so as to compress the range of albedo values. This should work for most simple objects but may perform poorly on some complex materials. Shi *et al.* advocated the use for 3-channels colour information in [24] and proved this information (when available) can resolve the ambiguity. Another approach is to use the information given by the outliers of the Lambertian model, like interreflections [7], specularities [8] or shadows [26]. Those approaches suppose that such outliers are available (which is the case for most real-world scenes) and that one can detect them, which can be achieved using low-rank approximation as in [19] and [29]. One could also use another illumination model like the Torrance and Sparrow model as described in [12], or make no assumption at all about the model by using a reference object, as in the work of Hertzmann and Seitz [16], or, assuming an additive non-Lambertian reflectance component, identify pixels which hold the isotropy and reciprocity constraints like in [27].

However all these methods are either doing some assumptions on the model (even though these assumptions are mostly realistic) or very difficult to set up and long to converge. Recently, Favaro and Papadhimitri proposed in [10] a new method for solving the GBR ambiguity, making very few assumptions on the model, by using local diffuse maxima. Their method showed to perform as good or better than previous ones with fewer assumptions, and thus will be our reference for the experiments.

3 Photometric Stereo

3.1 Calibrated Photometric Stereo

In the sequel, we note I_p^i the intensity of pixel p in the i^{th} image, with $i \in [1, m]$ and $m \geqslant 3$. ρ_p is the albedo in pixel p, $N_p = [N_x, N_y, N_z](p)$ is the surface normal in p, and $S^i = [S_x^i, S_y^i, S_z^i]^\top$ will be the light source, in norm and direction, in the i^{th} image. We assume the light is constant over each image (directional light), so that S^i does not depend on p.

According to the Lambertian model, in every pixel p and every image i, one can write the equation:

$$I_p^i = \rho_p N_p S^i \tag{1}$$

Writing $I_p = [I_p^1, \ldots, I_p^m]$, $S = [S^1, \ldots, S^m]$ and $M_p = \rho_p N_p$, we obtain the system of linear equations $I_p = M_p S$. If S is known and is of rank 3 (3 sources at least are non-coplanar), a least-square solution is given by the Moore-Penrose pseudo-inverse, and one can recover both the normal and the albedo in p:

$$\widehat{M}_p = I_p S^+ \qquad \widehat{N}_p = \frac{\widehat{M}_p}{\|\widehat{M}_p\|} \qquad \widehat{\rho}_p = \|\widehat{M}_p\| \tag{2}$$

Stacking each image column-wise, we can note $I = [I_1^\top, \dots, I_{|\Omega|}^\top]^\top$ where Ω is a mask of the scene within the image and $|\Omega|$ is the number of pixels inside this mask. Similarly, we note $M = [M_1^\top, \dots, M_{|\Omega|}^\top]^\top$, so that the Lambert's law can be written $I = MS$. We can now rewrite Eq. 2 into Eq. 3 in order to get $\widehat{M}$:

$$\widehat{M} = IS^+ \tag{3}$$

3.2 Uncalibrated Photometric Stereo

When the light matrix S is unknown, things are much more complicated. We now want to estimate both $\widehat{M}(p) = \widehat{\rho}(p)\widehat{N}(p)$ in every pixel p and a $3 \times m$ light matrix $\widehat{S}$.

Finding $\widehat{S}$ and $\widehat{M}$ satisfying $I = \widehat{M}\widehat{S}$ is not that hard, as it can be done in a least square sense using SVD [14]. Indeed, I can be decomposed in $I = UWV^T$, with $U \in \mathbb{R}^{|\Omega|\times|\Omega|}$, $W \in \mathbb{R}^{|\Omega|\times m}$ and $V \in \mathbb{R}^{m\times m}$. As both lightings and normals lie in $\mathbb{R}^3$, I should be of rank 3, and thus it is reasonable to restrain W to its first 3×3 submatrix, and U and V to their first 3 columns, *i.e* $U \in \mathbb{R}^{|\Omega|\times 3}$, $W \in \mathbb{R}^{3\times 3}$ and $V \in \mathbb{R}^{m\times 3}$, so that $I \approx UWV^T$. The solution can be finally obtained by Eq. 4 :

$$\begin{cases} \widehat{M} = UP^T \\ \widehat{S} = QV^T \end{cases} \tag{4}$$

where P and Q are two 3×3 matrices verifying $P^TQ = W$.

But the solution is not unique, as there is an infinity of such (P, Q) matrices. Yuille and Snow showed that imposing the integrability constraint on the estimated normal field $\widehat{N}$ reduces this ambiguity to a GBR [30]. The integrability constraint has the form:

$$\overline{\text{curl}}\,\widehat{N} = 0, \quad \text{where } \overline{\text{curl}}\,[a, b, c] = \frac{\partial}{\partial y}\frac{a}{c} - \frac{\partial}{\partial x}\frac{b}{c} \tag{5}$$

and extends immediately to $\widehat{M}$.

Expanding this equation, they show one can identify 6 over the 9 coefficients in P^{-1}, and the remaining 3 correspond to the GBR. They propose to fix those 3 to random values, and then the only transformation which would hold both the Lambertian assumption and the integrability constraint is $\widehat{M'} = \widehat{M}G$ and $\widehat{S'} = G^{-1}\widehat{S}$ with G and G^{-1} given in Eq. 6:

$$G(\mu, \nu, \lambda) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mu & \nu & \lambda \end{pmatrix} \qquad G^{-1}(\mu, \nu, \lambda) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{\mu}{\lambda} & -\frac{\nu}{\lambda} & \frac{1}{\lambda} \end{pmatrix} \tag{6}$$

Estimating the best parameters μ, ν and λ is still an open problem for research, especially difficult as the images produced by the transformed normals and lightings are the exact same, and the normal fields are exactly as "integrable". We now propose a new method for estimating those parameters.

4 Solving the GBR Ambiguity with Total Variation

4.1 Total Variation of a Vector Field

The total variation of a function is a widely used measure for regularization. For an almost everywhere differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, it can be written as: $TV(f) = \int_{\mathbb{R}^n} |\nabla f(x)| dx$ and extends to the class of so-called functions of bounded variations. In the sequel we take $n = 2$ as we deal with planar images. When f takes its values in $\mathbb{R}^m$ with $m > 1$, we define $TV(f) = \int_{\mathbb{R}^2} \|J(f)\|_F dx$, where $\|J(f)\|_F$ is the Frobenius norm of the Jacobian matrix of f, although other choices may be considered (see [5] for some discussion).

Approximating this in the discrete case, and adapting it to a vector field $M : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3,\ p \mapsto [M_x(p), M_y(p), M_z(p)]$, we get Eq. 7:

$$TV(M) = \sum_{p \in \Omega} \sqrt{\|\nabla M_x(p)\|^2 + \|\nabla M_y(p)\|^2 + \|\nabla M_z(p)\|^2} \tag{7}$$

where ∇ is a suitable discrete gradient operator.

4.2 Why Use Total Variation

In [3], Alldrin *et al.* propose to choose the GBR parameters which minimize the entropy of the histogram of albedo. They want to favor materials which are "homogeneous", *i.e.* made of a small amount of components. This looks a reasonable assumption, but the entropy of the albedo can be pretty tricky to minimize. Besides, this entropy does not consider spatial variation of the albedo, and when looking for "homogeneous" zones, one would expect that similar albedo pixels would be close to each other.

When trying to find such homogeneous materials, a similar approach can be to use the total variation of the albedo: it also favors homogeneous zones as we would expect the variations of albedo to be small apart from the edges. And, contrary to standard Tikhonov regularization, these edges are better preserved: this will lead to different zones in the image, and inside each zone the albedo would vary few, but we allow it to vary between adjacent zones. This effect of total variation, called stair-casing, is well-known and studied [21].

Thus, to take into account the spatial consistency, total variation minimization of the albedo seems to be a more consistent choice than entropy minimization, which cannot choose between different configurations having the same histogram of albedo, as shown in Fig. 1. As the albedo is linked to the vector field M by $\rho = ||M||$, and as the GBR transformation G with parameters μ, ν and λ transforms M into $MG(\mu, \nu, \lambda)$, we could look for a GBR transformation which minimizes $TV(||MG(\mu, \nu, \lambda)||)$, *i.e* estimate the three parameters (μ, ν, λ) such that:

$$(\widehat{\mu}, \widehat{\nu}, \widehat{\lambda}) = \underset{(\mu,\nu,\lambda) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+}{\operatorname{argmin}} TV(\|MG(\mu, \nu, \lambda)\|) \tag{8}$$

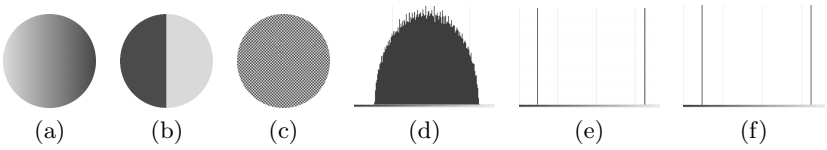


Fig. 1. Three different albedo configurations and the corresponding histograms: (a) has a huge entropy value, cf. histogram (d); (b) and (c) have the same small entropy, cf. histograms (e) and (f), but the spatial distributions of the albedo are very different. Total variation would tend to favor distributions like (b).

But we can do even better: it seems reasonable to also ensure that the unit normal field N should vary few inside zones. Thus one can consider minimizing both the total variation of the albedo and the total variation of the field. As they are simply linked by $M = \rho N$, we can try to estimate this GBR transformation:

$$(\widehat{\mu}, \widehat{\nu}, \widehat{\lambda}) = \underset{(\mu,\nu,\lambda)\in\mathbb{R}\times\mathbb{R}\times\mathbb{R}^+}{\operatorname{argmin}} TV(MG(\mu, \nu, \lambda)) \tag{9}$$

Doing so, we will favor homogeneous zones in terms of both albedo and normals, but still allowing edges, so that we do not prevent edges in the albedo map or in the normal field (which would lead to smooth shapes).

The following calculation gives a hint of it, and shows that minimizing $TV(M)$ is linked to minimizing simultaneously $TV(\rho)$ and $TV(N)$:

$$\begin{aligned}
\|J(M)\|_F &= \|J(\rho N)\|_F && (10)\\
&= \|N^T J(\rho) + \rho J(N)\|_F && (11)\\
&= \|\left(\nabla\rho\, N\right)^\top + \rho J(N)\|_F && (12)\\
&\leqslant \|\left(\nabla\rho\, N\right)^\top\|_F + \|\rho J(N)\|_F && (13)\\
&= \|\nabla\rho\|_2 + \rho\,\|J(N)\|_F && (14)\\
&\leqslant \|\nabla\rho\|_2 + \|J(N)\|_F && (15)
\end{aligned}$$

The equality (14) comes on one hand from a direct calculation using the fact that N has norm 1 and on the other hand from the fact that ρ is positive. The last inequality comes from the fact that $\rho \in [0, 1]$. From this, it follows that

$$TV(M) \leq TV(\rho) + TV(N). \tag{16}$$

Note also that $\int_{\mathbb{R}^2} \rho \|J(N)\|_F$ is the *ρ-weighted* total variation of N and when minimizing it, it allows for "relaxing" the minimization of $TV(N)$ where the albedo is low, i.e, where the material is dark. As it is obvious that an albedo equal to zero induces an ill-posedness in normals, this "relaxation" allows us not to consider areas which would induce errors in the reconstruction.

5 Our Model

5.1 Formal Definition

One can now rewrite the full problem of uncalibrated (Lambertian) photometric stereo as the estimation of the field M, the lighting matrix S and the three GBR parameters μ, ν and λ. We can write this estimation problem as the constrained regularized optimization problem of Eq. 17:

$$\begin{cases} (\widehat{M}, \widehat{S}, \widehat{\mu}, \widehat{\nu}, \widehat{\lambda}) = \underset{(M,S,\mu,\nu,\lambda)}{\operatorname{argmin}} \sum_{p\in\Omega} \|I_p - M_p S\|^2 \\ \qquad\qquad\qquad\qquad\qquad\qquad +\alpha\, TV(MG(\mu,\nu,\lambda)) \\ \quad \text{s.t.} \quad \overline{\operatorname{curl}}(M) = 0 \end{cases} \tag{17}$$

- The loss function is here the l_2-norm between the data and the images produced by our estimated field and lightings, which is the reprojection error. This allows a Gaussian random noise on the input data.
- The penalty function is the TV-regularization of the estimated field transformed by a GBR, which as we explained in the previous section will allow us to estimate the "optimal" parameters μ, ν and λ.
- The constraint is the integrability constraint, which can be rewritten as $\frac{\partial}{\partial y}(\frac{M_x(p)}{M_z(p)}) = \frac{\partial}{\partial x}(\frac{M_y(p)}{M_z(p)})$ for any pixel $p \in \Omega$.
- α is the weight between the loss function and the penalty function.

To solve the problem in 17 which looks very much like the Rudin-Osher-Fatemi model [23] apart from the transformation G, one could think separating the problem into two optimization problems. One of them would be a simple least-square problem, the other looks very much like a total variation based zooming problem like in [20], which could be solved by some kind of primal-dual scheme like [6]. But, although this looks really nice, our tests have shown to be quite disappointing: finding the right α is really painful as wrong choices of it will either produce an over-smoothed field or let the GBR unsolved.

We propose another approach which is actually much simpler as it does not involve such a hyper-parameter. We replace 17 by the following pair of minimization problems, which must be solved sequentially:

$$\begin{cases} (\widehat{M}, \widehat{S}) = \underset{(M,S)}{\operatorname{argmin}} \sum_{p\in\Omega} \|I_p - M_p S\|^2 \\ \quad \text{s.t.} \quad \overline{\operatorname{curl}}(M) = 0 \\ \\ (\widehat{\mu}, \widehat{\nu}, \widehat{\lambda}) = \underset{(\mu,\nu,\lambda)}{\operatorname{argmin}} TV(\widehat{M}G(\mu,\nu,\lambda)) \end{cases} \tag{18}$$

After the optimization process, the resulting field is given by $M = \widehat{M}G(\widehat{\mu}, \widehat{\nu}, \widehat{\lambda})$, and the resulting lighting matrix by $S = G^{-1}(\widehat{\mu}, \widehat{\nu}, \widehat{\lambda})\widehat{S}$.

5.2 The Full 3-Step Solution

To solve the optimization problem in 18, we adopt the common method used by (among others) Alldrin *et al.* in [3] or by Favaro and Papadhimitri in [10]:

1. First find a pair (M, S) which minimizes the loss function. As this is a l_2-norm, this can be achieved using SVD as proposed by Hayakawa in [14] (see Sec. 3). This gives $U \in \mathbb{R}^{|\Omega|\times 3}$, $W \in \mathbb{R}^{3\times 3}$ and $V \in \mathbb{R}^{m\times 3}$, so that $I \approx UWV^T$.In the end we get $M = UP^T, S = QV^T$, where P and Q are unknown 3×3 matrices holding $P^TQ = W$.
2. Then restrict P by integrability. In some sense we "project" M on the space of integrable fields to force the estimated field to respect the constraint. This is done identifying 6 out of the 9 cofactors of P (*i.e.* the coefficients of P^{-1}), as explained by Yuille and Snow in [30]. We fix the remaining 3 cofactors to empirical values, assuming they will be "corrected" in the next step.
   We know that this linear transformation will not affect the loss function, so up to this point we have solved the first part of 18.
3. Finally solve for the GBR, to estimate the three parameters μ, ν and λ that minimize $TV(\widehat{M}G(\mu, \nu, \lambda))$, where $\widehat{M} = UP^T$. This is achieved using some standard convex optimization method. In our tests we used the "fminunc" function of Matlab which is basically a variant of a Gauss-Newton algorithm, specifying a literal expression for the gradient which can be easily obtained by differentiating the penalty function with respect to each parameter.
   As the GBR is a linear transformation, the loss function will not be affected, and we know a GBR maintains integrability, so the constraint is not violated. Then both the loss function and the penalty function are minimized, and the constraint is respected. Thus, we can finally compute $\widehat{M} = UP^TG(\widehat{\mu}, \widehat{\nu}, \widehat{\lambda})$ and $\widehat{S} = G^{-1}(\widehat{\mu}, \widehat{\nu}, \widehat{\lambda})P^{-T}WV^T$ which are our field and lighting solutions.

5.3 Initialization Issues

The total variation being convex but not strictly convex, the starting point for the gradient descent has to be carefully chosen. In the step 3 of the algorithm above, we initialize the optimization process with the solution to the l_2-regularized problem with $\lambda = 1$:

$$(\widehat{\mu}_0, \widehat{\nu}_0) = \underset{(\mu,\nu)\in\mathbb{R}\times\mathbb{R}}{\operatorname{argmin}} \sum_{p\in\Omega} \|J(M(p)G(\mu, \nu, 1))\|_F^2 \tag{19}$$

which, writing down the Euler-Lagrange equations, gives:

$$\widehat{\mu}_0 = \sum_{p\in\Omega} \frac{\frac{\partial M_x}{\partial x}(p)\frac{\partial M_z}{\partial x}(p) + \frac{\partial M_x}{\partial y}(p)\frac{\partial M_z}{\partial y}(p)}{(\frac{\partial M_z}{\partial x}(p))^2 + (\frac{\partial M_z}{\partial y}(p))^2} \quad \widehat{\nu}_0 = \sum_{p\in\Omega} \frac{\frac{\partial M_y}{\partial x}(p)\frac{\partial M_z}{\partial x}(p) + \frac{\partial M_y}{\partial y}(p)\frac{\partial M_z}{\partial y}(p)}{(\frac{\partial M_z}{\partial x}(p))^2 + (\frac{\partial M_z}{\partial y}(p))^2} \tag{20}$$

which depends only on the already computed $M = UP^T$ field, and thus can be computed directly.

Problem 9 can then be solved efficiently by a few Gauss-Newton iterations (less than 10 iterations in every test we ran).

Note that we do not solve this TV problem the usual way: one would expect to use some algorithm like Chambolle's [6] to recover the whole field. Here we do not need doing so because we assume we already have a field M, and we only want to estimate the 3 parameters of the GBR which transform this field. It is a much easier 3-parameters problem which can be solved by standard optimization methods.

6 Experiments

In this section, we give some results obtained with the method described before. As dealing with outliers like shadows or highlights is not our point in this paper, we just skip the preprocessing part. We use preprocessed data that can be found on Papadhimitri's and Alldrin's homepages, five set of images from the Yale Dataface B [13] and synthetic images generated with the Lambertian model.

One image of each set is presented in Fig. 2. Note that the Sphere and Teapot dataset contain 8 images, Cat, Owl and Horse datasets contain 12 images, the Doll dataset contains 15 images, and the Faces datasets contain 21 images each.

Fig. 2. One image of each dataset used for validating our method

We compare in Fig. 3 the recovered normals and depth maps on the Cat dataset to both a "ground truth" which is the result from calibrated photometric stereo, and to the results from [10]. Note how close the three results are. Some rendered images can be found in Fig. 4 and Fig. 5.

We also show in Table 1 a comparison of the angular errors (expressed in degrees) between normal fields estimated by uncalibrated photometric stereo techniques and by calibrated photometric stereo. We chose to compare our method to the Diffuse-Maxima method [10] which seems to be the most efficient known method, and to Minimum-Entropy [3] because our approach can be seen as an

extension of this one. Our method appears to be as good as the most state-of-the-art uncalibrated photometric stereo method for Lambertian models, and slightly faster, even though the tests were made without any special care in the optimization task, so the method could still be greatly accelerated.

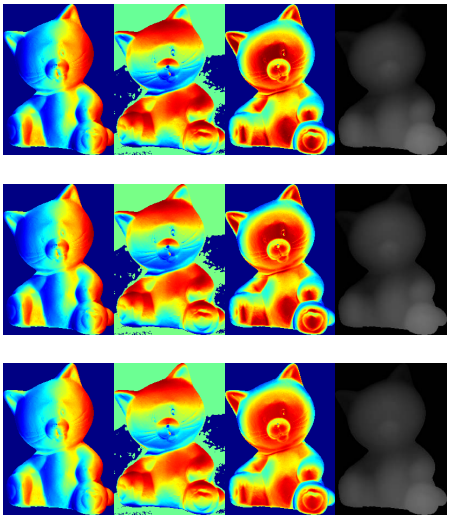


Fig. 3. Normal and depth comparison. From left to right: first, second, and third components of the normal, and height map. First row: calibrated photometric stereo. Second row: diffuse maxima [10]. Third row: our method.

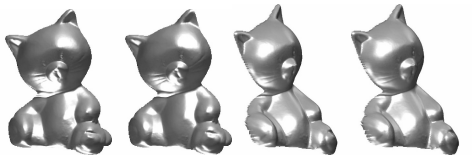


Fig. 4. Cat reconstruction: (a,c) calibrated PS (frontal and lateral views); (b,d) our method (frontal and lateral views)

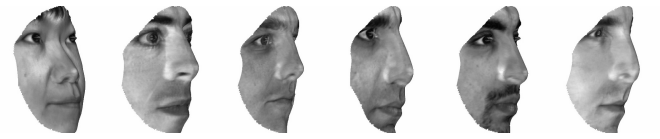


Fig. 5. Faces reconstructions (lateral views), with estimated albedo mapped as texture

Table 1. Performance comparison with the Minimum-Entropy method (ME) [3] and with the Diffuse-Maxima (DM) method [10]. We give the mean angular error (expressed in degrees) of the estimated normal field and the CPU time.

Dataset	Sphere	Teapot	Cat	Doll	Owl	Horse
ME	**5.64** (19.22 *s*)	24.01 (27.83 *s*)	13.91 (20.82*s*)	27.67 (15.18 *s*)	16.16 (25.15 *s*)	11.10 (15.80 *s*)
DM	6.84 (1.59 *s*)	23.94 (2.09 *s*)	5.37 (1.22 *s*)	12.15 (1.11 *s*)	**6.63** (1.07 *s*)	**4.80** (1.62 *s*)
TV	6.29 (1.40 *s*)	**16.48** (1.41 *s*)	**5.26** (0.67 *s*)	**11.90** (0.69 *s*)	7.06 (0.76 *s*)	5.53 (0.67 *s*)

Dataset	YaleB05_P00	YaleB06_P00	YaleB07_P00	YaleB08_P00	YaleB09_P00	YaleB10_P00
ME	9.44 (14.90 *s*)	18.00 (14.57 *s*)	12.63 (14.62 *s*)	19.61 (14.79 *s*)	17.10 (14.45 *s*)	9.22 (13.72 *s*)
DM	**6.90** (2.07 *s*)	11.46 (1.98 *s*)	**8.59** (1.84 *s*)	11.49 (1.29 *s*)	13.09 (1.11 *s*)	10.21 (3.63 *s*)
TV	13.33 (0.59 *s*)	**7.94** (0.50 *s*)	10.25 (0.55 *s*)	**8.80** (0.52 *s*)	**7.94** (0.51 *s*)	**8.34** (0.56 *s*)

7 Conclusion

In this paper, we presented a novel approach to the resolution of the generalized bas-relief ambiguity, introducing the total variation of the estimated field. We showed how this approach could produce a new model for uncalibrated photometric stereo and gave a very simple and efficient algorithm for finding a solution to this problem. Unlike most attempts to solve this issue, our method does not rely on any property of the scene except its Lambertian reflectance, so the method should work for any Lambertian dataset. For a real world application though, it would be necessary to make a preprocessing step on the data in order to remove the outliers.

We also compared the solution given by our algorithm to the most efficient known method, and the results show that our method performs as good as it. The key point of it is the optimization step of the algorithm, which could be greatly improved in order to get a really fast solver. This would open the doors to some real-time reconstruction in the wild using photometric stereo, and could be useful for many applications like augmented reality.

References

1. Abrams, A., Hawley, C., Pless, R.: Heliometric Stereo: Shape from Sun Position. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 357–370. Springer, Heidelberg (2012)
2. Ackermann, J., Langguth, F., Fuhrmann, S., Goesele, M.: Photometric Stereo for Outdoor Webcams. In: CVPR, pp. 262–269 (2012)
3. Alldrin, N.G., Mallick, S.P., Kriegman, D.J.: Resolving the Generalized Bas-relief Ambiguity by Entropy Minimization. In: CVPR (2007)
4. Belhumeur, P.N., Kriegman, D.J., Yuille, A.L.: The Bas-Relief Ambiguity. IJCV 35(1), 33–44 (1999)
5. Bresson, X., Chan, T.: Fast Dual Minimization of the Vectorial Total Variation Norm and Applications to Color Image Processing. Inverse Problems and Imaging 2(4), 455–484 (2008)
6. Chambolle, A.: An Algorithm for Total Variation Minimization and Applications. JMIV 20(1), 89–97 (2004)
7. Chandraker, M.K., Kahl, F., Kriegman, D.J.: Reflections on the Generalized Bas-Relief Ambiguity. In: CVPR, vol. I, pp. 788–795 (2005)

8. Drbohlav, O., Chantler, M.: Can Two Specular Pixels Calibrate Photometric Stereo?. In: ICCV, vol. II, pp. 1850–1857 (2005)
9. Durou, J.-D., Aujol, J.-F., Courteille, F.: Integrating the Normal Field of a Surface in the Presence of Discontinuities. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 261–273. Springer, Heidelberg (2009)
10. Favaro, P., Papadhimitri, T.: A Closed-Form Solution to Uncalibrated Photometric Stereo via Diffuse Maxima. In: CVPR, pp. 821–828 (2012)
11. Frankot, R.T., Chellappa, R.: A Method for Enforcing Integrability in Shape from Shading Algorithms. PAMI 10(4), 439–451 (1988)
12. Georghiades, A.S.: Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo. In: ICCV, vol. II, pp. 816–823 (2003)
13. Georghiades, A.S., Kriegman, D.J., Belhumeur, P.N.: From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. PAMI 23(6), 643–660 (2001)
14. Hayakawa, H.: Photometric stereo under a light-source with arbitrary motion. JOSA A 11(11), 3079–3089 (1994)
15. Hernández, C., Vogiatzis, G., Brostow, G.J., Stenger, B., Cipolla, R.: Multiview Photometric Stereo. PAMI 30(3), 548–554 (2008)
16. Hertzmann, A., Seitz, S.M.: Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs. PAMI 27(8), 1254–1264 (2005)
17. Horn, B.K.P.: Obtaining Shape from Shading Information. In: Shape from Shading, pp. 123–171. MIT Press (1989)
18. Horn, B.K.P.: Height and Gradient from Shading. IJCV 5(1), 37–75 (1990)
19. Lin, Z., Chen, M., Ma, Y.: The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-rank Matrices. Tech. Rep. UILU-ENG-09-2215, UIUC (2009)
20. Malgouyres, F., Guichard, F.: Edge Direction Preserving Image Zooming: A Mathematical and Numerical Analysis. SIAM Num. Anal. 39(1), 1–37 (2001)
21. Nikolova, M.: Minimizers of Cost-functions Involving Non-smooth Data-fidelity Terms. Application to the Processing of Outliers. SIAM Num. Anal. 40(3), 965–994 (2002)
22. Onn, R., Bruckstein, A.M.: Integrability Disambiguates Surface Recovery in Two-Image Photometric Stereo. IJCV 5(1), 105–113 (1990)
23. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear Total Variation Based Noise Removal Algorithms. Physica D: Nonlin. Phen. 60(1-4), 259–268 (1992)
24. Shi, B., Matsushita, Y., Wei, Y., Xu, C., Tan, P.: Self-calibrating Photometric Stereo. In: CVPR (2010)
25. Strong, D., Chan, T.: Edge-preserving and Scale-dependent Properties of Total Variation Regularization. Inv. Probl. 187, S165–S187 (2003)
26. Sunkavalli, K., Zickler, T., Pfister, H.: Visibility Subspaces: Uncalibrated Photometric Stereo with Shadows. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 251–264. Springer, Heidelberg (2010)
27. Tan, P., Mallick, S.P., Quan, L., Kriegman, D.J., Zickler, T.: Isotropy, reciprocity and the generalized bas-relief ambiguity. In: CVPR (2007)
28. Woodham, R.J.: Photometric Method for Determining Surface Orientation from Multiple Images. Opt. Engin. 19(1), 139–144 (1980)
29. Wu, L., Ganesh, A., Shi, B., Matsushita, Y., Wang, Y., Ma, Y.: Robust Photometric Stereo via Low-Rank Matrix Completion and Recovery. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part III. LNCS, vol. 6494, pp. 703–717. Springer, Heidelberg (2011)
30. Yuille, A.L., Snow, D.: Shape and Albedo from Multiple Images using Integrability. In: CVPR, pp. 158–164 (1997)

Minimizing TGV-Based Variational Models with Non-convex Data Terms*

Rene Ranftl, Thomas Pock, and Horst Bischof

Institute for Computer Graphics and Vision
Graz University of Technology
{ranftl,pock,bischof}@icg.tugraz.at

Abstract. We introduce a method to approximately minimize variational models with Total Generalized Variation regularization (TGV) and non-convex data terms. Our approach is based on a decomposition of the functional into two subproblems, which can be both solved globally optimal. Based on this decomposition we derive an iterative algorithm for the approximate minimization of the original non-convex problem. We apply the proposed algorithm to a state-of-the-art stereo model that was previously solved using coarse-to-fine warping, where we are able to show significant improvements in terms of accuracy.

Keywords: Total Generalized Variation, Optimization, Stereo.

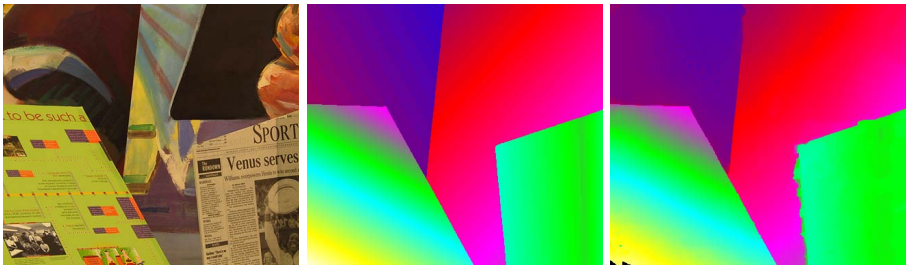
1 Introduction

Total Generalized Variation (TGV) [1], a generalization of the Total Variation (TV) regularization, has recently been successfully applied to a number of problems, like Optical Flow [2], Stereo [3] and Image Fusion [4]. Especially for Stereo and Optical Flow, TGV is arguably a better prior than the classical TV prior. For example in the second-order case, TGV does not penalize piecewise affine solutions. Such assumptions on planarity of the scene are frequently made in stereo matching (e.g. [5,6,7]) and also find application in optical flow estimation (e.g. [2,8]).

However, TGV regularization currently is restricted to convex functionals (i.e. convex data terms). If the functional is non-convex, as it is the case in stereo matching, one has to rely on convex approximations to the non-convex problem, which often decreases the performance of the model. This is not the case for TV, where global solutions can be computed even in the presence of non-convex data terms, provided that the continuous label-space is discretized and some natural ordering can be imposed onto the resulting discrete label space [9]. The idea of this approach is to lift the functional to a higher dimensional space, where the resulting functional is convex. Similar results were shown by Ishikawa [10] for discrete first-order Markov Random Fields (MRF). The lifting approach [9] was later extended to a broader class of convex first-order priors such as Quadratic and Huber regularization [11].

* This work was supported by the Austrian Science Fund (project no. P22492).

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 282–293, 2013.



(a) *Venus* (b) Groundtruth (c) Proposed

Fig. 1. Example from the Middlebury stereo dataset

Previous work on Stereo and Optical Flow that used TGV regularization [2,3] relied on the classical coarse-to-fine warping scheme [12] to approximately solve the original non-convex problem. The basic idea of this approach is to solve a series of convex models that arise from linearizations of the non-convex data term. In order to capture large motion or disparity ranges, respectively, this procedure has to be embedded into a coarse-to-fine framework, which is known to suffer from loss of fine details.

In the context of discrete MRFs, planarity assumptions can be enforced using a second-order prior. The resulting models can be approximately solved using a move-making strategy: The multi-label problem is reduced to a series of binary subproblems (each deciding if a node retains its label or switches to a proposed label), where each subproblem can be solved partially optimal [5]. The outcome of this approach crucially depends on the quality of the proposals in each move. Moreover, each of the subproblems is only solved partially optimal, which means that some nodes in the MRF may remain unlabeled.

Contribution. In this work we show how approximate solutions to non-convex functionals with TGV regularization can be computed. Our approach does not suffer from loss of fine details like the coarse-to-fine approaches do. The framework builds on the observation that functionals with TGV regularization and non-convex data terms can be split in two subproblems, where one is convex and the other, although non-convex, falls into the class of functionals covered by the lifting procedure described in [11] and can therefore be solved globally optimal.

In contrast to [5], where in each iteration a binary labeling problem, defined on a second-order energy, is solved, our approach solves a first-order multi-label problem in each iteration, in order to minimize the full second-order energy. This frees us from the need to specify proposals and also guarantees a complete labeling. Our splitting approach is similar to Alternating Convex Search [13], which itself falls under the broader class of Block-Relaxation methods [14].

We apply the proposed algorithm to a variational stereo model [3], which was solved using a coarse-to-fine strategy in the original formulation. By switching the optimization strategy to the herein proposed method, we are able to show

significant improvements in terms of accuracy. An exemplary result of the proposed method is shown in Figure 1. This is an example where the scene consists only of planes, which is perfectly modelled by the prior. Consequently we are able to recover high-quality disparity maps. Our evaluation shows that we obtain state-of-the-art results on the challenging KITTI stereo benchmark [15] as well as the Middlebury high-resolution benchmark [16].

2 Alternating Optimization

We focus on models with second-order TGV regularization, as this is the most widely used and also the simplest instance of TGV (besides TV), i.e. we consider functionals of the form

$$\min_{u,w} \overbrace{\alpha \int_\Omega |Dw|_\Gamma + \underbrace{\int_\Omega |Du - w|_\Sigma}_{E_1(w|u)} + \lambda \int_\Omega \rho(u)}^{E_2(u|w)}, \tag{1}$$

where $u : \Omega \to \mathbb{R}$ and $w : \Omega \to \mathbb{R}^2$, D is the distributional derivative, which is also well defined for discontinuous functionals, and the norms are defined as $|x|_M = \langle x, Mx \rangle^{\frac{1}{2}}$, M symmetric and positive definite. The introduction of the operator M will later allow us to easily incorporate anisotropic edge-weighted diffusion into the model. Note that for $\Gamma = I$ and $\Sigma = I$, the definition reduces to the standard definition of second-order TGV [1]. We will assume throughout the rest of this paper that the data term $\rho(u)$ is non-convex. Note that an extension of this basic formulation to higher-order instances of TGV is straight-forward, as it only involves a modified version of subproblem $E_1(w|u)$.

Our main observation is as follows: It is possible to decompose problem (1) into the two subproblems $E_1(w|u)$ and $E_2(u|w)$. Let the pair (u^*, w^*) be a global minimizer of (1), then it is obvious that the relation

$$u^* = \arg\min_u E_2(u|w^*) \tag{S1}$$

$$w^* = \arg\min_w E_1(w|u^*) \tag{S2}$$

holds, i.e. given w^* it is possible to deduce u^* by solving a possibly simpler subproblem and vice versa. Note that (S1) is a non-convex problem, while (S2) is a convex problem, which is equivalent to a generalized vectorial TV-L1 denoising problem [17]. This observation points to an iterative scheme for finding approximate solutions to (1):

$$\begin{aligned} u^{n+1} &= \arg\min_u E_2(u|w^n) \\ w^{n+1} &= \arg\min_w E_1(w|u^{n+1}). \end{aligned} \tag{A1}$$

Note that by definition we have $E(u^n, w^n) \geq E(u^{n+1}, w^n) \geq E(u^{n+1}, w^{n+1})$ and $0 \leq E(u, w) < \infty$, $\forall (u, w)$, therefore the procedure will converge in the functional value, although not necessarily to a global optimum.

The update steps in (A1) already constitute the basic iterations of the proposed algorithm for optimizing (1). It remains to show how to solve the individual subproblems in each step.

2.1 Minimizing $E_2(u|w)$

The subproblem $E_2(u|w)$ is a non-convex variational problem with a non-convex data term and a convex regularization term. It was shown by Pock et al. [11] that problems with this special structure can be solved globally optimal using the framework of calibrations. The basic idea is to lift the problem to a higher-dimensional space, where a globally optimal solution to the original problem can be computed.

Let us first introduce the general framework: In order to find a minimizer u^* of functionals of the form

$$\min_u \int_\Omega f(x, u(x), Du), \tag{2}$$

we can solve the auxiliary problem

$$\min_{v \in \mathcal{C}} \sup_{\phi \in \mathcal{K}} \int_{\Omega \times \mathbb{R}} \phi \cdot Dv, \tag{3}$$

where the convex sets $\mathcal{C}$ and $\mathcal{K}$ are given by

$$\mathcal{C} = \left\{ v \in BV(\Omega \times \mathbb{R}; [0,1]) : \lim_{t \to -\infty} v(x,t) = 1, \quad \lim_{t \to \infty} v(x,t) = 0 \right\}$$

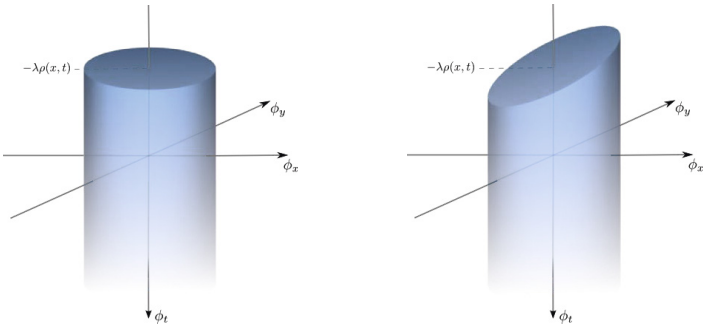
and

$$\begin{aligned} \mathcal{K} = \{ \phi = (\phi_x, \phi_t) \in\ & C_0(\Omega \times \mathbb{R}; \mathbb{R}^d \times \mathbb{R}) : \\ & \phi_t(x,t) \geq f^*(x,t,\phi_x(x,t)), \forall x,t \in \Omega \times \mathbb{R} \}. \end{aligned} \tag{4}$$

Here f^* denotes the convex conjugate of the function f. Note that the sets $\mathcal{C}$ and $\mathcal{K}$ are defined point-wise. The intuition behind this formulation is, that instead of minimizing u directly, one represents the energy in terms of characteristic functions of its upper level-sets v. Given a minimizer v^* the corresponding minimizer u^* can be recovered by $u^*(x) = \int_{\mathbb{R}} v^*(x,t)dt$.

This formulation is very general, the specific form of the convex regularization term only influences the set $\mathcal{K}$. Pock et al. [11] derived the set $\mathcal{K}$ for Quadratic, TV, Huber and Lipschitz regularization terms. In problem $E_2(u|w)$, the regularization term is similar to TV regularization, with the difference that a constant vector is subtracted from the gradient, before the absolute value is measured. We identify $f(x,t,p) = |p(x) - w^n(x)|_\Sigma + \lambda\rho(x,t)$, and consequently its convex conjugate with respect to p is

$$f^*(x,t,\phi) = \begin{cases} \langle \phi_x(x,t), w^n(x) \rangle - \lambda\rho(x,t), & \text{if} \quad |\phi_x(x,t)|_\Sigma \leq 1 \\ \infty, & \text{else.} \end{cases}$$



(a) TV (b) TGV

Fig. 2. The feasible set $\mathcal{K}$ for (a) TV and (b) TGV

The resulting set $\mathcal{K}$ is illustrated in Figure 2(b). The feasible set for TV regularization is shown in Figure 2(a). It can be seen that for problem $E_2(u|w)$ the feasible set is slightly more complicated than in the TV case. While for TV the set is given by the interior of a cylinder with radius 1, which is bounded from below by a vertical plane centered at $(0, 0, -\lambda\rho(x,t))^T$, the set in the TGV case is bounded from below by plane that includes the point $(0, 0, -\lambda\rho(x,t))^T$ but can be arbitrarily oriented (in fact the normal of this plane is given by $[w^n, -1]^T$). This makes projection onto this set slightly harder, as a closed-form solution is no longer available.

Discretization and Optimization. In order to solve (3) it is necessary to discretize the domain $\Omega \times \mathbb{R}$ of the continuous functions v and ρ. For the sake of simplicity let us only consider the case $\Omega \subset \mathbb{R} \times \mathbb{R}$, higher-dimensional cases can be derived analogously.

We discretize on a three-dimensional grid of size $N_x \times N_y \times N_t$ with discretization steps Δx, Δy and Δt:

$$G^\Delta = \{(i\Delta x, j\Delta y, k\Delta t) : (0,0,0) \leq (i,j,k) < (N_x, N_y, N_t)\}\,. \tag{5}$$

Here the triple (i, j, k) denotes the location in the grid.

For numerical reasons we replace the vector field ϕ_x, with a rotated version $\Sigma^{\frac{1}{2}}\phi_x$, which leads to a simplification of the convex set $\mathcal{K}^\Delta$, without changing the formulation.

The feasible sets for the discrete version of (3) are then given by

$$\mathcal{C}^\Delta = \left\{v^\Delta \in [0,1]^{N_x N_y N_t} : v^\Delta_{i,j,0} = 1, v^\Delta_{i,j,N_t-1} = 0\right\} \tag{6}$$

and

$$\begin{aligned}\mathcal{K}^\Delta = \big\{\phi^\Delta = (\phi^\Delta_x, \phi^\Delta_y, \phi^\Delta_t) \in \mathbb{R}^{3N_xN_yN_t} :& \\ (\phi^\Delta_t)_{i,j,k} + \lambda(\rho)_{i,j,k} &\geq \left\langle (\phi^\Delta_x, \phi^\Delta_y)^T_{i,j,k}, \Sigma^{\frac{1}{2}} w_{i,j} \right\rangle, \\ |(\phi^\Delta_x, \phi^\Delta_y)^T_{i,j,k}|_2 &\leq 1, \quad \forall (i,j,k) \in G^\Delta\big\}\,.\end{aligned} \tag{7}$$

In order to discretize the differential operator D, we use forward differences with Neumann boundary conditions. Furthermore we allow Σ to vary locally, which allows us to incorporate image-driven TGV regularization similar to [3] into the framework, i.e. we define a linear operator $\nabla_\Sigma : \mathbb{R}^{N_x N_y N_t} \to \mathbb{R}^{3N_x N_y N_t}$, with

$$(\nabla_\Sigma v^\Delta)_{i,j,k} = \begin{pmatrix} \Sigma_{i,j}^{\frac{1}{2}} & 0 \\ 0\ 0 & 1 \end{pmatrix} \begin{pmatrix} (\delta_x v^\Delta)_{i,j,k} \\ (\delta_y v^\Delta)_{i,j,k} \\ (\delta_t v^\Delta)_{i,j,k} \end{pmatrix} \tag{8}$$

and

$$(\delta_x v^\Delta)_{i,j,k} = \begin{cases} (v^\Delta_{i+1,j,k} - v^\Delta_{i,j,k})/\Delta x & \text{if} \quad i < N_x - 1 \\ 0 & \text{else} \end{cases} \tag{9}$$

$$(\delta_y v^\Delta)_{i,j,k} = \begin{cases} (v^\Delta_{i,j+1,k} - v^\Delta_{i,j,k})/\Delta y & \text{if} \quad i < N_y - 1 \\ 0 & \text{else} \end{cases} \tag{10}$$

$$(\delta_t v^\Delta)_{i,j,k} = \begin{cases} (v^\Delta_{i,j,k+1} - v^\Delta_{i,j,k})/\Delta t & \text{if} \quad i < N_t - 1 \\ 0 & \text{else.} \end{cases} \tag{11}$$

Note that (8) reduces to the standard discretization of a gradient operator, if $\Sigma_{i,j}^{\frac{1}{2}}$ is set to identity everywhere. On the other hand, it is possible to incorporate image-driven diffusion into the model by setting the matrix appropriately. We will later discuss the specific choice of this matrix.

The discrete version of (3) is now given by

$$\min_{v^\Delta \in \mathcal{C}^\Delta} \max_{\phi^\Delta \in \mathcal{K}^\Delta} \left\langle \nabla_\Sigma v^\Delta, \phi^\Delta \right\rangle \tag{12}$$

For optimization of the convex-concave saddle-point problem (12) we use the primal-dual algorithm [18]. The iterations of this algorithm are shown in Algorithm 1.

A crucial part of this algorithm are the pointwise projections $\mathrm{Proj}_{\mathcal{K}^\Delta}(.)$ and $\mathrm{Proj}_{\mathcal{C}^\Delta}(.)$ respectively. The projection of the primal variables is simple and can be carried out in closed-form:

$$(\mathrm{Proj}_{\mathcal{C}^\Delta}(\hat{v}))_{i,j,k} = \begin{cases} \max\{0, \min\{1, \hat{v}_{i,j,k}\}\} & \text{if } k > 1 \\ 1 & \text{else.} \end{cases}$$

Algorithm 1. Primal-dual algorithm for solving (12)

1. *Initialize*
   Set $(v^\Delta)^0 \in \mathcal{C}^\Delta$, $(\phi^\Delta)^0 \in \mathcal{K}^\Delta$, $(\bar{v})^0 = (v^\Delta)^0$, $n = 0$
   Choose time-steps $\tau, \sigma > 0$, $\tau\sigma < \frac{1}{\|\nabla_\Sigma\|^2}$
2. *Iterate*

$$\begin{cases} (\phi^\Delta)^{n+1} & \leftarrow \mathrm{Proj}_{\mathcal{K}^\Delta}((\phi^\Delta)^n + \sigma(\nabla_\Sigma \bar{v}^n)) \\ (v^\Delta)^{n+1} & \leftarrow \mathrm{Proj}_{\mathcal{C}^\Delta}((v^\Delta)^n - \tau(\nabla_\Sigma^T(\phi^\Delta)^{n+1})) \\ \bar{v}^{n+1} & \leftarrow 2(v^\Delta)^{n+1} - (v^\Delta)^n \end{cases}$$

The projections for the dual variables $\text{Proj}_{\mathcal{K}^\Delta}(.)$, although also point-wise, are more complicated. The feasible set $\mathcal{K}^\Delta$ is defined point-wise via the intersection of two convex sets. We experimented with different variants to incorporate these constraints: Lagrange multipliers, solving the projection problem in each iteration of the primal-dual algorithm using FISTA [19] (including a preconditioned variant) and finally Dykstra's Projection algorithm [20]. Our experiments show that Dykstra's algorithm provides the best performance for this type of problem and is very light-weight, we therefore resort to this variant to incorporate the dual constraints. The iterations of Dykstra's algorithm are shown in Algorithm 2, where we set $n = [w^n_{i,j,k}, -1]^T$ and $c = \lambda\rho_{i,j,k}$. In practice we run the algorithm until the distances to both convex sets ($|x^n - y^n|_2$ and $|y^n - x^{n+1}|_2$) are below a tolerance of 10^{-3} (which is typically achieved in under 10 iterations).

Algorithm 2. Algorithm for projecting onto the set $\mathcal{K}$

1. *Initialize*
   Set $n = 0$, $x^0 = \phi_{i,j,k}$, $p^0 = 0$, $q^0 = 0$
2. *Iterate*

$$\begin{cases} y^n & \leftarrow \frac{(x^n + p^n)}{\max\{1, |x^n + p^n|_2\}} \\ p^{n+1} & \leftarrow p^n + x^n - y^n \\ x^{n+1} & \leftarrow \begin{cases} y^n + q^n & \text{if} \quad \langle y^n + q^n, n\rangle \leq c \\ y^n + q^n - \frac{\langle y^n + q^n, n\rangle - c}{\langle n, n\rangle} n & \text{else} \end{cases} \\ q^{n+1} & \leftarrow q^n + y^n - x^{n+1} \end{cases}$$

2.2 Minimizing $E_1(w|u)$

The subproblem $E_1(w|u)$ is a non-smooth convex optimization problem, which can be solved using standard techniques. We will show how to cast this problem in a saddle-point formulation and again apply the primal-dual algorithm [18].

The optimization problem reads

$$\min_w \int_\Omega |Du^{n+1} - w|_\Sigma + \alpha \int_\Omega |Dw|_\Gamma, \tag{13}$$

where u^{n+1} is given by the last solution of problem $E_2(u|w)$. Note that this problem corresponds to denoising the gradients of u^{n+1}.

Using the definition $\text{div}_M z = \text{div}(M^{\frac{1}{2}} z)$, the equivalent saddle-point formulation is given by:

$$\min_w \sup_{\substack{\|p\|_\infty \leq 1 \\ \|q\|_\infty \leq 1}} -\int_\Omega u^{n+1} \,\text{div}_\Sigma\, p \,\text{dx} - \int_\Omega \left\langle w, \Sigma^{\frac{1}{2}} p + \alpha \,\text{div}_\Gamma\, q \right\rangle \text{dx}. \tag{14}$$

Discretization of (14) follows analogously to the lifted problem: The two-dimensional grid is given by

$$\hat{G}^\Delta = \{(i\Delta x, j\Delta y) : (0,0) \leq (i,j) < (N_x, N_y)\}, \tag{15}$$

Algorithm 3. Primal-dual algorithm for solving (17)

1. *Initialize*
   Set $(u^{\Delta})^0 = u^{n+1}, (w^{\Delta})^0 = \nabla_{\Sigma} u^{n+1}, (\bar{u})^0 = (u^{\Delta})^0$, $(\bar{w})^0 = (w^{\Delta})^0$
   Set $((p^{\Delta})^0)_{i,j} = (0,0)^T$, $(q^{\Delta})_{i,j} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $n = 0$
   Choose time-steps $\tau, \sigma > 0$, $\tau\sigma < \frac{1}{\|A\|^2}$, where $A = \begin{pmatrix} \nabla_{\Sigma} & -I \\ 0 & \mathcal{D}_{\Gamma} \end{pmatrix}$
2. *Iterate*

$$\begin{cases} (p^{\Delta})^{n+1} & \leftarrow \mathrm{Proj}_{\|p\|_{\infty} \leq 1}((p^{\Delta})^n + \sigma(\nabla_{\Sigma} \bar{u}^n - \mathrm{diag}(\Sigma_{i,j}^{\frac{1}{2}}) \bar{w}^n)) \\ (q^{\Delta})^{n+1} & \leftarrow \mathrm{Proj}_{\|q\|_{\infty} \leq 1}((q^{\Delta})^n + \sigma(\mathcal{D}_{\Gamma} \bar{w}^n)) \\ (u^{\Delta})^{n+1} & \leftarrow \mathrm{Proj}_{\mathcal{B}}((u^{\Delta})^n - \tau \nabla_{\Sigma}^T (p^{\Delta})^{n+1}) \\ (w^{\Delta})^{n+1} & \leftarrow (w^{\Delta})^n - \tau(\mathcal{D}_{\Gamma}^T (q^{\Delta})^{n+1} - \mathrm{diag}(\Sigma_{i,j}^{\frac{1}{2}})(p^{\Delta})^{n+1}) \\ \bar{u}^{n+1} & \leftarrow 2(u^{\Delta})^{n+1} - (u^{\Delta})^n, \quad \bar{w}^{n+1} \leftarrow 2(w^{\Delta})^{n+1} - (w^{\Delta})^n \end{cases}$$

where the tuple (i, j) again denotes a location in the grid, which also coincides with the spatial coordinates of the lifted problem. The discrete saddle-point problem can be written as

$$\min_{w^{\Delta}} \max_{\substack{\|p^{\Delta}\|_{\infty} \leq 1 \\ \|q^{\Delta}\|_{\infty} \leq 1}} \left\langle \nabla_{\Sigma} u^{n+1}, p^{\Delta} \right\rangle - \left\langle \mathrm{diag}(\Sigma^{\frac{1}{2}}) w^{\Delta}, p^{\Delta} \right\rangle + \alpha \left\langle \mathcal{D}_{\Gamma} w^{\Delta}, q^{\Delta} \right\rangle, \tag{16}$$

where the discrete differential operators ∇_{Σ} and $\mathcal{D}_{\Gamma}$ are again based on forward differences with Neumann boundary conditions, i.e. we have

$$(\nabla_{\Sigma} u^{\Delta})_{i,j} = \Sigma_{i,j}^{\frac{1}{2}} \begin{pmatrix} (\delta_x u^{\Delta})_{i,j} \\ (\delta_y u^{\Delta})_{i,j} \end{pmatrix} \qquad (\mathcal{D}_{\Gamma} w^{\Delta})_{i,j} = \Gamma_{i,j}^{\frac{1}{2}} \begin{pmatrix} (\delta_x w_1^{\Delta})_{i,j} & (\delta_y w_2^{\Delta})_{i,j} \\ (\delta_y w_1^{\Delta})_{i,j} & (\delta_x w_2^{\Delta})_{i,j} \end{pmatrix}.$$

In practice direct usage of (13) for the estimation of the second-order part w may be problematic if the discretization step Δt for the solution of the lifted problem was chosen too coarsely. In this case discretization artifacts are propagated from the lifted problem to problem (13), which may deteriorate the estimation of the second-order part, since in the context of this subproblem such artifacts are merely additional edges.

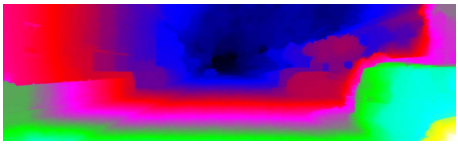
To cope with this problem, we modify (16) to allow u^{n+1} to slightly vary in a neighborhood of half the discretization step Δt of the lifted problem:

$$\min_{w^{\Delta}, u^{\Delta} \in \mathcal{B}} \max_{\substack{\|p^{\Delta}\|_{\infty} \leq 1 \\ \|q^{\Delta}\|_{\infty} \leq 1}} \left\langle \nabla_{\Sigma} u^{\Delta} - \mathrm{diag}(\Sigma_{i,j}^{\frac{1}{2}}) w^{\Delta}, p^{\Delta} \right\rangle + \alpha \left\langle \mathcal{D}_{\Gamma} w^{\Delta}, q^{\Delta} \right\rangle, \tag{17}$$

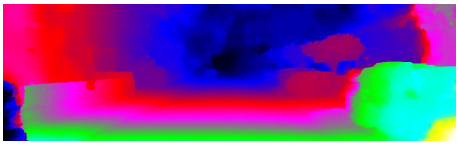
where $\mathcal{B} = \left\{ u^{\Delta} \in \mathbb{R}^{N_x N_y} : |(u^{\Delta})_{i,j} - (u^{n+1})_{i,j}| \leq \Delta t/2 \right\}$.

The iterations for optimizing (17) are shown in Algorithm 3. As before, we again have to perform projections onto convex sets in each iteration of the algorithm. The projections of the dual variables are given by $(\mathrm{Proj}_{\|r\|_{\infty} \leq 1}(r))_{i,j} = \frac{r_{i,j}}{\max\{1, |r_{i,j}|_2\}}$. For the primal variables u, the projection onto $\mathcal{B}$ can be computed by clamping $(u)_{i,j}$ in the interval $[(u^{n+1})_{i,j} - \frac{\Delta t}{2}, (u^{n+1})_{i,j} + \frac{\Delta t}{2}]$.

(a) Reference image



(b) Disparity - ITGV [3]



(c) Disparity - Proposed

(d) Error - ITGV [3]

(e) Error - Proposed

Fig. 3. Example from the KITTI benchmark for (b) ITGV [3] and (c) the proposed algorithm. The corresponding error maps are shown in (d) and (e). Occluded pixels are marked red in the error maps.

3 Application to Stereo

We show the effectiveness of our optimization approach on the application of stereo matching. We use the variational model that was proposed in [3] as basis for our experiments. This model introduced an image-driven TGV regularizer and based the matching term on the Census Transform. The original formulation used a warping procedure together with a coarse-to-fine scheme for the optimization. Such approaches are commonly used in variational stereo and optical flow, but are known to suffer from loss of detail due to the downsampling procedure.

Let us briefly explain, how the model [3] is realized in our framework: The matching term is based on the ternary Census transform [21]. We denote the ternary Census transform of the image I by $C(I)$. Then the matching cost for disparity t is given by the Hamming distance [22] between the ternary Census transforms of the warped matching image I_L and the reference image I_R, i.e.:

$$\hat{\rho}(x,t) = \Delta(C(I_L(x+[t,0]^T)), C(I_R(x))), \qquad \Delta(p,q) = \sum_{p_i \neq q_i} 1. \tag{18}$$

In order to cope with small calibration errors and to improve robustness with respect to the discretization, we employ a similar strategy to the Birchfeld-Tomasi dissimilarity measure [23], i.e. we sample the cost in a neighborhood of x and assign the minimum value as the final data term:

$$\rho(x,t) = \min\{(\hat{\rho}(x,t), \hat{\rho}(x+a,t), \hat{\rho}(x-a,t), \hat{\rho}(x+b,t), \hat{\rho}(x-b,t)\}, \tag{19}$$

where the offset vectors a and b are given by $a = [\frac{\Delta x}{2}, 0]^T$ and $b = [0, \frac{\Delta y}{2}]^T$.

Image-driven regularization can be realized by setting $\Gamma_{i,j}^{\frac{1}{2}} = I$ and $\Sigma_{i,j}^{\frac{1}{2}} = \exp(-\gamma|\nabla I_L|_{i,j}^{\beta})nn^T + n^{\perp}n^{\perp^T}$, where $n = (\frac{\nabla I_L}{|\nabla I_L|})_{i,j}$ and $\gamma, \beta > 0$.

Evaluation. We focus the evaluation on qualitative results on the task of stereo estimation, instead of direct comparisons of final energies. A meaningful and fair comparison in terms of final energies between our approach and the baseline [3] is hardly possible, since the results of the baseline heavily depend on the parameters of the coarse-to-fine strategy.

We first compare the proposed approach to the baseline algorithm using the KITTI stereo benchmark [15]. This benchmark consists of 195 test images and 194 training images captured from an automotive platform. Groundtruth data is given in the form of semi-dense disparity maps that were captured using a laser sensor. We used the groundtruth that is provided with the training images to tune the parameters of the model. For all experiments the discretization of the disparity range was fixed to $\Delta t = 1px$ and Algorithm (A1) was run for 10 iterations. The optimization of each subproblem was run for 2000 iterations.

Figure 3 shows an example from the test set and compares the proposed approach (Figure 3 (c)) to the baseline approach (Figure 3 (b)). We observe that the proposed approach preserves more fine details and is better at handling large disparities than the coarse-to-fine approach. This higher accuracy is also reflected in the average number of bad pixels on the test set (Table 1), where the proposed approach currently ranks second, while the baseline is ranked on the 7th place. Note, that the method shows a slightly worse inpainting capability, when compared to the baseline, in areas, where there is no overlap between the input images, which results in a slightly higher error if those regions are considered in the evaluation (Avg-All in Table 1).

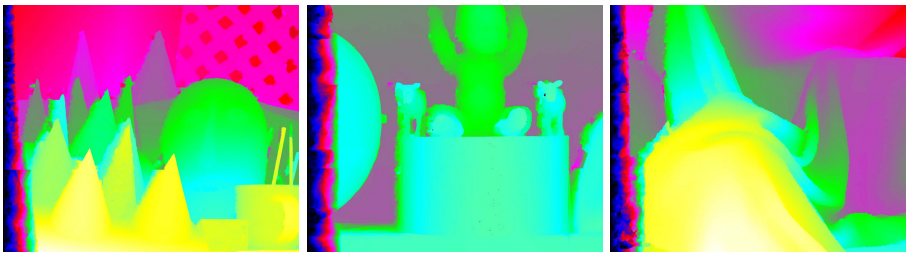
Our second evaluation uses a subset of 9 images from the Middlebury high-resolution benchmark [16] (*Teddy*, *Cones*, *Lamp2*, *Cloth3*, *Aloe*, *Art*, *Dolls*, *Baby3*,

Table 1. Results on the KITTI-Benchmark. Columns *Out-Noc* and *Out-All* show the average percentage of pixels with an error larger than 3px in non-occluded and all regions, respectively. Columns *Avg-Noc* and *Avg-All* show the mean absolute errors.

Rank	Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime
1	PCBP [7]	4.13 %	5.45 %	0.9 px	1.2 px	5 min
2	Proposed	5.05 %	6.91 %	1.0 px	1.6 px	6 min
3	iSGM [24]	5.16 %	7.19 %	1.2 px	2.1 px	8s
7	ITGV [3]	6.31 %	7.40 %	1.3 px	1.5 px	7s

Table 2. Error on the Middlebury high-resolution benchmark

Method	> 2 pixels	> 3 pixels	> 4 pixels	> 5 pixels
PCBP [7]	2.8 %	2.4 %	2.1 %	2.0 %
Proposed	4.4 %	3.1 %	2.5 %	2.2 %
ELAS [25]	4.7 %	3.9 %	3.5 %	3.2 %
OCV-SGBM [26]	5.9 %	5.5 %	5.3 %	5.2 %



(a) *Cones* (b) *Baby3* (c) *Cloth3*

Fig. 4. Example results from the Middlebury stereo benchmark

Rocks2). Exemplary results for this benchmark are shown in Figure 4. The average scores for different error thresholds and a comparison to state-of-the-art methods is shown in Table 2. We again observe that the proposed method is competitive to the other methods.

4 Conclusion

We presented an approach to approximately solve variational models with Total Generalized Variation regularization and non-convex data terms. Our approach alternates between solving a non-convex subproblem that can be solved globally optimal using functional lifting, and solving a convex subproblem.

We demonstrated the benefit of our approach on a variational stereo model that was previously solved using coarse-to-fine warping. Experiments on the challenging KITTI stereo benchmark show that this alternating minimization algorithm is able to significantly increase the performance of the model and consequently provides state-of-the-art results.

For future work, we plan to extend our approach to non-convex variants of TGV (i.e. truncated potentials). While we expect such regularization terms to be stronger priors and a splitting is in principle still possible, the problem is much harder to solve, because both of the resulting subproblems are non-convex.

References

1. Bredies, K., Kunisch, K., Pock, T.: Total Generalized Variation. SIAM J. Img. Sci. 3(3), 492–526 (2010)
2. Werlberger, M.: Convex Approaches for High Performance Video Processing. PhD thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria (2012)
3. Ranftl, R., Gehrig, S., Pock, T., Bischof, H.: Pushing the Limits of Stereo Using Variational Stereo Estimation. In: Proc. Intelligent Vehicles Symposium (2012)
4. Pock, T., Zebedin, L., Bischof, H.: TGV-Fusion. In: Calude, C.S., Rozenberg, G., Salomaa, A. (eds.) Maurer Festschrift. LNCS, vol. 6570, pp. 245–258. Springer, Heidelberg (2011)
5. Woodford, O., Torr, P., Reid, I., Fitzgibbon, A.: Global stereo reconstruction under second order smoothness priors. In: Proc. CVPR, pp. 1–8 (2008)
6. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch Stereo - Stereo Matching with Slanted Support Windows. In: Proc. BMVC, pp. 14.1–14.11 (2011)

7. Yamaguchi, K., Hazan, T., McAllester, D., Urtasun, R.: Continuous markov random fields for robust stereo estimation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 45–58. Springer, Heidelberg (2012)
8. Trobin, W., Pock, T., Cremers, D., Bischof, H.: An unbiased second-order prior for high-accuracy motion estimation. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 396–405. Springer, Heidelberg (2008)
9. 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)
10. Ishikawa, H.: Exact optimization for markov random fields with convex priors. PAMI 25, 1333–1336 (2003)
11. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: Global solutions of variational models with convex regularization. SIAM J. Img. Sci. 3(4), 1122–1145 (2010)
12. Brox, T., Bruhn, A., Papenberg, 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)
13. Gorski, J., Pfeuffer, F., Klamroth, K.: Biconvex sets and optimization with biconvex functions: a survey and extensions. Mathematical Methods of Operations Research 66, 373–407 (2007)
14. de Leeuw, J.: Block-relaxation algorithms in statistics. Technical report, Dept. of Statistics, UCLA (1994)
15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Proc. CVPR, pp. 3354–3361 (2012)
16. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV), pp. 131–140 (2001)
17. Nikolova, M.: A variational approach to remove outliers and impulse noise. JMIV 20(1-2), 99–120 (2004)
18. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. JMIV 40, 120–145 (2011)
19. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Img. Sci. 2(1), 183–202 (2009)
20. Boyle, J.P., Dykstra, R.L.: A method for finding projections onto the intersection of convex sets in Hilbert spaces. Lexture Notes in Statistics 37, 28–47 (1986)
21. Zabih, R., Ll, J.W.: Non-parametric local transforms for computing visual correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994)
22. Hamming, R.W.: Error detecting and error correcting codes. Bell System Technical Journal 29(2), 147–160 (1950)
23. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. PAMI 20(4), 401–406 (1998)
24. Hermann, S., Klette, R.: Iterative semi-global matching for robust driver assistance systems. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part III. LNCS, vol. 7726, pp. 465–478. Springer, Heidelberg (2013)
25. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 25–38. Springer, Heidelberg (2011)
26. Hirschmueller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Proc. CVPR, pp. 807–814 (2005)

A Mathematically Justified Algorithm for Shape from Texture

Helge Rhodin[1] and Michael Breuß[2]

[1] Graphics, Vision and Video Group,
Max-Planck-Institut für Informatik, Campus E1 4, 66123 Saarbruecken, Germany
hrhodin@mpi-inf.mpg.de
[2] Applied Mathematics and Computer Vision Group,
BTU Cottbus, Institute for Applied Mathematics and Scientific Computing,
Platz der Deutschen Einheit 1, 03046 Cottbus, Germany
breuss@tu-cottbus.de

Abstract. In this paper we propose a new continuous Shape from Texture (SfT) model for piecewise planar surfaces. It is based on the assumptions of texture homogeneity and perspective camera projection. We show that in this setting an unidirectional texture analysis suffices for performing SfT. With carefully chosen approximations and a separable representation, novel closed-form formulas for the surface orientation in terms of texture gradients are derived. On top of this model, we propose a SfT algorithm based on spatial derivatives of the dominant local spatial frequency in the source image. The method is motivated geometrically and it is justified rigorously by error estimates. The reliability of the algorithm is evaluated by synthetic and real world experiments.

Keywords: single view reconstruction, shape from texture, texture gradients, unidirectional texture analysis.

1 Introduction

Three-dimensional (3-D) shape reconstruction is a highly challenging task if only a single input image is available. The *Shape from Texture* (SfT) process is among the approaches that can be applied in this setting. Based on assumptions on the regularity of texture in the input image, SfT aims at reconstructing the 3-D shape of object surfaces by reversing the apparent distortion of the texture caused by the objects' geometry. An illustration of a characteristic reconstruction is shown in Figure 1. It is worth to note that SfT is complementary to other single-view methods, like e.g. shape from shading which does not consider texture information for reconstruction [1], and it is potentially very useful for the purpose of 3-D reconstruction since real-world images often incorporate texture.

Research in the domain of SfT has accomplished a solid theoretical foundation. However, existing SfT algorithms appear to be applicable only to settings with relatively specific model assumptions. Furthermore, the majority of existing algorithms are computationally expensive as they rely on costly texture analysis

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 294–305, 2013.

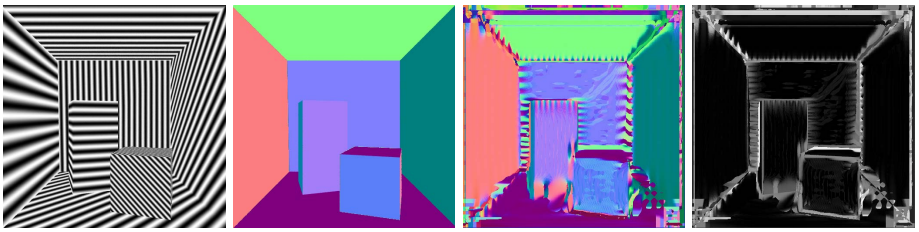


Fig. 1. a) Textured version of the Cornell box, **b)** its surface normal ground truth in standard colour code, **c)** dense normal reconstruction by our algorithm with sampling distance $w = 3$, and **d)** the angular surface normal error displayed by intensities in $[0, 1]$

methods such as wavelet transforms in multiple directions. In this paper we contribute in closing this gap between theory and application of SfT. Making use of a well-engineered combination of model components, we show that it is possible to keep the benefits of the developed theory, such as locality and precision, while defining a more efficient SfT algorithm with significantly reduced computational effort for texture analysis.

We advance the findings of Gårding [2] with respect to the relation between projective effects of geometry and texture gradients in order to enable a meaningful application of unidirectional texture analysis for SfT. Therefore, we propose a novel combination of model components: *(i)* A dedicated semi-perspective model for camera projection, *(ii)* a separable parametrization of surface slant, and *(iii)* linearisation for small sampling distances. The benefit of these choices is that they allow to express surface slant in a closed-form equation via unidirectional local texture measurements. We proceed in the algorithmical realization of SfT with a dense estimate of the surface slant by sampling the frequency content of the digital source image with one-dimensional Gabor wavelets along pixel rows or columns, respectively. It makes our new SfT algorithm very efficient compared to other methods in the field that either rely on two-dimensional texture analysis or one-dimensional texture analysis in multiple directions.

Our paper is structured as follows. After a discussion of the SfT methods in the field in Section 2 we proceed in Section 3 with the detailed derivation of our model. After proposing then our new SfT algorithm in Section 3.3 we discuss a number of experiments and give a conclusion in Sections 4 and 5, respectively.

2 Review of Shape from Texture Methods

A large group of SfT algorithms relies on the identification and extraction of repeating texture elements in a preprocessing step. During shape reconstruction either the individual instances are examined for projective distortions [3], or the global distribution of elements is measured and related to the apparent surface shape [4]. Also joined approaches have been proposed [5]. Unavoidably, the selective texture element extraction process discards information before the actual

shape reconstruction. The alternative is to apply continuous-scale, potentially lossless filters such as transformations in the frequency domain [6, 7].

Independently of the texture model, SfT algorithms have to be restrained to a restricted class of regular textures, in order to make the reconstruction unambiguous and meaningful. In particular textures that fake perspective effects such as perspective drawings have to be excluded. In the SfT literature texture *isotropy* and *homogeneity*, i.e. the directional and spatial uniformity of texture characteristics, are utilised for this purpose. For instance, Witkin estimates the degree of texture isotropy of line drawings by the distributions of apparent line-tangents [4]. Galasso and Lasenby measure homogeneity by combining Fourier and wavelet filters [6] and Clerc and Mallat utilise a probabilistic interpretation [7]. Wavelet-like filters are frequently employed due to practical reasons and their consistency with visual coding in the early stages of human perception [8].

Explicit back projection is one way to relate the measured degree of isotropy and deviation from homogeneity to the underlying surface shape. In this approach, the object shape is inferred by searching for the most regular back projection of the observed image onto a set virtual surfaces, like planes [6] or low-dimensional parametrised curves [9]. Alternatively, implicit shape representations leading to the task to solve linear [7] or non-linear equations [10] have been proposed. In view of the technical difficulties involved in these methods, modelling assumptions are particularly attractive for which closed-form relations between texture distortion and surface shape are available.

2.1 Closed-Form Shape from Texture Methods

Gårding proposes a unified scheme for various texture gradients in the slant-tilt surface representation [2]. The slant-tilt representation parametrises surface orientation in relation to the image screen by the tilt direction τ which is the direction with the largest inclination of the surface and slant σ as the angle of inclination in direction τ, i.e. the maximal amount of slant between surface and image plane. Texture gradients are the spatial gradients of texture characteristics such as density, size, and length on the image screen. From these texture gradients Gårding achieves closed-form solutions for tilt, slant, and additional curvature parameters. Two-dimensional Gabor filters are applied to estimate the texture gradients in practice.

The slant-tilt representation involves a coupling of slant and tilt in the reconstruction process. As a result, errors in the tilt direction estimation are carried over to the dependent slant estimation.

2.2 Separable Representations

In addition to his solutions in the slant-tilt representation Gårding proposes to separably compute surface slant in fixed directions by texture length gradients [2, Section 3.3]. A classical texture length measure is the cycle length of a periodic texture as utilized for the example in Figure 1. Thereby, Gårding proposes to determine the degrees of freedom of 3-D orientation by a pair of 2-D procedures.

Let us note that he employs a non-classic representation for the perspective camera projection, in that he utilises a simplified projection model via suitable first-order approximations and a spherical image screen instead of an image plane.

Greiner et al. investigate a method which is separable in 2-D procedures but not based on texture gradients [11]. Surface slant in one direction is determined robustly by texture length measures at two arbitrarily spaced points. Moreover, in contrast to first-order approximations of perspective projection a semi-perspective projection map is applied. However, let us stress that it is assumed that the surface at the measurement points is textured by the same homogeneous texture and, even more restricting, that both points lie on the same plane. In contrast, formulas based on texture gradients only examine a local neighbourhood and assume that the model assumptions are fulfilled in this small surrounding.

3 Shape from Texture Model

In this section we develop a pair of formulas for reconstructing surface slant from unidirectional length gradients. We apply a dedicated semi-perspective projection model that intuitively explains the formation of length gradients and directly models imaging devices with planar image screens. Our projection model allows to clearly separate important effects of the perspective projection, and this is a key feature in order to obtain finally a closed-form solution for the surface slant. We utilise the piecewise-planar surface assumption which ensures that surface orientation is uniquely determined for homogeneous textures. Of particular importance is that we construct individual formulas for surface orientation dependent on the direction of the unidirectional length gradient. First, we derive relations abstractly by examining the projection of arc length. A numerical algorithm and its utilized texture arc length measure is described in Section 3.3.

3.1 Perspective Camera Model

We model 3-D perspective projection as two independent 2-D projections in horizontal and vertical direction. In each direction we apply a semi-perspective projection P, that approximates the usual pinhole projection model locally by an affine transformation. Figure 2a explains how the perspective projection $P_{\text{perspective}}$ around $\tilde{q}$ on plane S is approximated by the concatenation of three affine transformations. Their effects are:

(i) Marked in blue, the *foreshortening effect*, i.e. the scaling of the projection of an object in relation to its orientation, is modelled by orthographic projection $P_{S \to V}$ from surface S on the auxiliary line V which is placed orthogonally to the projection direction. Thereby, the remaining projection steps are independent of the object orientation in relation to the projection direction.

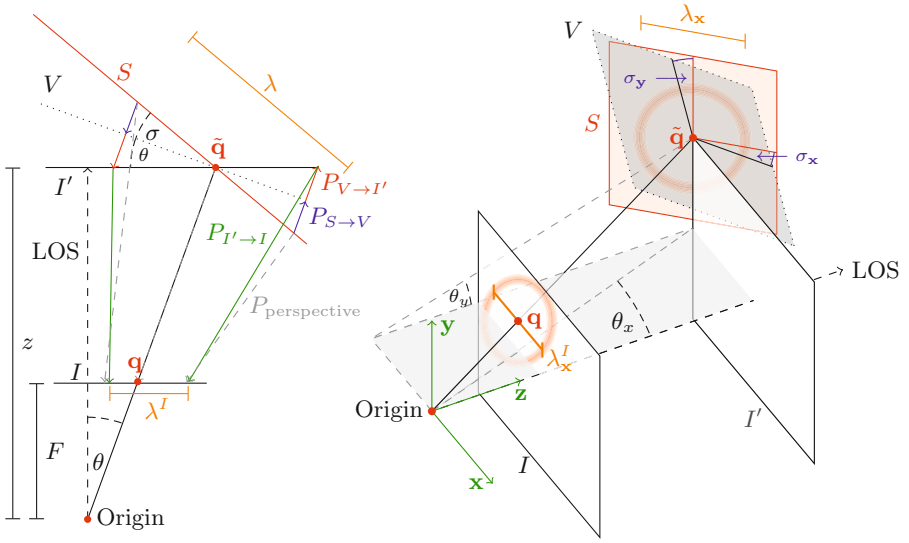


Fig. 2. a) Semi-perspective projection model in 2-D which separates the foreshortening, positioning, and perspective effect; **b)** coordinate system for 3-D surface orientation parametrised by vertical and horizontal slant $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ and projection angles $\theta_{\mathbf{x}}, \theta_{\mathbf{y}}$

(ii) Marked in red, the *positioning effect*, i.e. the influence of the angular position θ of the object in the field of view, is modelled by orthographic projection $P_{V\to I'}$ from V to the auxiliary line I' placed in parallel to the image plane I. This effect is not explicitly accounted in the semi-perspective projection applied by Greiner et al. [11].

(iii) Marked in green, the *perspective effect*, i.e. the reduction of projected size that is inversely proportional to the depth of the object to the observer, is modelled by planar perspective projection $P_{I'\to I}$ from I' onto the image plane I.

A similar semi-perspective projection model employing two affine transformation steps was introduced by Ohta et. al. [12]. Applied at the projection of a line segment of S with arc length λ, the three described projection steps are

$$P_{S\to V}(\lambda) := \cos(\sigma)\lambda, \quad P_{V\to I'}(\lambda) := \frac{\lambda}{\cos(\theta)}, \quad \text{and} \quad P_{I'\to I}(\lambda) := \frac{F}{z}\lambda, \tag{1}$$

where F is the focal length, slant σ is defined as the angle between the plane S and the auxilliary line V and z is the depth of the centre of the line segment to the optical centre. The projection direction is parametrised by angle θ. Figure 2b shows the setting of surface S and planes V, I', I in 3-D. The image plane is spanned by vector $\mathbf{x}$ pointing in horizontal direction and $\mathbf{y}$ in vertical direction. We parametrise the orientation of S by horizontal and vertical slant $\sigma_{\mathbf{x}}$,$\sigma_{\mathbf{y}}$ and the horizontal and vertical projection direction by $\theta_{\mathbf{x}}$, $\theta_{\mathbf{y}}$, where subscripts distinguish directions.

The displayed projection of a circle of width $\lambda_{\mathbf{x}}$ in Figure 2b explains the projection of arc length in 3-D. Dependent on the surface orientation it is distorted into an ellipse like shape. We name its projected width in direction v on the image screen the projected arc length in direction v and denote it with $\lambda^I_{\mathbf{v}}$. In general, $\lambda^I_{\mathbf{v}}$ jointly depends on vertical and horizontal slant $\sigma_{\mathbf{x}}, \sigma_{\mathbf{y}}$. Separability of $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ is obtained if $\mathbf{v}$ is oriented in either horizontal direction $\mathbf{x}$ or vertical direction $\mathbf{y}$. For horizontal direction $\mathbf{x}$ the concatenation of projections (1) gives

$$\lambda^I_{\mathbf{x}} = P_{I' \to I}\left(P_{V \to I'}\left(P_{S \to V}\left(\lambda_{\mathbf{x}}\right)\right)\right) = \lambda_{\mathbf{x}} \frac{\cos(\sigma_{\mathbf{x}})}{\cos(\theta_{\mathbf{x}})} \frac{F}{z}, \tag{2}$$

where $\lambda_{\mathbf{x}}$ denotes the original arc length in direction $\mathbf{x}$, i.e. the original width of the projected object in the direction of the back projection of $\mathbf{x}$ onto S. The projection of arc length $\lambda_{\mathbf{y}}$ is defined equivalently and for the projection of non-circular objects $\lambda_{\mathbf{y}}$ can differ from $\lambda_{\mathbf{x}}$.

3.2 Surface Slant Reconstruction

Here, we prove Theorem 1 which represents our key result relating texture length gradients to surface slant in our new model.

Theorem 1. *Let S be an arbitrarily oriented plane and $\lambda^I_{\mathbf{x}}$ the projection of constant arc length $\lambda_{\mathbf{x}}$ on S in horizontal direction $\mathbf{x}$, as introduced before. Then horizontal surface slant $\sigma_{\mathbf{x}}$ is determined via the the normalised projected length gradient $\partial_{\mathbf{x}}\lambda^I_{\mathbf{x}}/\lambda^I_{\mathbf{x}}$ on the image plane I up to an error ϵ with*

$$\sigma_{\mathbf{x}} = \arctan\left(-\frac{1}{2}\,\frac{\frac{\partial_{\mathbf{x}}\lambda^I_{\mathbf{x}}}{\lambda^I_{\mathbf{x}}}}{\cos^2(\theta_{\mathbf{x}})} + \tan(\theta_{\mathbf{x}})\right) + \epsilon, \tag{3}$$

and vertical slant $\sigma_{\mathbf{y}}$ by the normalised length gradient $\frac{\partial_{\mathbf{y}}\lambda^I_{\mathbf{x}}}{\lambda^I_{\mathbf{x}}}$ in $\mathbf{y}$-direction by

$$\sigma_{\mathbf{y}} = \arctan\left(-\frac{\frac{\partial_{\mathbf{y}}\lambda^I_{\mathbf{x}}}{\lambda^I_{\mathbf{x}}}}{\cos^2(\theta_{\mathbf{y}})} + \tan(\theta_{\mathbf{y}})\right) + \epsilon, \tag{4}$$

with $\sigma_{\mathbf{x}}$,$\sigma_{\mathbf{y}}$, $\theta_{\mathbf{x}}$ and $\theta_{\mathbf{y}}$ defined as before, and $\partial_{\mathbf{x}}\lambda^I$ the partial derivative of the projected arc length λ^I in direction $\mathbf{x}$.

Proof. Starting point of the slant derivation is the quotient of arc length projections at two points $\mathbf{q}, \mathbf{p}$ on the image plane. Let $\mathbf{q}$,$\mathbf{p}$ be on a horizontal line then the quotient of the projections of λ_x at $\mathbf{q}, \mathbf{p}$ is according to equation (2)

$$\frac{\lambda^I_{\mathbf{x}}(\mathbf{q})}{\lambda^I_{\mathbf{x}}(\mathbf{p})} = \frac{z^{\mathbf{P}}\,\cos\left(\sigma^{\mathbf{q}}_{\mathbf{x}}\right)\cos\left(\theta^{\mathbf{p}}_{\mathbf{x}}\right)}{z^{\mathbf{q}}\,\cos\left(\sigma^{\mathbf{p}}_{\mathbf{x}}\right)\cos\left(\theta^{\mathbf{q}}_{\mathbf{x}}\right)}, \tag{5}$$

where superscripts distinguish the parameters for locations $\mathbf{q}$ and $\mathbf{p}$. Note that the original arc lengths $\lambda_{\mathbf{x}}$ is assumed constant and drops out. Let us now consider the quotient of projected arc length at vertically spaced points $\mathbf{q}$ and $\mathbf{p}$, i.e.

$\mathbf{p} = \mathbf{q} + (0, w)$. The parameters $\sigma_{\mathbf{x}}$ and $\theta_{\mathbf{x}}$ in projection formula (2) are constant in vertical direction. Thus, they drop out for $\mathbf{q}, \mathbf{p}$ on the considered vertical line:

$$\frac{\lambda^I_{\mathbf{x}}(\mathbf{q})}{\lambda^I_{\mathbf{x}}(\mathbf{p})} = \frac{z^{\mathbf{p}} \cos(\sigma^{\mathbf{q}}_{\mathbf{x}}) \cos(\theta^{\mathbf{p}}_{\mathbf{x}})}{z^{\mathbf{q}} \cos(\sigma^{\mathbf{p}}_{\mathbf{x}}) \cos(\theta^{\mathbf{q}}_{\mathbf{x}})} = \frac{z^{\mathbf{p}}}{z^{\mathbf{q}}}. \tag{6}$$

The foreshortening and positioning effects acting in vertical position on a slanted plane are invariant to the horizontal position. This is different in equation (5) as slant $\sigma_{\mathbf{x}}$ is defined in relation to the horizontal projection direction $\theta_{\mathbf{x}}$ which varies dependent on the horizontal position on the image plane. It follows that the horizontal gradient of $\lambda^I_{\mathbf{x}}$ is simultaneously influenced by all three projective effects, while the vertical gradient is solely dependent on the perspective effect.

The left hand sides of equations (5) and (6) can be measured on the image plane, it remains to relate the depth quotient $z^{\mathbf{q}}/z^{\mathbf{p}}$ to the surface slant. To finally obtain the notion of texture gradients, we assume in the following an infinitely small sampling distance w between $\mathbf{q}, \mathbf{p}$, and we aim to consider the limit $w \to 0$. We derive in the Appendix by trigonometric relations that in this case slant $\sigma^{\mathbf{q}}$ at point $\mathbf{q}$ is determined for the quotient of $z^{\mathbf{q}}, z^{\mathbf{p}}$ by

$$\tan(\sigma^{\mathbf{q}}) = \frac{\left(\frac{z^{\mathbf{p}}}{z^{\mathbf{q}}} - 1\right)}{\cos^2(\theta^{\mathbf{q}}) w} + \tan(\theta^{\mathbf{q}}). \tag{7}$$

To obtain vertical slant $\sigma_{\mathbf{y}}$ in closed form we substitute the projected arc length quotient (6) for vertically spaced $\mathbf{p} = \mathbf{q} + (0, w)$ into (7):

$$\tan(\sigma^{\mathbf{q}}_{\mathbf{y}}) = \frac{\frac{\lambda^I_{\mathbf{x}}(\mathbf{q})}{\lambda^I_{\mathbf{x}}(\mathbf{p})} - 1}{\cos^2(\theta^{\mathbf{q}}_{\mathbf{y}}) w} + \tan(\theta^{\mathbf{q}}_{\mathbf{y}}) \tag{8}$$

Similarly, for horizontal slant we substitute (5) for horizontally spaced $\mathbf{p} = \mathbf{q} + (w, 0)$ into (7). Due to the additional dependency on the slant $\sigma_{\mathbf{x}}$ in equation (5) we have to solve for σ explicitly. This is done in the Appendix, we derive

$$\tan(\sigma^{\mathbf{q}}_{\mathbf{x}}) = \frac{1}{2} \frac{\frac{\lambda^I_{\mathbf{x}}(\mathbf{q})}{\lambda^I_{\mathbf{x}}(\mathbf{p})} - 1}{\cos^2(\theta^{\mathbf{q}}_{\mathbf{x}}) w} + \tan(\theta^{\mathbf{q}}_{\mathbf{x}}). \tag{9}$$

Finally, we conclude the proof by applying for horizontally spaced $\mathbf{p} = \mathbf{q} + (w, 0)$ the difference quotient relation

$$\lim_{w \to 0} \frac{\frac{\lambda^I_{\mathbf{x}}(\mathbf{q})}{\lambda^I_{\mathbf{x}}(\mathbf{p})} - 1}{w} = \lim_{w \to 0} \frac{\lambda^I_{\mathbf{x}}(\mathbf{q}) - \lambda^I_{\mathbf{x}}(\mathbf{p})}{w \, \lambda^I_{\mathbf{x}}(\mathbf{p})} = -\frac{\partial_{\mathbf{x}} \lambda^I_{\mathbf{x}}(\mathbf{q})}{\lambda^I_{\mathbf{x}}(\mathbf{q})} \tag{10}$$

at equation (9) and accordingly for vertically spaced $\mathbf{p} = \mathbf{q} + (0, w)$ on (8) to obtain the two postulated formulas (3), (4) for horizontal and vertical slant. Error ϵ occurs for non-zero sampling distances w, its influence is evaluated in Section 4. □

Corresponding formulas apply to vertical arc length $\lambda^I_{\mathbf{y}}$ with $\mathbf{x}$ and $\mathbf{y}$ swapped. Therefore, the slants $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ can be derived by projected arc length measures in direction $\mathbf{x}$ or $\mathbf{y}$, respectively.

3.3 Shape from Texture Algorithm

For the numerical implementation we analyse texture by its dominant local spatial wavelength in horizontal direction. It is a reasonable length measure as the wavelength of the texture is proportional to the arc length of the covered surface if the texture homogeneity assumption of constant dominant spatial wavelength and fixed wave orientation throughout the surface is fulfilled. This is the case for textures with a strong directional component such as wood grain and repetitive textures occurring in man-made environments. Figure 3 illustrates how the wavelength spectrogram of Figure 1a) is distorted according to the scene geometry. The gradient of dominant wavelength is linked to the surface orientation. We determine surface slant $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ from this length measure through equations (3), (4). The required partial derivatives are approximated by forward differences. We estimate the dominant local spatial wavelength by a sampling of the source image with unidirectional one-dimensional Gabor filters, which have an optimal space-frequency localization. Our SfT model is not restricted to the described texture analysis by the dominant local spatial frequency. For other possible means to compute texture gradients we refer to the work of Clerc and Mallat [7] where texture gradients are derived for a larger class of homogeneous textures.

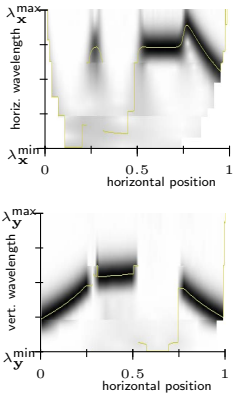


Fig. 3. Wavelength content in horizontal and vertical direction for the bisecting horizontal slice of Figure 1a with marked dominant frequency

In order to obtain reliable estimates of the surface slant one can compute wave-length estimates in horizontal and vertical direction and compute surface orientation from the estimate with the larger response. Low responses usually correspond to situations where the analysis direction is close to orthogonal to the texture pattern direction, compare Figure 3 and the geometry in Figure 1.

4 Numerical Evaluation

First we analyse the error ϵ of equation (4) for differently oriented and positioned planes. Error ϵ originates from non-zero sampling distances w and the semi-perspective projection model, which was applied to separate projective effects. We obtain ϵ in closed form by deriving the unknown projected arc length λ^I in equation (4) with the usual pinhole camera model from the parameters F, θ, z, λ, and σ. The influence of the different parameters on ϵ is illustrated in Figure 4a-c: Graph *a)* indicates that ϵ is bounded for all values of σ and for values of θ not close to $\pm\pi/2$, graph *b)* shows that ϵ is insignificant for sufficiently small quotients λ/z and a small sampling distance w, and graph *c)* displays the nearly linear dependence on the sampling distance w. The error in equation (3) leads to graphs of the same shape. This analysis covers typical parameter ranges and shows that our dedicated camera model is as accurate as the usual pinhole model.

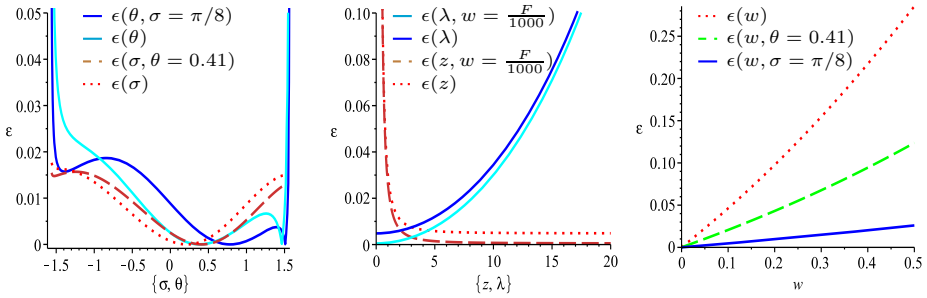


Fig. 4. Reconstruction error ϵ in relation to the parameters $\lambda = F$, $z = 10F$, $w = 0.01F$, $\theta = 0.22$, and $\sigma = \frac{\pi}{4}$, unless otherwise stated in the legend. **a)** Influence of σ, θ, **b)** of λ, z, and **c)** of the sampling distance w, lengths are measured in multiples of F.

In the first practical experiment we apply the described algorithm to a rendered view of the Cornell box scene textured by differently oriented and modulated sinusoids. Figure 1 shows the source image, its per-pixel surface normal reconstruction in standard colour code and its error. This experiment shows that independently of the texture orientation and scaling piecewise planar surfaces are reconstructed precisely for an idealised texture. Only at edges separating planar surface parts errors occur.

The applicability of our method to natural scenes is evaluated at hand of test images provided by Super and Bovic [13], see Figure 5. The image dimensions are 248x160 for the landscape images and 248x216 for the portrait images. These were also used by Greiner et al. [11] for experimental validation. Therefore, we determined an estimate of the focal length of the imaging device by cross-validation. We determined $F = 500$ pixel from the

Table 1. Slant reconstruction for the images shown in Figure 5 measured at the image centre in degree

Source image	Surface slant σ_x, σ_y	
	truth	our method
Alu Wall	0, 0	3.3, -0.9
	10, 0	14, -0.8
	20, 0	17.9, 1.3
	30, 0	28.0, -2.8
Steps	0, 70	-2.0, 61.8
Blind	0, 40	6.1, 41.6
Ventilator	0, 20	1.8, 16.6

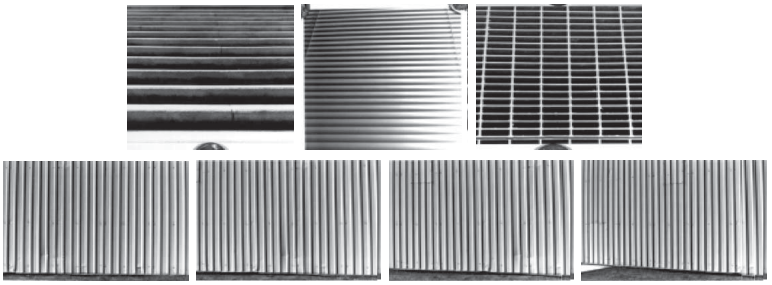


Fig. 5. Pictures of real world textured planar surfaces published by Super and Bovic [13]. **Top:** stairs, blind, and ventilator, **Bottom:** aluminum wall.

alu wall image with the given slant ground truth of 20 degree and applied the same value for the remaining images. Table 1 shows that our reconstructions are reasonable. To make our algorithm more robust to noise we calculated the wavelength derivatives from locations spaced $w = 10$ pixel apart and average the slant in the centered window of dimension 30x30 pixel.

The method of Super and Bovic infers surface orientation by a sampling with two-dimensional Gabor filters and explicit back-projection in the slant tilt system [13]. For comparing the reconstruction error we transformed their results to horizontal and vertical slant. Also, the approach of Greiner et al. [11] is restricted to a single planar surface and globally homogeneous textures. Moreover, robustness of the texture analysis is increased by fitting a parametrised curve to selected support points in the texture analysis.

To be comparable to these global approaches, we increase the sample width to $w = 30$ pixel and average over a larger window of size 80x80 pixel. The results are compared in Table 2. Our reconstructions are superior to the results of Super and Bovic and, considering the specified ground truth precision of 1-3 degree [13], comparable to the results of Greiner et al.

Table 2. Error analysis of the surface slant reconstruction for the source images shown in Figure 5, compared to the findings in [13] and [11]

source image	Absolute slant error $\epsilon_{\mathbf{x}}, \epsilon_{\mathbf{y}}$ our method	Super, Bovic	Greiner et al.
Alu Wall	0.8, 0.0	1.0, 0	0.0, n/a
	0.1, 1.4	0.3, 0.5	0.4, n/a
	0.0, 0.0	3.3, 0.3	0.6, n/a
	0.1, 0.2	3.6, 0.2	0.6, n/a
Steps	3.3, 0.3	3.5, 2.0	n/a, 1.5
Blind	6.0, 1.7	0.6, 7.9	n/a, 0.2
Ventilator	2.4, 2.4	1.8, 4.9	n/a, 0.2
Mean	**1.34**	**2.14**	**0.50***

*Mean of the available estimates.

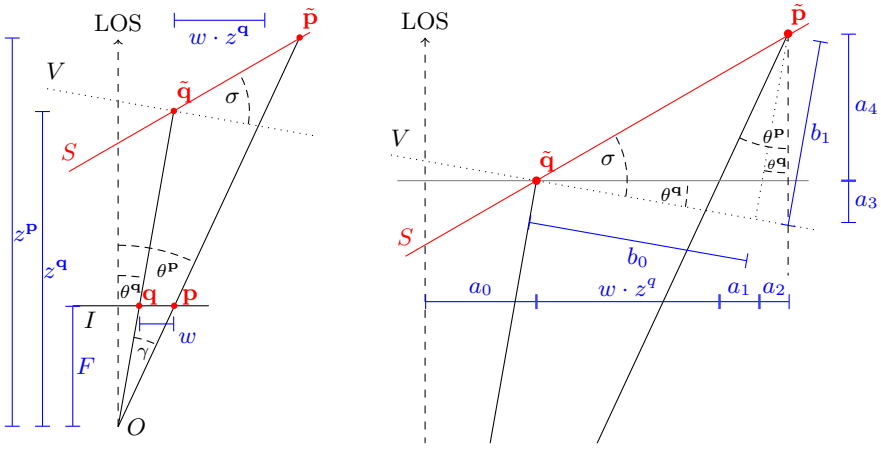


Fig. 6. a) Geometric relations for two points $\mathbf{q}, \mathbf{p}$ on the image plane and their origins $\tilde{\mathbf{q}}, \tilde{\mathbf{p}}$ on line S, **b)** Close-up view with additional right angled auxiliary triangles

5 Conclusion

In this paper we elaborated on a novel approach to SfT in that only an unidirectional texture analysis is employed to reconstruct surface orientation. The main tool in this is a well-engineered composition of model components. Our work shows that a careful mathematical investigation can lead to improvements even for difficult problems in computer vision.

Acknowledgments. The authors would like to thank Super and Bovic for releasing their test images.

A Appendix

Derivation of Relation (7) between σ and Depths $z^{\mathbf{q}}, z^{\mathbf{p}}$. The relation between the depths $z^{\mathbf{q}}, z^{\mathbf{p}}$ of the back projections of points $\mathbf{q}$ and $\mathbf{p}$ onto line S and the slant $\sigma^{\mathbf{q}}$ of S is explained geometrically in Figure 6a. Figure 6b shows its close-up view with additional auxiliary lines. For convenience we assume that lengths are measured in multiples of F. For the rectangular triangle with sides $\overline{\tilde{\mathbf{q}}\tilde{\mathbf{p}}}, b_0$ and b_1 the definition of tangens is for slant angle $\sigma^{\mathbf{q}}$

$$\tan(\sigma^{\mathbf{q}}) = \frac{b_1}{b_0}. \tag{11}$$

Similarly, b_0, b_1 are related to the projection angles $\theta^{\mathbf{q}}, \theta^{\mathbf{p}}$, and depths $z^{\mathbf{q}}, z^{\mathbf{p}}$ by the definition of cosinus and tangens. From known angle and side length pairs we determine lengths b_0, b_1 through:

$$\begin{aligned} &b_1 = \cos(\theta^{\mathbf{q}})(a_4 + a_3),\ b_0 = \cos(\theta^{\mathbf{q}})(wz^q + a_1),\ a_1 = \tan(\theta^{\mathbf{p}})a_4 - a_2, \\ &a_2 = \tan(\theta^{\mathbf{q}})a_4,\quad a_3 = \tan(\theta^{\mathbf{q}})(wz^q + a_1 + a_2),\ a_4 = z^{\mathbf{p}} - z^{\mathbf{q}}, \end{aligned} \tag{12}$$

with spatial sampling distance w and auxiliary variables a_i. In combination the system of equations from (11) and (12) gives the desired relation between $\sigma^{\mathbf{q}}$ and the quotient of depths by

$$\tan(\sigma^{\mathbf{q}}) = \frac{(1 + \tan(\theta^{\mathbf{q}})\tan(\theta^{\mathbf{p}}))\left(\frac{z^{\mathbf{p}}}{z^{\mathbf{q}}} - 1\right) + w\tan(\theta^{\mathbf{q}})}{w + (\tan(\theta^{\mathbf{p}}) - \tan(\theta^{\mathbf{q}}))\left(\frac{z^{\mathbf{p}}}{z^{\mathbf{q}}} - 1\right)}. \tag{13}$$

Next, we simplify this formula in the limit $w \to 0$. The angle $\theta^{\mathbf{p}}$ depends on $\theta^{\mathbf{q}}$ with $\theta^{\mathbf{p}} = \theta^{\mathbf{q}} + \gamma$. We reach (7) by approximating $\tan\theta^{\mathbf{p}}$ at $\gamma = 0$ with its Taylor linearisation and applying the approximation

$$\gamma = \theta^{\mathbf{p}} - \theta^{\mathbf{q}} = \tan^{-1}(\tan(\theta^q) + w) - \theta^{\mathbf{q}} = \frac{w}{1 + \tan(\theta^{\mathbf{q}})} + O(w^2), \tag{14}$$

consider Figure 6a for the geometric relation. Remaining terms in $O(w)$ vanish in the considered limit $w \to 0$ and are dropped out.

Derivation of Equation (9). We first substitute $\sigma_{\mathbf{x}}^{\mathbf{q}} = \sigma_{\mathbf{x}} + \gamma$ and $\theta_{\mathbf{x}}^{\mathbf{q}} = \theta_{\mathbf{x}} + \gamma$ in (5), and linearise in γ. Solving for $\frac{z^{\mathbf{p}}}{z^{\mathbf{q}}}$ leads to

$$\frac{z^{\mathbf{p}}}{z^{\mathbf{q}}} = \frac{\lambda_{\mathbf{x}}^I(\mathbf{q})}{\lambda_{\mathbf{x}}^I(\mathbf{p})} \cdot \frac{1 - \gamma \tan(\sigma^{\mathbf{q}})}{1 - \gamma \tan(\theta^{\mathbf{q}})}. \tag{15}$$

We substitute the thereby linearised version of (5) into (7) and relate γ to w by (14). Finally, solving for $\tan(\sigma)$ with the help of a computer algebra system and dropping terms in $O(w)$ leads to (9). Note that some factors in $O(w)$ cancel out during the derivation, hence, terms in $O(w)$ cannot be dropped earlier.

References

1. Horn, B.: Robot vision. MIT press (1986)
2. Garding, J.: Shape from texture for smooth curved surfaces in perspective projection. Journal of Mathematical Imaging and Vision 2, 630–638 (1992)
3. Forsyth, D.: Shape from texture without boundaries. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part III. LNCS, vol. 2352, pp. 225–239. Springer, Heidelberg (2002)
4. Witkin, A.P.: Recovering surface shape and orientation from texture. Artificial Intelligence 17, 17–45 (1981)
5. Forsyth, D.: Shape from texture and integrability. In: Proc. ICCV, pp. 447–452 (2001)
6. Galasso, F., Lasenby, J.: Shape from texture via fourier analysis. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., et al. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 803–814. Springer, Heidelberg (2008)
7. Clerc, M., Mallat, S.: The texture gradient equation for recovering shape from texture. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 536–549 (2002)
8. Watson, A., Ahumada, A.: A standard model for foveal detection of spatial contrast. Journal of Vision 5(9), 717–740 (2005)
9. Lee, K., Kuo, C.: Direct shape from texture using a parametric surface model and an adaptive filtering technique. In: Proc. CVPR, pp. 402–407 (1998)
10. Loh, A., Hartley, R.: Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In: Proc. BMVC, pp. 69–78 (2005)
11. Greiner, T., Rao, S.G., Das, S.: Estimation of orientation of a textured planar surface using projective equations and separable analysis with m-channel wavelet decomposition. Pattern Recognition 43(1), 230–243 (2010)
12. Ohta, Y., Maenobu, K., Sakai, T.: Obtaining surface orientation from texels under perspective projection. In: Proc. IJCAI, pp. 746–751 (1981)
13. Super, B.J., Bovik, A.C.: Planar surface orientation from texture spatial frequencies. Pattern Recognition 28, 729–743 (1995)

Multi Scale Shape Index for 3D Object Recognition

Ujwal Bonde, Vijay Badrinarayanan, and Roberto Cipolla

Deparment of Engineering, University of Cambridge, Cambridge, UK

Abstract. We present Multi Scale Shape Index (MSSI), a novel feature for 3D object recognition. Inspired by the scale space filtering theory and Shape Index measure proposed by Koenderink & Van Doorn [6], this feature associates different forms of shape, such as umbilics, saddle regions, parabolic regions to a real valued index. This association is useful for representing an object based on its constituent shape forms. We derive closed form scale space equations which computes a characteristic scale at each 3D point in a point cloud without an explicit mesh structure. This characteristic scale is then used to estimate the Shape Index. We quantitatively evaluate the robustness and repeatability of the MSSI feature for varying object scales and changing point cloud density. We also quantify the performance of MSSI for object category recognition on a publicly available dataset.

1 Introduction

The availability of cheap IR sensors have considerably lowered the cost of real time 3D data acquisition [11] and has led to a renewed interest in 3D object recognition [13,5]. This has encouraged research into the development of a number of shape inspired features for 3D, several of which are extensions to popular 2D features [23,12,5] and do not directly operate on point cloud data, while others [18,17] are not robust enough to sensor noise. In this work, we propose a novel feature called the Multi Scale Shape Index (MSSI) which is jointly motivated by scale space filtering theory [21,10] and the shape categorization work of Koenderink [6]. Shape Index (SI) maps points on surfaces to a linear scale $[-1 : 1]$ and thus classifies them into categories such as Umbilics, Parabolics and Saddle points. Fig. 1 shows a few canonical shapes and their corresponding shape index. The proposed MSSI feature operates directly on a point clouds and are robust to noise in the data.

The SI measure at a 3D point is a function of the principal curvatures at that point. This measure was originally proposed for the continuous domain. However, computing the principal curvatures at a point from noisy 2.5D or 3D data can be erroneous if the characteristic scale at that point is not known (see Fig. 2 for an example). In this work, we show how to compute the characteristic scale at a point in a discrete domain (point cloud) and then estimate the shape index at this scale. We then construct the MSSI feature at a point as a concatenation of its characteristic scale, shape index and a measure of curvedness [6]. An example

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 306–318, 2013.

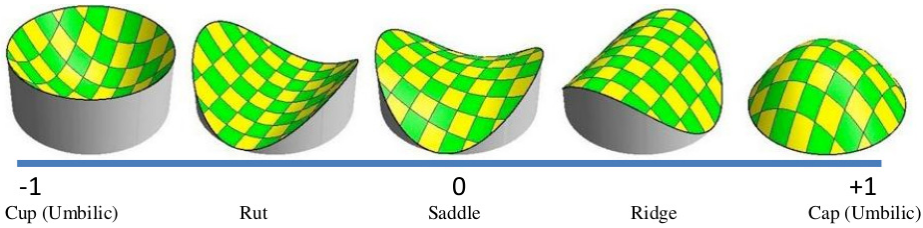


Fig. 1. Illustration of shape index measure mapping shapes to real number. From [6].

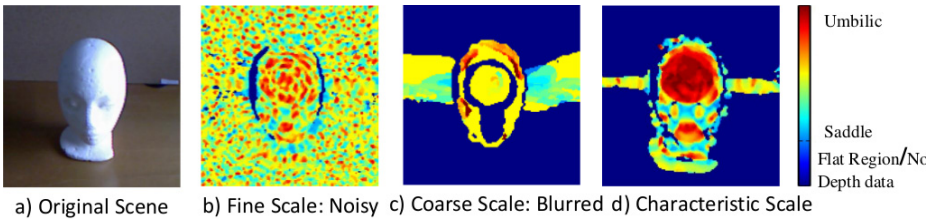


Fig. 2. Effects of computing Shape Index at an erroneous scale for a real world scene. Shape Index computed at the characteristic scale (d) is more stable as compared to one computed at a fine scale (b), which is sensitive to noise, or a coarse scale (c) which blurs high curvature regions. Best viewed in colour.

of each of these features for a real world scene is shown in Fig. 3. The RGB image and the corresponding depth map from a Kinect sensor is shown in Fig. 3 (a,b). The dummy's head has a large scale and is classified as an umbilic (doubly convex shape) in Fig. 3 (d). It also has low curvedness as seen in Fig. 3 (e). The tip of the nose has low scale, and is an umbilic with high curvedness. The map of the triplet of these three features is the MSSI map shown in Fig. 3 (c). To show the efficacy of our proposed feature for category recognition we compare it with the work of Lai *et al.* [7] using their publically available dataset.

The remainder of our paper is organised as follows. In Sec. 2 we discuss relevant literature. The MSSI feature computation is described in Sec. 3. Our experimental setting and results are elaborated in Sec. 4. We conclude in Sec. 5.

2 Literature Review

Their exist many 3D features in literature that try to capture local shape. Recently, Zaharescu *et al.* [23] provided an extension of HOG features for meshes, by bining the directional derivatives of the mean curvature. However, computing the mean curvature in the discrete domain is not straight-forward as we show in section 3. The Heat kernel signature proposed in [18] is based on the fundamental solution to the heat diffusion equation. However this method is sensitive to noise and changes in the configuration of the original mesh. Furthermore,

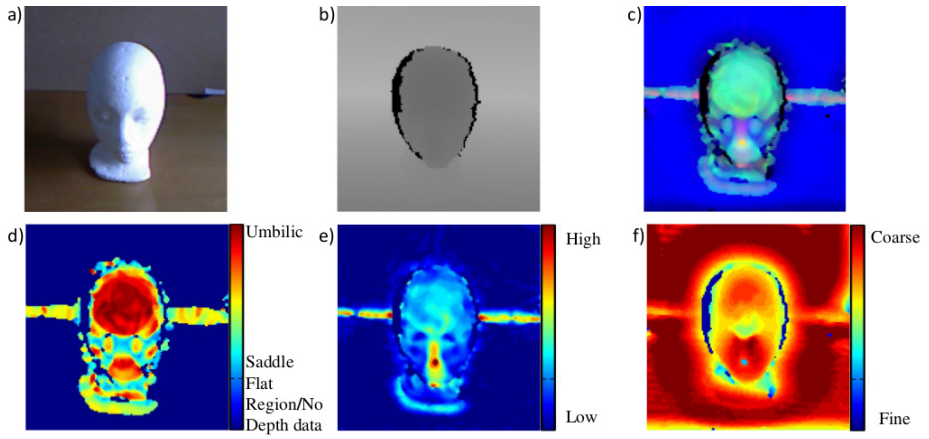


Fig. 3. RGB image of an example scene (a), its depth map obtained from kinect sensor(b). The shape index map (d) categorizes forehead and nose as umbilics, while the nose bridge is estimated as a saddle. Background is correctly assigned as a flat region. The characteristic scale map(e) assigns the nose tip to a fine scale, while the forehead has a relatively coarser scale. The Curvedness map (f) assigns the nose tip to a very high curvedness value while the background has very low curvedness. The three components together gives the MSSI feature map (c). Best viewed in colour.

although approaches for triangulating and generating surfaces/meshes from a point cloud do exist, they are slow, noise sensitive or require dense point clouds as pointed out in the survey in [3].

Features that operate directly on 3D point clouds do exist [19,5] and are extensions of popular 2D features(SIFT,SURF). These methods however do not assign stable canonical frames which are needed for them to be rotationally invariant. To address this issue, the authors of [12] provide a method to compute a stable canonical frame. Unlike their previous approach [19] which worked directly on point clouds, this method requires a mesh structure.

Many of the advances made in 2D Object Recognition in the past decade have been adopted for 3D Object Recognition and have shown promising results. Knopp *et al.* [5] extends the Implicit Shape Model (ISM) model proposed by [9] to 3D Object Recognition. They use a Hessian-based interest point detector that encodes an extension of 2D SURF features to 3D [2]. These interest points are then clustered to form the ISM. Their method showed promising results on clean meshed data. However, they do not report any results on real world 3D/2.5D point cloud data. Lai *et al.* combine colour and depth information using pyramid Histogram Of Gradients(HOG) features [7,4]. These features are used with a linear Support Vector Machine(SVM) to perform sliding window based object detection. They show competitive results using depth and colour features individually, and improve it further by combining both features on one of the largest publicly available 2.5D dataset. Both these methods concentrate on a fusion of depth and intensity/colour features for recognition in real world

scenarios. However, there is no explicit attempt to capture shape for recognition. In this work, we propose a local shape based feature (MSSI) to exploit depth data and demonstrate that competitive results can be achieved with lesser training data.

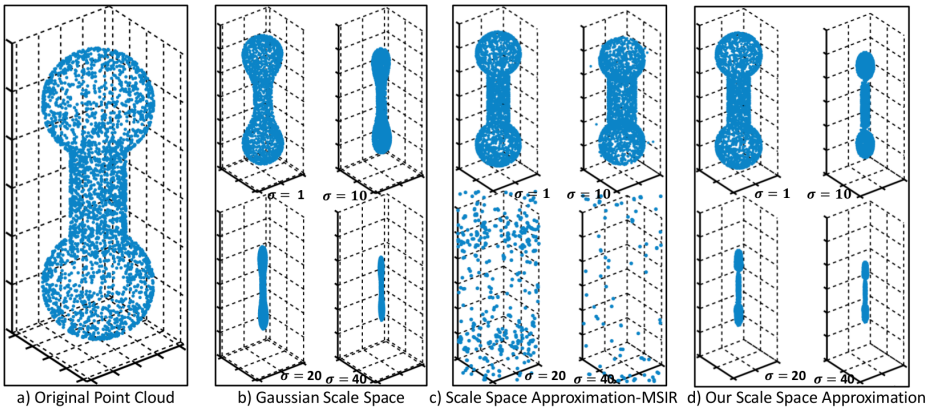


Fig. 4. Scale space approximation for a synthesized dumbell shape. Our method approximates the original scale space while predictions from MSIR [20] are erroneous.

3 Proposed Multi-Scale Shape Index (MSSI) Feature

Shape index (SI) as proposed by Koenderink [6] is a function of the principal curvatures(κ_1, κ_2)[1]. Principal curvatures are primarily defined for a continuous parameterization of the 3D surface. While their exist methods to approximate them for discrete spaces (point clouds) [15] they require a *support region* to compute them. However, the size of the support region itself is dependent on the principal curvatures. This can be seen from Fig. 4; when the shape index is computed at a fine scale, it is sensitive to noise thus falsely classifying noisy low curvature (flat) regions as umbilics. On the other hand, at a coarser scale, regions of high curvature get blurred out (nose) while low curvature regions are classified correctly. To address this ambiguity in the size of the support region, we propose to obtain a *characteristic scale* automatically by relating the effect of blurring at different scales to the underlying local shape of the point clouds. Our approach is motivated by Multi-Scale Interest Region (MSIR) [20] approach to locate *interest regions*. However, the scale space model in their work is neither accurate for basic shapes nor is stable as shown in Fig. 4 and Fig. 5. In the remainder of this section we derive the relationship between characteristic scale and principal curvature and compare it with MSIR.

Curves in 3D Space. We start by considering a continuous arc-length parameterized curve $\alpha(s)$ in $\mathbb{R}^3$, where $s \in (-B,\ B)$. Here, B denotes extent of the

[1] $SI = \frac{2}{\pi} arctan\left(\frac{\kappa_2+\kappa_1}{\kappa_2-\kappa_1}\right) \quad \kappa_1 \geq \kappa_2,$

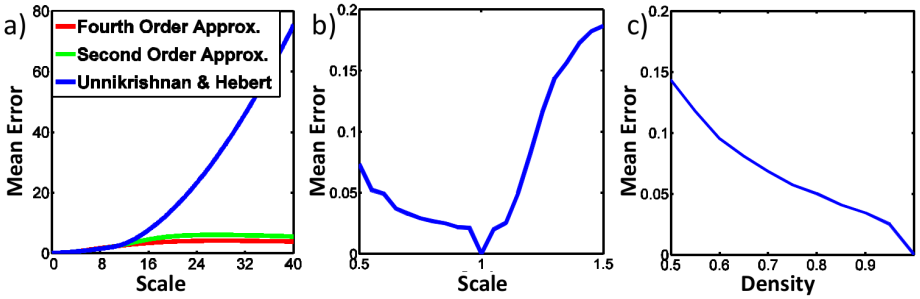


Fig. 5. a) Mean prediction error of the approximated scale space for the synthesized dumbell shape in Fig 4. MSIR [20] is accurate only at lower scales but our fourth order model approximates quite accurately the Gaussian scale space at all scales. b) Mean estimation error of the computed shape index as we vary the scale for the head model, Fig. 3. Due to the relatively coarser scale of the head a larger error is seen as the scale is increased and a smaller error for lower scales. c) Mean estimation error of the computed shape index as we vary point cloud density is smooth for smaller changes in density. As the density decreases to low values very few points remain to correctly estimate both the characteristic scale as well as the shape index.

curve. We define $A : \mathbb{R}^3 \times \mathbb{R}^+ \rightarrow \mathbb{R}^3$ as the family of curves obtained by filtering the original curve at different scales. i.e,

$$A(\alpha(s), \sigma) = \int_{-B}^{B} \phi(s-u, \sigma)\alpha(u)\, du, \tag{1}$$

where, ϕ is the Gaussian kernel. We consider the evolution of a point x as it is filtered. Without loss of generality, we take this point to be $s = 0$ and define $x = \alpha(0)$. Performing a Taylor series expansion of α around x up to fourth order terms, we can approximate the integral as shown below:

$$A(x, \sigma) \approx \frac{1}{\sqrt{2\pi}\sigma} \int_{-B}^{B} e^{-\frac{u^2}{2\sigma^2}} \left(x + u\alpha'(0) + \frac{u^2}{2!}\alpha''(0) + \frac{u^3}{3!}\alpha'''(0) + \frac{u^4}{4!}\alpha''''(0) \right) du \tag{2}$$

For better readability we set $x' = \alpha'(0)$, $x'' = \alpha''(0)$ and so on. Observing that the second and fourth term in the equation go to zero and performing the integration over the remaining terms we get:

$$A(x, \sigma) = \Phi\left(\frac{B}{\sqrt{2}\sigma}\right)\left(x + x''\frac{\sigma^2}{2} + x''''\frac{\sigma^4}{8}\right) - \sqrt{\frac{2}{\pi}} B\sigma e^{-\frac{B^2}{2\sigma^2}} \left(\frac{x''}{2} + B^2\frac{x''''}{24} + 3\sigma^2\frac{x''''}{24} \right) \tag{3}$$

Using results from differential geometry (see the supplementary material) the above equation can be approximated as:

$$A(x,\sigma) \approx \Phi\left(\frac{B}{\sqrt{2}\sigma}\right)\left(x + \kappa\mathbf{N}\frac{\sigma^2}{2} + \left(3\kappa'\kappa\mathbf{T} - \kappa^3\mathbf{N}\right)\frac{\sigma^4}{8}\right) - \sqrt{\frac{2}{\pi}}B\sigma e^{-\frac{B^2}{2\sigma^2}}\left(\frac{\kappa\mathbf{N}}{2} + B^2\frac{\left(3\kappa'\kappa\mathbf{T} - \kappa^3\mathbf{N}\right)}{24} + 3\sigma^2\frac{\left(3\kappa'\kappa\mathbf{T} - \kappa^3\mathbf{N}\right)}{24}\right), \tag{4}$$

Note that functions κ, $\mathbf{T}$, $\mathbf{N}$ are evaluated at x which is suppressed for better readability. Here, κ, $\mathbf{T}$, $\mathbf{N}$ are the curvature, tangent and normal to the curve α and Φ is the error function. If $\sigma \ll B$ the error function can be approximated to 1 and the second term $\to 0$. This resulting equation is similar to the MSIR model. However, for a bounded curve, as the scale of blurring increases($\sigma \to B$) the contribution of the second term and the error function is significant and cannot be ignored (see Fig. 5). We next extend these equations to surfaces in 3D space.

Extension to 3D Surfaces. Let x be a point on a surface M. Further, let the normal at x be denoted as N and its tangent plane as T_x. There then exists a family of planes Π_θ that contain the normal N. The normals of these planes lie in the tangent plane T_x. Let the angle subtended by these planes with the first principal direction be θ [6]. These planes then intersect the surface to give a family of curves $\alpha_\theta(s)$ which are called the *normal sections*. Now, the net *displacement* of the point $x = \alpha_\theta(0)$, after blurring, will be equal to the average displacement caused by each normal section. Using Eq.(1) we have:

$$A(x,\sigma) = \frac{1}{2\pi}\int_0^{2\pi} A(\alpha_\theta(0),\sigma)\,d\theta = \frac{1}{2\pi}\int_0^{2\pi}\left(\int_{-B_\theta}^{B_\theta}\phi(0-u,\sigma)\alpha_\theta(u)\,du\right)d\theta. \tag{5}$$

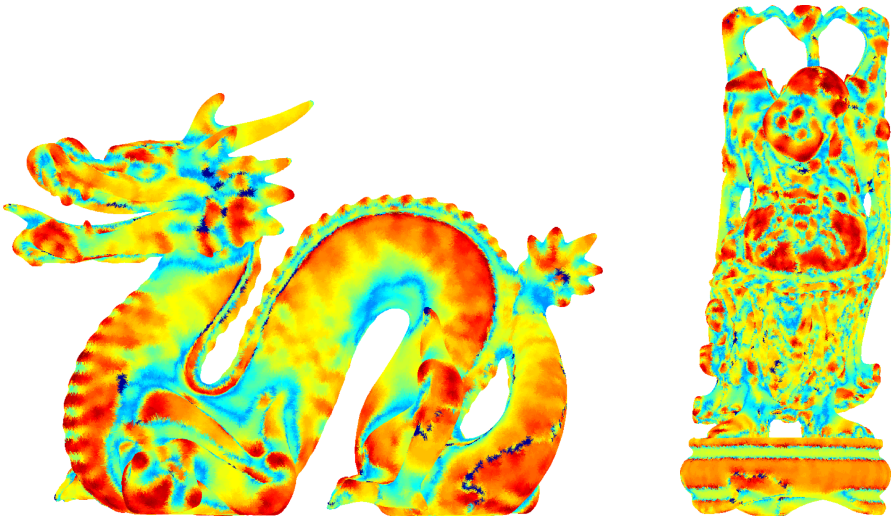


Fig. 6. Computed shape index map on stanford dragon and happy buddha models. Spikes on the back of dragon and the pointed tail are estimated as umbilics. Regions where the dragons body twists are estimated as saddle. The intricate structure on happy buddha produces more variations in the shape index. Best viewed in colour.

Solving this equation is not trivial as both (B_θ) and (α_θ) are functions of θ. Moreover, we do not have any explicit form for B_θ which represents the extent of the 3D surface in all directions. Using empirical evidence, we propose setting B_θ to be a constant value B proportional to the average geodesic distance in all directions. In practice, for a given point in a discrete point cloud, this is equal to the average geodesic distance of that point to all other points. From Eqs.(4),(5) we get (see supplementary material for further details):

$$\tilde{A}(x,\sigma) \triangleq \frac{A(x,\sigma)}{\Phi\left(\frac{B}{\sqrt{2}\sigma}\right)} \approx \left(x + H\mathbf{N}\frac{\sigma^2}{2} - \frac{H}{16}(5H^2 - 3G)\sigma^4\mathbf{N}\right), \tag{6}$$

Again, note that functions H, G and $\mathbf{N}$ are evaluated at x which is suppressed for better readability. Here, H, G and $\mathbf{N}$ are the mean curvature, Gaussian curvature and normal to the surface M. $\tilde{A}(x,\sigma)$ is the normalized family of surface obtained from Gaussian blurring with scale σ. This can be further rearranged to give:

$$D(x,\sigma) \triangleq ||\tilde{A}(x,\sigma) - x||_2 = \left(H\frac{\sigma^2}{2} - \frac{H}{16}(5H^2 - 3G)\sigma^4\right) = \frac{H\sigma^2}{2}\left(1 - \frac{\sigma^2 H^2}{4}\left(1 + \frac{1.5}{S^2}\right)\right),$$

where, S relates to the shape index via $S = tan(2\pi \times SI)$, and $D(x,\sigma)$ can be viewed as the approximate distance traveled by a point when blurred with a kernel of scale σ. Inspired by the principle of automatic scale selection, as defined by Lindeberg [10], we define the characteristic scale (σ_{max}) as the maxima in the normalized distance $(\frac{D(x,\sigma)}{\sigma})$ traveled by a point. This is given by $\partial\left(\frac{D(x,\sigma)}{\sigma}\right)/\partial\sigma = 0$, which gives:

$$\sigma_{max}^2 = \frac{4}{3H^2(1 + \frac{1.5}{S^2})}. \tag{7}$$

This derivation can also be carried out in the discrete domain by assuming a uniform point cloud sampling and approximating the integral by a sum.

Eq.(7) relates the shape index to the characteristic scale and thus motivates the term *Multi-Scale Shape Index*. The characteristic scale that we obtain is different from that proposed in MSIR due to two reasons: a) we explicitly consider the curve to be bounded and b) we model the effect of blurring until the fourth order of the Taylor series. Fig. 4 shows an example of a dumbell shaped point cloud on which we demonstrate the effect of these changes. The actual scale space obtained by Gaussian blurring is shown, along with the prediction obtained using our model and that of MSIR. Fig. 5 shows a quantitative comparison of the mean error for the two models. MSIR is accurate only at lower scales but our fourth order model approximates the Gaussian scale space accurately.

Algorithm 1 gives a stepwise procedure to compute the characteristic scale from a point cloud. As input our algorithm requires the range of scales (σ_k) to search over and an initial smoothing parameter (σ_s) before computing the scale space. We set B to be proportional to the average geodesic distance and call this as the bounding factor (B_{fct}) which is also an input for our algorithm.

The estimated characteristic scale is used as a *support region* to compute the shape index. We only calculate the magnitude of the shape index and not its sign. Fig. 6 shows the shape index map on some publically available 3D models.

We compare the robustness and repeatability of the computed shape index against variations in scale and point cloud density. We use a 3D point cloud of the head model used in Fig. 3 for these experiments. As we cannot establish the ground truth, we treat the shape index computed at the original scale and cloud density as the reference.

input : Point Cloud: $X = \{x_i\}$, Range of Scales: σ_k, Bounding factor: B_{fct}, intial smoothing: σ_s
output: Characteristic Scale for each point $\sigma_{max}(x_i)$

1. Compute a Disjoint Minimum Spanning Tree on X to form a graph $\mathbb{G}$.
2. Using Dijkstra's algorithm approximate the graph distance between points, $d_{\mathbb{G}}$, as the geodesic distance.
3. Calculate the average geodesic distance for each point d_{avg}.

foreach x_i **do**

4. $\hat{x_i} \leftarrow \frac{1}{n_{x_i}} \sum_j exp\left(-\frac{d^2_{\mathbb{G}}(x_i,x_j)}{2\sigma_s^2}\right) x_j$, where n_{x_i} is the normalizing factor.

`// Intial Smoothing`

foreach σ_k **do**

5. $A(x_i, \sigma_k) \leftarrow \frac{1}{n_{x_i}} \sum_j exp\left(-\frac{d^2_{\mathbb{G}}(x_i,x_j)}{2\sigma_k^2}\right) x_j$
6. $\tilde{A}(x_i, \sigma_k) \leftarrow A(x, \sigma) \times \left(\Phi\left(\frac{B_{fct} * d_{avg}(x_i)}{\sqrt{2}\sigma_k}\right)\right)^{-1}$
7. $D(x_i, \sigma_k) \leftarrow ||\tilde{A}(x, \sigma) - x||_2$

end

8. $\sigma_{max}(x_i) \leftarrow max_{\sigma_k} D(x_i, \sigma_k)$

end

Algorithm 1: Computation of the characteristic scale for point clouds

Change in Scale. We vary the scale from half the original scale to 1.5 times the original scale. The left panel in Fig. 5 plots the resulting deviation from the ground truth. All parameters (σ_k, σ_s, B_{fct}) are kept constant. Since the head is of a relatively coarser scale and the initial smoothing is kept constant, a higher rate of deviation from the ground truth is seen as we increase the size of the head model. On the other hand, decreasing the scale of the model while keeping the initial smoothing constant does not affect the coarser scale regions and thus a lower deviation from the ground truth is observed in this case.

Change in Density. We vary the density from the original density to half its density. The right panel in Fig. 5 plots the resulting deviation from the ground truth averaged over 10 different trials. Once again σ_k, σ_s, B_{fct} are kept constant. A smooth deviation form the ground truth is observed as the density is reduced to 3/4 its original density. As the density decreases to low values very few points remain to correctly estimate both the characteristic scale as well as the shape index and thus a higher deviation from ground truth is observed at low densities.

3.1 Object Recognition with the MSSI Feature

The shape index alone does not capture all the information about the underlying shape [6]. Being a ratio of the principal curvatures, it does not provide any information about the magnitude of the curvatures. For example, a tennis ball and a football are both spherical, but have completely different size with the tennis ball having a higher magnitude of the principal curvatures compared to a football. This notion is captured by the curvedness measure proposed by Koendrink [2]. The characteristic scale is used as another feature to capture the scale and thus we form a triplet of features, which we call the Multi-Scale Shape Index (MSSI) feature.

Detecting interest points, followed by a bag-of-visual-words approach is a common strategy in 2D object recognition [1,8]. However in 3D, as reported in the survey by [22], corner detectors are relatively less robust to noise compared to region based methods. We therefore follow a region based approach to object shape encoding. We start by super-pixellizing the MSSI feature map. We use the fast and efficient SLIC super-pixels [14]. Fig. 7 shows an example of super-pixels for different viewpoints of the head model. As seen from the images, these super-pixels are fairly stable across viewpoints. This empirical observation motivates us to use super-pixels for category recognition. To further capture the variations of shape within a superpixel, we also include the angle between each pixel normal and its corresponding *superpixel normal*[3]. Although this is correlated to the variation in shape index, empirically we observed that it improves recognition rate by introducing redundancy.

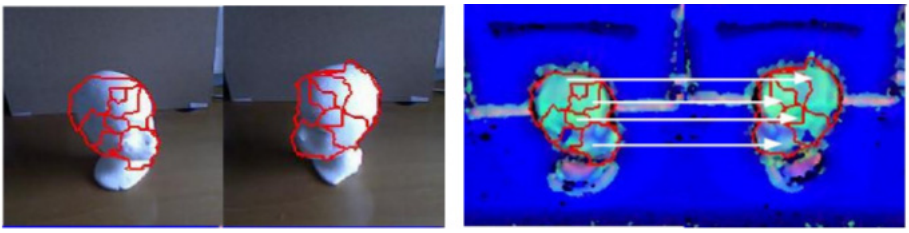


Fig. 7. An illustration of the stability of MSSI feature based superpixels across viewpoints. Super-pixels with similar MSSI features appear at approximately the same relative location in both view points. See supplementary material for an example on cluttered scenes. Best viewed in colour.

We cluster the concatenation of MSSI features and normals at each pixel into a preset number of clusters[4]. The super-pixel descriptors are obtained by binning the MSSI+normal features for each of their pixels. As these super-pixels using MSSI features, the resulting super-pixel descriptors are very sparse. Therefore

[2] $curvedness = \sqrt{\frac{\kappa_1^2+\kappa_2^2}{2}}$.
[3] Mean of normals of all pixels within it.
[4] We empirically fixed this to 300.

to enrich the descriptor of a super-pixel, we compute a weighted average of descriptors of super-pixels that are at most two hops away from it (1 and 2 neighbourhood in a graph sense). The weights used are proportional to the depth difference between the super-pixels.

We train our super-pixel based recognition approach using an RBF kernel SVM. We use a 1-vs-all setup. The super-pixel in the test set are classified individually during testing. The resulting classification gives us an initial region of interest for possible object locations. Thresholding on the number of connected pixels within these region of interests gives the final object detection.

4 Experiments, Results and Discussion

Many 3D object recognition datasets have been introduced in the recent years [16,13,7]. Of these, one of the largest is the RGB-D dataset [7]. We compare our recognition algorithm using MSSI features with the pyramid hog based depth features of Lai *et. al* [7]. We used the original authors code to obtain their results.

Dataset: The RGB-D dataset [7] contains challenges for both instance level as well as category level object recognition and detection. We perform our experiments for the category level object detection. For the category case, five items are ground truth-ed by the authors. Of these we choose four categories: cap, coffee mug, flashlight and soda can. We do not choose the bowl category since there is a large variation in the size of the bowl category. This results in a large variation in the characteristic scale (and thus in MSSI feature) which is difficult to capture with limited training instances per category. Each category has 4 or more instances and we train on only 2 instances. The training set contains about 600-1000 depth and RGB (which we do not use) images of 640x480 resolution each captured on a turntable at 3 different angles. As mentioned earlier since shape is fairly constant with small changes in viewing angles, we use only 1/3rd of the training data. The test set contains 8 video sequences with 98-230 frames per sequence. The number of objects in each sequence varies as does the clutter. We currently use 320x240 resolution images to process this large dataset and hence the results quoted in [7] are different from those computed here. At this resolution, we consider minimum object size to be at least 1000 pixels. Varying the threshold on the number of connected pixels we plot the resulting Precision-Recall (P-R) curve in Fig. 8. The qualitative recognition results of our system are shown in Fig.9.

We downsampled the dataset depth images by half to 320x240 resolution to speed up computation. This particularly affects performance on small object categories (soda can). We expect to perform better when our system is scaled up to a larger resolution of images. From Fig. 8 we see that our MSSI features based recognition easily outperforms the method of [7] in two classes (coffee mug and flashlight). MSSI features cope better with reduced training data and lower image resolution. For the soda can category, although our performance is relatively better than [7], both methods suffer due to the small size of the object. Only on the cap class, we are marginally worse off in performance.

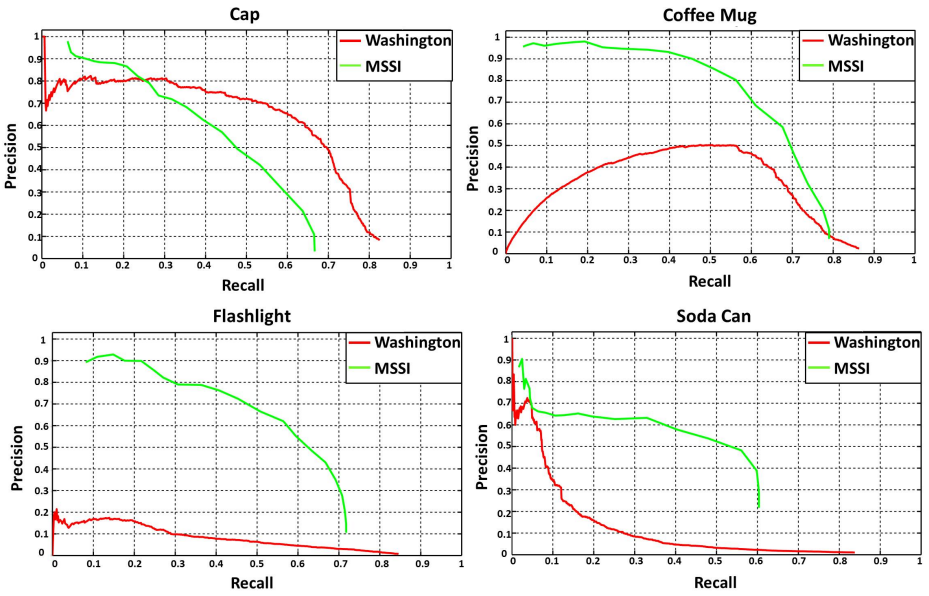


Fig. 8. Precision/Recall curves for our MSSI feature based category recognition and its comparison with depth features based recognition of Lai *et. al* [7](shown as washigton in plots). MSSI features based recognition clearly outperforms the method of Lai *et. al* in the coffee mug and flashlight classes. For the cap class, we are only marginally worse off in performance, and for the soda can category, although our performance is relatively better than Lai *et. al*, both methods suffer due to small size of the object.

We now discuss the effect of some of the influential parameters of our system:

Effect of Bounding Factor B_{fct}. Setting B_{fct} to a large value we obtain only coarse scale changes and mask the effect of smaller scale objects in the scenes. On the other hand by setting it to a small value we obtain small neighbourhood scale changes which mostly originate from the sensor noise present in the data. In general we found the average geodesic distance (or a fraction of it) to be a good approximation.

Weighting of Super-Pixel Neighbourhoods. To form the final descriptor for a super-pixel, we compute a weighted average of individual super-pixel descriptors in its 2-neighbourhood. We found the performance to vary based on the weights that were assigned to the neighbourhood super-pixels. In our experiments, we used a Gaussian weighting, based on depth difference between the super-pixels. We set the standard deviation for the 1-neighbours to 20 and 10 for the 2-neighbours in our experiments.

The current algorithm is computationally expensive, for example it takes about 15 minutes on the stanford dragon model on a single core CPU with our unoptimized code.

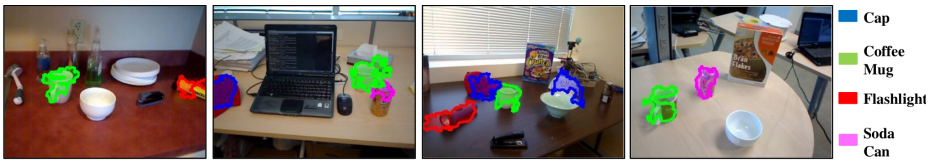


Fig. 9. Sample results of our object recognition and segmentation system. Our methods performs well in the presence of partial occlusion (cap), clutter and change in object pose. More results can be found in the supplementary material. Best viewed in colour.

5 Conclusions

In this work we presented a novel shape based feature called multi scale shape index (MSSI). This feature is a triplet of shape index, curvedness and characteristic scale. The shape index component of this feature assigns a real valued index to shapes such as umbilics (double convex), parabolics (double concave) and saddle points (convex-concave). We developed a scale-space method to compute MSSI at each discrete point at its characteristic scale from noisy 2.5D data. We studied the robustness and repeatability of this feature and demonstrated its efficacy in category recognition. Our quantitative studies indicate that the MSSI feature based recognition outperforms the current state-of-the-art method and is better able to cope with lesser training data.

Acknowledgements. This research is supported by the Boeing Company.

References

1. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. TPAMI 26(11), 1475–1490 (2004)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
3. Fabio, R.: From point cloud to surface: the modeling and visualization problem. In: International Workshop on Visualization and Animation of Reality-based 3D Models, vol. 34, p. 5 (2003)
4. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
5. Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L.: Hough transform and 3d surf for robust three dimensional classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 589–602. Springer, Heidelberg (2010)
6. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. Image and Vision Computing 10(8), 557–564 (1992)
7. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: ICRA (2011)

8. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
9. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV (2004)
10. Lindeberg, T.: Feature detection with automatic scale selection. IJCV 30(2), 79–116 (1998)
11. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: ISMAR (2011)
12. Petrelli, A., Di Stefano, L.: On the repeatability of the local reference frame for partial shape matching. In: CVPR (2011)
13. Pham, M.T., Woodford, O.J., Perbet, F., Maki, A., Stenger, B., Cipolla, R.: A new distance for scale-invariant 3d shape recognition and registration. In: ICCV (2011)
14. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels. Technical Report 149300 EPFL (June 2010)
15. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: ICRA (2011)
16. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: ICCV (2007)
17. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. TPAMI 26(5), 550–571 (2004)
18. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Computer Graphics Forum (2009)
19. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 356–369. Springer, Heidelberg (2010)
20. Unnikrishnan, R., Hebert, M.: Multi-scale interest regions from unorganized point clouds. In: CVPR Workshop on Search in 3D, S3D (2008)
21. Witkin, A.P.: Scale-space filtering. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, 329–332 (1987)
22. Yu, T.H., Woodford, O.J., Cipolla, R.: An evaluation of volumetric interest points. In: 3DIMPVT (2011)
23. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.: Surface feature detection and description with applications to mesh matching. In: CVPR (2009)

Compression of Depth Maps with Segment-Based Homogeneous Diffusion

Sebastian Hoffmann, Markus Mainberger,
Joachim Weickert, and Michael Puhl

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

Abstract. The efficient compression of depth maps is becoming more and more important. We present a novel codec specifically suited for this task. In the encoding step we segment the image and extract between-pixel contours. Subsequently we optimise the grey values at carefully selected mask points, including both hexagonal grid locations as well as freely chosen points. We use a chain code to store the contours. For the decoding we apply a segment-based homogeneous diffusion inpainting. The segmentation allows parallel processing of the individual segments. Experiments show that our compression algorithm outperforms comparable methods such as JPEG or JPEG2000, while being competitive with HEVC (High Efficiency Video Coding).

Keywords: depth map, image compression, segmentation, homogeneous diffusion inpainting, partial differential equations (PDEs).

1 Introduction

In recent years, 3D cinema technology has become increasingly popular. In the corresponding so called multi-view video + depth (MVD) format, multiple images are captured from different perspectives along with their respective depth images. To cope with the huge amount of data, an efficient compression is indispensable. Combined compression methods have been presented (see e.g. [1]), where the correlated information between the colour images and the corresponding depth maps is exploited. It is also possible to incorporate a temporal component into the compression framework. In this paper, however, we focus exclusively on the problem of depth map compression.

Besides well-established methods like JPEG or JPEG2000, compression algorithms based on partial differential equations (PDEs) recently gained attention. While in the encoding step, only a small subset of all pixels is selected and stored, the missing information is reconstructed by means of PDE-based interpolation when decoding. This idea was introduced by Galić et al. in 2005 [2] and extended in 2008 [3]. A further developed version of Schmaltz et al. [4] was

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 319–330, 2013.

able to beat JPEG2000. In the special case of cartoon-like images, which are in their nature very similar to depth maps, a much simpler and computationally favourable PDE-based codec has been proposed: In [5] the authors encode the grey or colour values on both sides of image edges. In contrast to the nonlinear anisotropic diffusion processes used in the aforementioned methods, a basic homogeneous diffusion inpainting is then sufficient to reconstruct missing pixels. For cartoon-like images the method could outperform JPEG2000. Indeed, for depth map compression, extracting and storing edges is actually a natural idea as they are crucial to obtain a good visual perception of the geometry. However, due to the fact that the above codec cannot handle homogeneous variations, it turns out that its application to depth maps leads to unsatisfactory results.

In [6, 7] modified versions of this edge-based approach have been suggested. The main changes consist of adding grey values at regular mask points, exploiting different edge detectors, and encoding parts of the extracted data with other methods. Similarly homogeneous diffusion can be incorporated in existing block-based approaches where additional edge information is used to attain sharp edges [8]. However, all these methods are either data intensive or lead to a fairly complex overall codec.

Other approaches try to split the depth image recursively into smaller parts, resulting in a tree structure, and recover the depth map on the lowest tree level by means of linear interpolation [9, 10]. In [11], the depth map is approximated by linear functions within segments. A similar method, also working on segments, has been introduced in [12] where mainly bilinear interpolation of data on a regular grid has been used to reconstruct the depth information. All these methods have the drawback that a lot of information has to be stored to be sufficiently flexible.

The goal of the present paper is to address the aforementioned problems. We present a conceptually simple codec for depth maps. While it is also based on homogeneous diffusion inpainting, it differs from [5–7] by the fact that it replaces edges by closed contours that result from a segmentation. This creates a decoupling into sub-problems and allows to benefit from parallel implementations. More importantly, by assuming homogeneous Neumann boundary conditions between segments, we show that it is unnecessary to store grey values at contours. Instead, we select hexagonal grid points as well as points at some specific locations. The corresponding grey values to be stored are optimised. In the end we do not only achieve a codec for depth maps that outperforms JPEG [13] and JPEG2000 [14], but even has the potential to compete with the substantially more complex HEVC (High Efficiency Video Coding), which is one of the most favorable methods to encode this type of images [15].

Our paper is organised as follows. First we introduce segment-based homogeneous inpainting in Section 2. Based on this concept we describe our encoding process in Section 3 and discuss the corresponding decoding steps in Section 4. Experimental results will be presented in Section 5, and a summary in Section 6 concludes the paper.

2 Segment-Based Homogeneous Diffusion

One key element of our compression codec is the *segment-based homogeneous diffusion (SBHD)* that is used to reconstruct an image from a small amount of stored data. Given a greyscale image $f(x, y)$, SBHD relies on a segmentation of the image domain into several sub-domains. For each of the segments, we assume the grey values at specific points - the so called mask points - to be given. This information is used to inpaint the rest of the respective segment.

The inpainting can be described as computing the steady state solution of the *homogeneous diffusion equation* [16]

$$\partial_t u = \Delta u \tag{1}$$

subject to the following *mixed boundary conditions*:

$$\begin{cases} u = f & \text{at mask points} \qquad (\textit{Dirichlet boundary conditions}) \\ \partial_{\boldsymbol{n}} u = 0 & \text{at segment boundaries } (\textit{homogeneous Neumann boundary cond.}) \end{cases}$$

Thereby, $\boldsymbol{n}$ is the unit normal vector to the respective segment boundary, and $\partial_{\boldsymbol{n}} u$ denotes the partial derivative of u in normal direction. The discretisation of this partial differential equation can be done in a straightforward way by using finite differences [17]. Then, as long as we have at least one mask pixel in each segment, there exists an unique solution of the discrete problem (cf. [18]).

As a result of SBHD we obtain an image containing (*i*) sharp edges at segment boundaries and (*ii*) smooth transitions within segments steered by the values at the mask points. These transitions can also represent unsharp edges. SBHD is therefore well suited for the representation of depth images.

An important advantage of this SBHD is the fact that, by construction, each mask point does not have a global influence: Its impact is limited by the respective segment boundaries. This allows a segment-wise parallel processing as we will see later. In order to get a solution of the diffusion equation we make use of the *fast explicit diffusion (FED)* scheme [19] together with a CUDA implementation on the GPU. Compared to a CPU version we achieve a substantial speedup due to the parallelism. Another speedup is gained by reducing the number of required iterations to reach a steady state. To this end, we initialise all unknown values at non-mask points with the mean value of the known grey values within each individual segment.

3 Data Extraction and Encoding

Our encoding algorithm consists of several parts that are described next.

3.1 Part 1 – Segmentation

The first step is to find a segmentation fulfilling two properties. On the one hand there should be a large contrast at the boundary between adjacent segments,

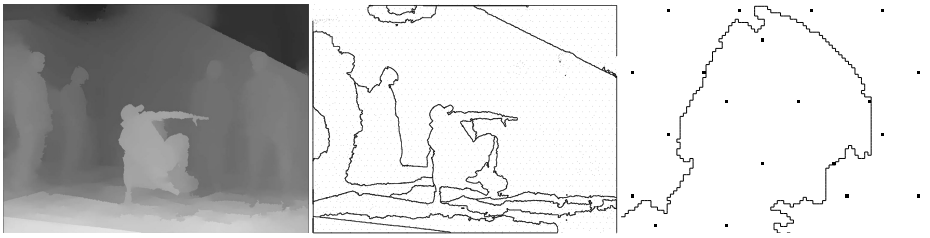


Fig. 1. Contours extraction example. **Left:** Original depth image *breakdancers* (1024 × 768). **Middle:** Extracted closed contours between pixels gained by the segmentation ($T_1 = 1$, $T_2 = 5$, $\sigma = 0.5$). **Right:** Zoom at the contour of one head. Gray values are given at specific mask points (black dots).

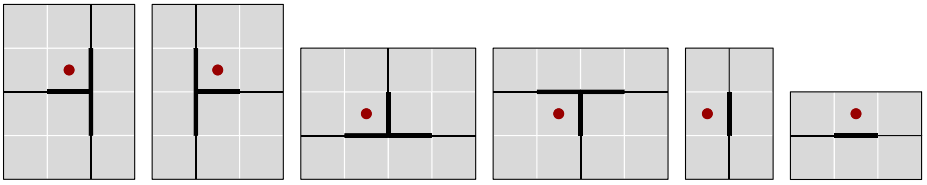


Fig. 2. Different types of edge crossings along with their respective reference point (red dot)

such that it pays off to save information at these locations. Since we store the contour information it is possible to precisely recover these sharp edges. On the other hand we want to have smooth transitions within each of the segments such that they can be reconstructed well by our SBHD inpainting from the existing information at mask points.

Our segmentation algorithm consists of two steps. First a region growing algorithm is applied to get four-connected segments. A threshold T_1 thereby determines whether or not a neighboring pixel belongs to the same segment. Small values of T_1 usually lead to a slight over-segmentation. Therefore, in a second step, we consider a Gaussian smoothed version of the original image with a standard deviation of σ. In this image we compute the mean contrast between adjacent pixels along a contour separating two neighboring segments. Successively we remove the boundary with the lowest average contrast. We repeat this as long as the lowest contrast is smaller than some threshold T_2. This method yields very precise contours between adjacent segments while allowing smooth transitions not to be split.

As already mentioned the segmentation immediately yields closed contours at between-pixel locations. An example depth image along with the extracted contours is depicted in Figure 1. As desired, there is no contour around the two people on the right hand side of the image because of the smooth transition to the background. It is the task of data points within the segments to restore such smooth flows in the reconstruction.

If we want to use an existing codec like the JBIG (*Joint Bi-level Image Experts Group*) standard [20] to encode the contour information we would have to store two binary images, i.e., the edges in x- and y-direction, respectively. Alternatively we store this information more efficiently with a chain code. The advantage of between-pixel edges is that there are only three possible directions, whereas we would have seven directions when considering pixel chains. Thus, the chain code is highly efficient.

In the beginning we extract all T-junctions in the edge map (see Fig. 2, types *i*-*iv*). To store them, we have to save the respective reference point coordinates along with the type number. Starting at these crossings we can build a chain code and stop whenever we reach an edge that has already been visited. Not needed starting elements can be removed afterwards. The only thing which remains are contours without any crossing. Therefore, we add two more starting element types representing only one edge, either in x- or in y-direction, respectively (see Fig. 2, types *v* and *vi*). The corresponding chain code has to be stored as well. Afterwards we employ a sophisticated lossless context-based entropy coder to encode the obtained edge information, namely the PAQ compression [21], version PAQ8o6.

3.2 Part 2 – Mask Points

So far we only focused on the location of segment boundaries in the image. However, we also have to encode some data for being able to reconstruct the grey values within each of the segments. Therefore, we use a mixture of regularly sampled mask points and points at freely chosen positions as described in the following subsections. The overall amount of selected mask pixels is a free parameter which allows us to steer the compression ratio.

(a) Hexagonal Mask

To reduce the coding costs for the location of prescribed grey values we initially choose points according to a specific pattern or algorithm to limit the coding overhead. One possibility would be to uniformly sample random points over the whole domain given a specific seed. A drawback of this method is that there are specific areas where points are clustered, i.e., the distance between neighboring points is not equal. Another approach would be to use mask points at a regular grid as done in [6]. However, since we want a good covering of the whole domain this is still not optimal. What we want is a mask where the minimum distance between two distinct points is maximised. To get a good approximation one could assume that no boundaries are present. In this context it is known that the hexagonal packing is the optimal one in the two-dimensional Euclidean plane [22]. This is why we make use of such an hexagonal grid pattern. In order to make sure that there is at least one mask point in each segment, we compute the hexagonal mask separately for each individual segment. Thereby a given density value determines the number of points.

Input: Original image f, initial mask m,
percentage p, percentage q, desired mask density r
Output: Mask m with density r

While density of m smaller than r do

1. Apply SBHD using the mask m and save result in u.
2. Choose randomly p percent of all non-mask points as a candidate set C.
3. At all locations in C, compute the squared difference between f and u
4. Let $T \subset C$ be the set containing the q percent of all locations from C where the error computed in Step 3 is largest.
5. Set all locations from T as mask points in m.
6. Clear T and C.

Fig. 3. Probabilistic densification algorithm

(b) Probabilistic Densificaton and Nonlocal Pixel Exchange

In addition to the hexagonal mask we want to store some more mask points for a further quality gain. We accept the larger coding costs of these free points and place them at locations where the quality can be improved most. Therefore, we perform in a first step a so-called *probabilistic densification*, which is similar to the probabilistic sparsification process described in [18]. We consecutively select points as additional mask points, starting with the ones from the hexagonal mask as an initialisation. The decision where we insert new points in each step depends on where the difference of the respective reconstruction to the original image is largest. This process is repeated until we reach a desired density. The exact algorithm is depicted in Figure 3. Experiments have shown that a good choice for the parameters is $p = 10\%$ and $q = 0.1\%$.

Moreover, it has been demonstrated that it pays off to apply a so called *nonlocal pixel exchange* [18], which tries to find better combinations for the mask points not lying on the hexagonal grid. Thereby a small number k of non-mask points is randomly selected, and the local error is computed. Then one mask point is chosen, exchanging its position with the position of the largest computed error. If this improves the reconstruction we use the new mask, otherwise the exchange is reverted. Thus, we can only improve our result. For more details we refer to [18]. We use the parameter $k = 10$, as this choice turns out to yield good results. After only 1000 iterations, the reconstruction quality can already be improved tremendously. The pixel exchange step can also be left out if one is interested in a faster encoding method with lower quality.

While only having to store one density parameter for the hexagonal mask, the location of the additional free mask points is encoded using the JBIG encoder [20].

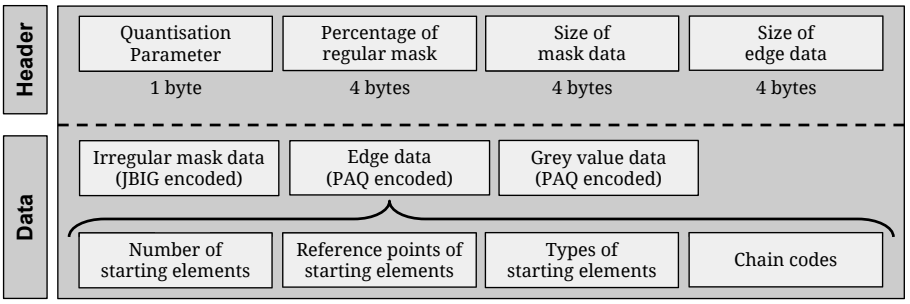


Fig. 4. File structure of our proposed codec

3.3 Part 3 – Grey Values

After the determination of the mask pixel locations it also pays off to optimise the grey values that are stored at these positions. Usually the respective grey values of the original image are adopted. However, this is not always the best solution with respect to the global reconstruction error. Instead of being fully exact at the selected pixels, it makes sense to allow for some deviation at these locations to achieve an overall smaller global error. In order to find optimal values we make use of a least squares approach as suggested in [18]. Note that, as we are using SBHD, the computations can be speeded up by computing the optimal grey values for the individual segments in parallel.

The resulting values at mask locations are then quantised to d different values and afterwards stored in a list. The order of the mask points is chosen such that we go from segment to segment, which has the advantage that subsequent values in the list are more likely to be similar due to the design of our segmentation. To obtain a compact representation, we use the PAQ encoder mentioned in Section 3.1.

3.4 Overview of File Structure

We are now able to write all the gathered information into one file having the structure as depicted in Figure 4. Note that we do not have to store the image dimensions in the header since they are already contained in the JBIG mask data.

4 Decoding

In the decoding phase we can follow a straightforward process chain. First of all we scan the stored header information and split the main data into the edge data, the irregular mask, and the grey value information. Edges can be restored by

placing the starting points and following the single contour chains. Afterwards we compute the hexagonal mask and add the irregular mask points to it. The grey values are placed at the corresponding locations, and we finally obtain the reconstruction via SBHD.

5 Experiments

In this section, the potential of the proposed codec is presented by considering three different existing methods. Besides the well-established standard JPEG and its successor JPEG2000, we will also compare our method with the designated future standard HEVC (High Efficiency Video Coding), version HM-8.2. Although this codec is designed for the purpose of video coding, it also provides an intra-coding mode for the efficient compression of still images [15].

As test images we use the so-called *breakdancers* and *ballet* depth maps, both taken from the MVD sequence in [23] (respective image size: 1024×768). We determine that 80% of all mask points lie at hexagonal grid locations and set the quantisation parameter to $d = 64$. This choice has experimentally shown to yield a good trade-off between coding costs and quality gain. We keep these parameters fixed for all experiments. Figure 5 depicts the results for a compression rate of 0.045 bits per pixel (bpp), which roughly corresponds to a compression ratio of $180 : 1$. The overall mask density in this case is 0.3% and 0.45% for the images *breakdancers* and *ballet*, respectively.

The transform-based methods JPEG and JPEG2000 often perform well when it comes to the compression of standard natural images. However, both methods suffer from artifacts around edges. When it comes to depth images, these block or ringing artifacts around object boundaries are visually perceived much more unpleasant than in smooth image regions. HEVC seems to overcome these problems, but tends to smooth out some of the edges, which can lead also to a distorted geometric perception. With our SBHD method it is hard to notice any difference between the original image and the reconstruction.

In a quantitative comparison we measure the error between the original image and the respective reconstruction by means of the peak-signal-to-noise ratio (PSNR). The resulting measurements can be seen in Figure 6. Note that for JPEG it is not possible to reach very high compression rates. As a larger PSNR value corresponds to a higher similarity between two images, one can see that our method clearly outperforms JPEG and JPEG2000. Our codec is even able to exceed the quality of HEVC for some compression rates.

Furthermore, we can also evaluate the results considering a more perceptual error measure like the structural similarity index (SSIM) [24]. We use the available MatLab version with standard parameters from [24]. Figure 7 depicts the results. Note that a SSIM closer to 1 denotes a better visual similarity. Compared to JPEG or JPEG2000, it is visible that our proposed method reaches better results for almost all tested compression rates. HEVC and SBHD give

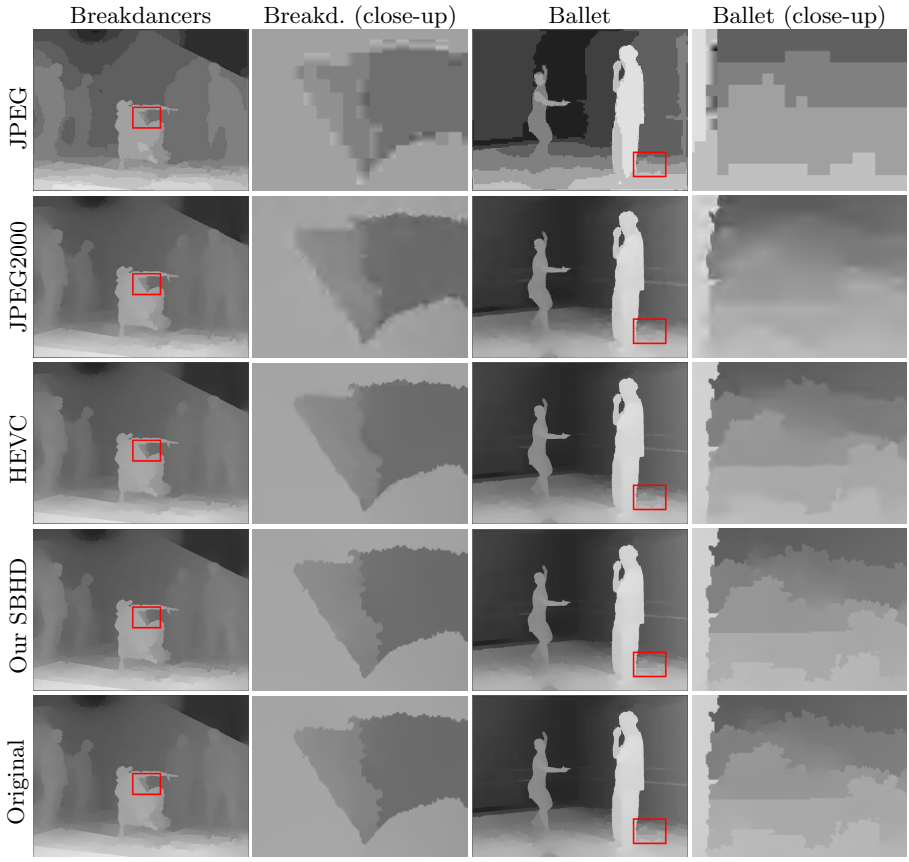


Fig. 5. Comparison of different compression methods for two depth images using a compression rate of 0.045 bpp. The boxes denote the area of the respective close-ups.

reconstructions of comparable quality. This result is remarkable since, in contrast to HEVC, our algorithm makes use of relatively simple and straightforward concepts. We thus believe that it has potential for further improvement.

In our current proof-of-concept implementation, it takes several minutes for an image to be encoded, depending on its size and the number of mask points. The decoding of a depth image of size 1024×768 can be done within a second on a modern PC. It is important to mention that there is a lot of potential for accelerating this process. For example, one can incorporate bidirectional multigrid methods into SBHD [5]. In this way, we expect that real-time decoding becomes feasible for practical applications.

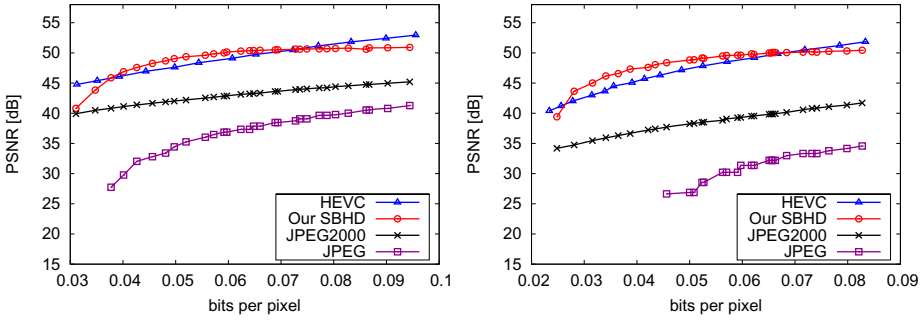


Fig. 6. Quantitative comparison to JPEG, JPEG2000 and HEVC. **Left:** *Breakdancers* image. **Right:** *Ballet* image.

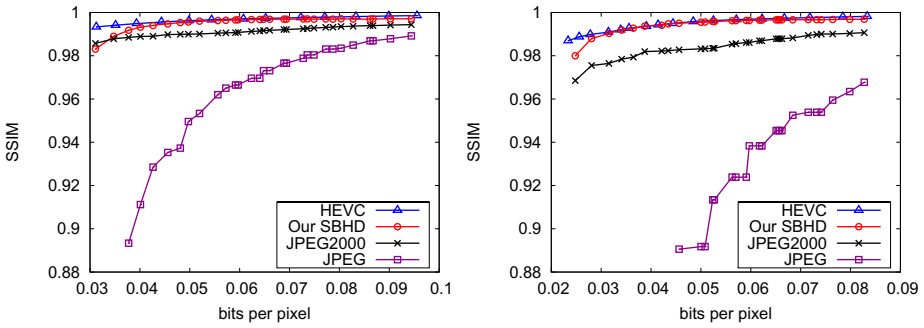


Fig. 7. Perceptual comparison to JPEG, JPEG2000 and HEVC using the SSIM measure. **Left:** *Breakdancers* image. **Right:** *Ballet* image.

6 Summary

We have shown that a combination of two relatively elementary concepts can lead to a remarkable compression quality of depth maps: a region-growing segmentation method as well as a homogeneous diffusion inpainting with carefully selected data points. In our evaluation, this segment-based homogeneous diffusion (SBHD) codec clearly outperforms JPEG and JPEG2000. Moreover, it performs competitively with HEVC, especially in terms of perceptual quality.

In our ongoing work we are extending our framework to colour-valued data such as cartoon-like images. Moreover, we are going to incorporate additional information such as multiple views, combined colour / depth images, and their temporal extensions. We are optimistic that this will help to demonstrate the widely unexplored strength of diffusion ideas for data compression.

Acknowledgements. We thank Marco Zamarin and Søren Forchhammer from the Technical University of Denmark for drawing our attention to this topic.

References

1. Ruiz-Hidalgo, J., Morros, J.R., Aflaki, P., Calderero, F., Marqués, F.: Multiview depth coding based on combined color/depth segmentation. Journal of Visual Communication and Image Representation 23(1), 42–52 (2012)
2. 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)
3. 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)
4. 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)
5. 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)
6. 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)
7. Li, Y., Sjostrom, M., Jennehag, U., Olsson, R.: A scalable coding approach for high quality depth image compression. In: 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, pp. 1–4 (October 2012)
8. Chen, J., Ye, F., Di, J., Liu, C., Men, A.: Depth map compression via edge-based inpainting. In: Picture Coding Symposium, Kraków, Poland, pp. 57–60 (May 2012)
9. Morvan, Y., de With, P.H.N., Farin, D.: Platelet-based coding of depth maps for the transmission of multiview images. In: Woods, A.J., Dodgson, N.A., Merritt, J.O., Bolas, M.T., McDowall, I.E. (eds.) Proceedings of SPIE: Stereoscopic Displays and Applications, San Jose, California, USA, vol. 6055 (January 2006)
10. Sarkis, M., Zia, W., Diepold, K.: Fast depth map compression and meshing with compressed tritree. In: Zha, H., Taniguchi, R.-I., Maybank, S. (eds.) ACCV 2009, Part II. LNCS, vol. 5995, pp. 44–55. Springer, Heidelberg (2010)
11. Jager, F.: Contour-based segmentation and coding for depth map compression. In: Visual Communications and Image Processing, Tainan City, Taiwan, 1–4 (November 2011)
12. Zanuttigh, P., Cortelazzo, G.M.: Compression of depth information for 3D rendering. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Potsdam, Germany, pp. 1–4 (May 2009)
13. Pennebaker, W.B., Mitchell, J.L.: JPEG: Still Image Data Compression Standard. Springer, New York (1992)
14. Taubman, D.S., Marcellin, M.W. (eds.): JPEG 2000: Image Compression Fundamentals, Standards and Practice. Kluwer, Boston (2002)
15. Nguyen, T., Marpe, D.: Performance analysis of HEVC-based intra coding for still image compression. In: Domanski, M., Grajek, T., Karwowski, D., Stasinski, R. (eds.) Picture Coding Symposium, Kraków, Poland, pp. 233–236 (May 2012)
16. 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)
17. Morton, K.W., Mayers, L.M.: Numerical Solution of Partial Differential Equations. Cambridge University Press, Cambridge (1994)

18. 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)
19. Grewenig, S., Weickert, J., Bruhn, A.: From box filtering to fast explicit diffusion. In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) DAGM 2010. LNCS, vol. 6376, pp. 533–542. Springer, Heidelberg (2010)
20. Joint Bi-level Image Experts Group: Information technology – progressive lossy/lossless coding of bi-level images. ISO/IEC JTC1 11544, ITU-T Rec. T.82 (1993); Final Committee Draft 11544
21. Mahoney, M.: Adaptive weighing of context models for lossless data compression. Technical Report CS-2005-16, Florida Institute of Technology, Melbourne, Florida (December 2005)
22. Chang, H.C., Wang, L.C.: A simple proof of thue's theorem on circle packing. ArXiv e-prints (September 2010)
23. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S.A.J., Szeliski, R.: High-quality video view interpolation using a layered representation. In: Hart, J.C. (ed.) ACM Transactions on Graphics, New York, USA, vol. 23, pp. 600–608 (August 2004)
24. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)

Scale Space Operators on Hierarchies of Segmentations

B. Ravi Kiran and Jean Serra

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE
{kiranr,j.serra}@esiee.fr

Abstract. A hierarchy of segmentations(partitions) is a multiscale set representation of the image. This paper introduces a new set of scale space operators or transformations on the *space of hierarchies* of partitions. An ordering of hierarchies is proposed which is endowed by an ω-ordering based on a global energy over the classes of the hierarchy. A class of Matheron semigroups are shown to exists in this ordering of hierarchies. A second contribution is the saliency transformation which fuses a saliency function corresponding to a hierarchy, with an external function, rendering a new or transformed saliency function. The results are demonstrated on the Berkeley dataset.

1 Introduction

This paper addresses the questions of synthesizing and improving hierarchies of segmentations by means of scale space operators. A hierarchy of partitions has been previously obtained, and is given. It provides a stack of coarser and coarser segmentations of the scene under study. Some external information, or "ground truth" composed of sets, drawings, auxiliary numerical functions, etc.. , may come, or not, with the hierarchy. The problem is thus twofold, and suggests to separate the situations with no external information form those with ground truth. They lead indeed to two rather different approaches.

The first one -no outside information- is based on already known techniques which extract an optimal cut form the hierarchy by minimizing some energy ω. The most often, the energy ω depends on a positive parameter [11] [4] [13] [5]. Under which conditions this parameter can be understood as a space scaler, leading to an improved hierarchy and to scale space semi-groups? This will be the matter of section 3, which is preceded by a reminder on optimal cuts in hierarchies.

The second situation involves disparate data. For answering the question "How to enrich the hierarchy with ground truths?" we have to find a common basis to express them, and from this basis, to build up a few laws of composition. The scale spacing will then intervene as distance functions associated with the ground-truths. These questions will be treated in sections 4.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 331–342, 2013.

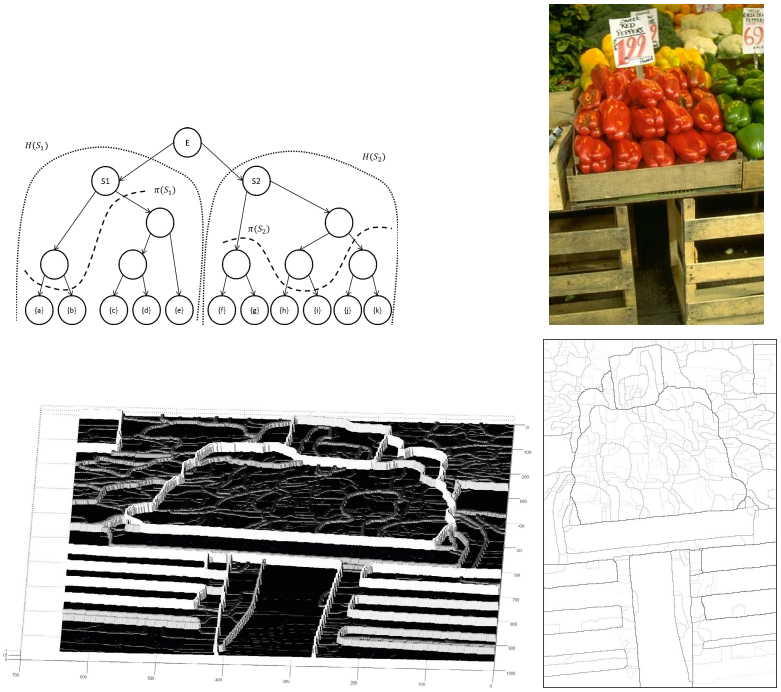


Fig. 1. Top: Dendrogram representation of hierarchy, Input 25098 Image, Bottom: Topographic view of UCM, Inverted (and contrasted for better view) Ultrametric contour Map(UCM) where the edges with strongest saliency values are the darkest, and the weakest values are the lowest, while zeros are white(background)

2 Optima Cuts and Hierarchies (Reminder)

The definitions and prerequisites needed in understanding the rest of the paper are given in this section [5], [12]. The usual distinction between continuous and digital spaces is not appropriate for the general theory developed in sections 2 to 4. What is actually needed reduces to the two following assumptions

i) the space E to partition is topological, like $\mathbb{R}^2, \mathbb{Z}^2$, or others,

ii) the smallest partition π_0 taken into account has a finite number of classes.

The first assumption allows us to speak of frontiers between classes, or edges. The second one aims to avoid things like fractal sets.

2.1 Partitions, Partial Partitions

Intuitively, a partition of E is a division of this set into classes, i.e. regions that do not overlap, and whose union gives E. Below, the symbols S, T stand for classes, and π for partitions. Partition π_1 is smaller than partition π_2 when each

class of π_1 is included in a class of π_2. This condition provides an ordering on the partitions, called refinement, which in turn induces a complete lattice.

Let S be a subset of E. Following Ch. Ronse [10], any partition $\pi(S)$ of S is called *partial partition* of support S (in short p.p.). In particular, the partial partition of S into a single class is denoted by $\{S\}$. If the q classes of the partition $\pi(S)$ are $\{T_u, 1 \leq u \leq q\}$, one writes

$$\pi(S) = T_1 \sqcup ..T_u.. \sqcup T_q,$$

where the symbol $\sqcup$ indicates that the classes are concatenated. The set of all partial partitions of E is denoted by $\mathcal{D}$.

An energy on $\mathcal{D}$ is a numerical function $\omega : \mathcal{D} \rightarrow [0, \infty]$. In the following, $\mathcal{D}$ will be provided with several energies ω, which may satisfy two axioms

i) ω is *h-increasing*, i.e.

$$\omega(\pi_1) \leq \omega(\pi_2) \quad \Rightarrow \quad \omega(\pi_1 \sqcup \pi_0) \leq \ \omega(\pi_2 \sqcup \pi_0). \tag{1}$$

where π_1 and π_2 are two partial partitions of same support, and π_0 a partial partition disjoint from π_1 and π_2 ,

ii) ω is *singular*, when the energy $\omega(\{S\})$ of class S is differs from that of any p.p. of S, i.e.

$$\pi(S) \text{ p.p. of } \{S\} \quad \Rightarrow \quad \omega(\{S\}) \neq \omega(\pi(S)). \tag{2}$$

The geometrical meaning of Rel.(1) is depicted in Figure 2.

2.2 Hierarchies of Partitions

A hierarchy H is a chain of ordered partitions π_i, i.e.

$$H = \{\pi_i, 0 \leq i \leq n \mid i \leq k \leq n \Rightarrow \pi_i \leq \pi_k\}, \tag{3}$$

where π_n is the partition $\{E\}$ of E in a single class called the *root*. The classes of the finest partition π_0 are called the *leaves*, and the intermediary classes are the *nodes*.

Let $S_i(x)$ be the class of partition π_i of H at point $x \in E$. Denote by $\mathcal{S}$ the set of all classes $S_i(x)$ of H, i.e. $\mathcal{S} = \{S_i(x), x \in E, 0 \leq i \leq n\}$. Expression (3) means that at each leaf x the family of those classes $S_i(x)$ of $\mathcal{S}$ that contain x forms a finite chain $\mathcal{S}_x$ in $\mathcal{P}(E)$, of nested elements from $S_0(x)$ to E :

$$\mathcal{S}_x = \{S_i(x), 0 \leq i \leq n\}.$$

According to a classical result, a family $\{S_i(x), x \in E, 0 \leq i \leq n\}$ of indexed sets generates the classes of a hierarchy iff

$$x, y \in E \ \Rightarrow S_i(x) \subseteq S_j(y) \text{ or } S_i(x) \supseteq S_j(y) \text{ or } S_i(x) \cap S_j(y) = \varnothing. \tag{4}$$

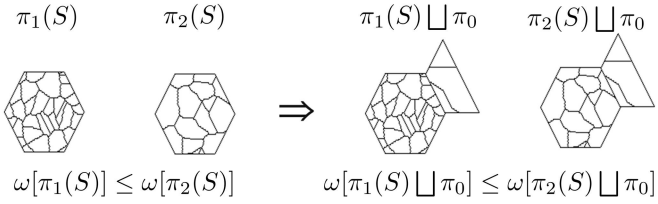


Fig. 2. h-increasingness

The partitions of a hierarchy may be represented by their classes, or by the saliency map of the edges, or again by a dendrogram where each node of bifurcation is a class S, as depicted in Figure 1. The classes of π_{i-1} at level $i-1$ which are included in class $S_i(x)$ are said to be *the sons* of $S_i(x)$. The set of all classes S of all partitions involved in H is denoted by $\mathcal{S}(H)$. Clearly, the descendants of each S form in turn a hierarchy $H(S)$ of root S, which is included in the complete hierarchy $H = H(E)$.

2.3 Cuts in a Hierarchy

Any partition π of E whose classes are taken in $\mathcal{S}$ defines a *cut* π in a hierarchy H. The set of all cuts of E is denoted by $\Pi(E) = \Pi$. Every "horizontal" section $\pi_i(H)$ at level i is obviously a cut, but several levels can cooperate in a same cut, such as $\pi(S_1)$ and $\pi(S_2)$, drawn with thick dotted lines in Figure 1. Similarly, the partition $\pi(S_1) \sqcup \pi(S_2)$ of the figure generates a cut of $H(E)$.

Given an energy ω over the set $\mathcal{D}(E)$ of the partial partitions of E, an *optimal cut* $\pi^* \in \Pi(E)$ is a cut that minimizes ω, i.e. such that $\omega(\pi^*) = \inf\{\omega(\pi) \mid \pi \in \Pi(E)\}$. Now, though the hierarchies are discrete, the number of their possible cuts becomes rapidly huge: a small hierarchy of 200 leaves and 10 levels generates billions of cuts! How to find out the best one? The following two theorems answer the question.

Theorem 1. *Let H be a hierarchy and ω be a h-increasing and singular energy. Energy ω induces an ordering on the set $\Pi(E)$ of all cuts of H. Given two cuts $\pi, \pi' \in \Pi(E)$, cut π is said to be less energetic than cut π' w.r.t. ω, and one writes $\pi \leq_\omega \pi'$, when in each class S of the refinement supremum $\pi \vee \pi'$ the p.p. of π inside S is less energetic than that of π' inside S. The energetic ordering induces the ω-lattice $(\wedge_\omega, \vee_\omega)$.*

In the notation, we distinguish the refinement lattice from the ω-lattice by using for the former the three symbols $\leq, \vee$, and $\wedge$, without ω subscript. The meaning of the energetic lattice $(\wedge_\omega, \vee_\omega)$ is clear: it associates energetic minimum and maximum with *each class* of $\pi \vee \pi'$, and not globally only.

Theorem 2. *Let ω be h-increasing and singular energy. Then for any $H \in \mathcal{H}$ and any node S of H with p sons $T_1..T_p$ of optimal cuts $\pi_1^*, ..\pi_p^*$, there exists a*

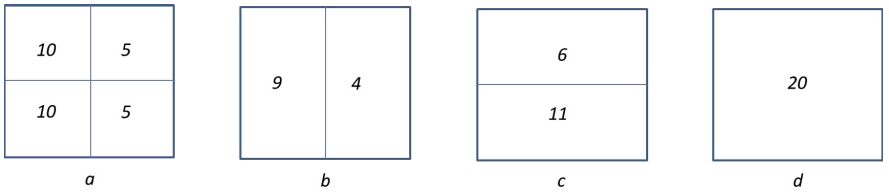


Fig. 3. The leaves are the four classes of a. The three levels of the hierarchy H_1 are [a b d] and those H_2) are [a c d], and d is the whole space. The indicated energies ω show that $H_1 \leq_\omega H_2$.

unique optimal cut of the sub-hierarchy of root S. It is either the cut $\pi_1^* \sqcup \pi_2^* .. \sqcup \pi_p^*$, or the one class partition $\{S\}$ itself:

$$\omega(\pi^*(S)) = \min\{\omega(\{S\}), \omega(\pi_1^* \sqcup \pi_2^* .. \sqcup \pi_p^*)\} \tag{5}$$

Theorem 2 governs the choices of models for energies, and their implementations:

Firstly, the dynamic programming Rel.(5) allows us to find the optimal cut of H in one ascending pass. The nodes of H above the leaves have to be visited according to an order which respects the inclusions. One then compares the energy of each node with that of the p.p. of its sons, and the less energetic of the two is kept for continuing the ascending pass, and so on until the top node E is reached [4], [5].

Secondly, the obtained optimal cut $\pi^*(E)$ is indeed globally less energetic than any other cut in H, but, moreover, if we compare π^* with any other partition π of E, then in each class S of the refinement supremum $\pi^* \vee \pi$ the energy of π^* is smaller than that of π.

3 Openings on $\mathcal{H}(\mathcal{S})$

Studies on hierarchies often hold on the family of all hierarchies whose nodes are taken among the set $\mathcal{S}$ of nodes of some initial hierarchy H, a family denoted by $\mathcal{H}(\mathcal{S})$ below. Now, optimal cutting is an operation which maps hierarchies on partitions. If we wish to insert it in a series of transformations on hierarchies, this optimal cutting must be interpreted differently.

We observe firstly that both energetic and refinement orderings on partial partitions induce orderings on the set $\mathcal{H}(\mathcal{S})$ of hierarchies, for which $H_1 \leq H_2$ when at any level i, $\pi_1(i) \leq \pi_2(i)$ (resp.$\pi_1(i) \leq_\omega \pi_2(i)$). For the refinement one, the optimal element is the cylindric hierarchy whose all horizontal sections are the leaves partition, and the maximal one is obtained by taking the one class partition $\{E\}$ at all levels, leaves level excepted. In the ω-lattice, the two extreme elements are the two cylinders H^* and H^{**} whose all sections above the leaves level are the optimal cut, or the maximal one.

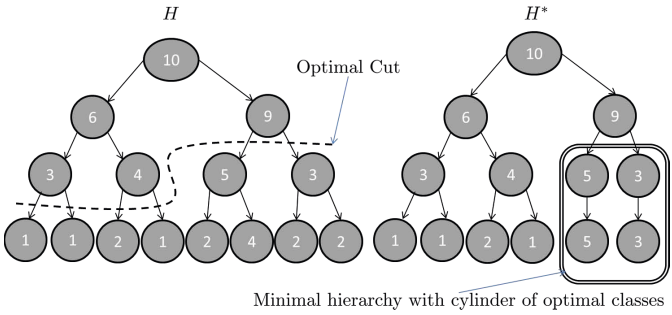


Fig. 4. Minimal pyramid H^* obtained by replacing non optimal classes in H up till level of the optimal cut

Consider now the refinement supremum $H \vee H^*$ of H and of the ω- optimal cylinder H^* and view it as an element of the ω-lattice $\mathcal{H}(\mathcal{S})$.

Theorem 3. *The operation $\gamma^*_\omega(H) = H \vee H^*$ from the ω-lattice $\mathcal{H}(\mathcal{S})$ into itself is an opening.*

Proof. γ^*_ω is anti-extensive, since each class S of H is replaced by a less energetic class of H^* when $S \leq S^*$ and left unchanged when not. On the other hand $\gamma^*_\omega[\gamma_\omega(H)] = H \vee H^* \vee H^* = \gamma^*_\omega(H)$, which is thus idempotent. Finally, γ^*_ω is also increasing since when $H \leq_\omega H'$ then each class of $H \vee H^*$ has an energy smaller or equal to that of the class of same level in $H \vee H'^*$, which achieves the proof. □

Introduce the cone $\mathcal{S}(x) = \{S_i(x), 1 \leq i \leq N\}$ of all classes of H that contain the leaf x. As x spans π_0, the cones $\{\ \mathcal{S}(x), x \in \pi_0\}$ characterize the hierarchy H. The transform $\gamma^*_\omega(H)$ can be described by its characteristic cones $\mathcal{S}^*(x)$:

$$\mathcal{S}^*(x) = \{S^*_j(x) = S^*_i(x), \quad 1 \leq j \leq i \ \ \}$$
$$\mathcal{S}^*(x) = \{S^*_j(x) = S_i(x), \quad i < j \leq N\},$$

where $S^*_i(x)$ denotes the class of the optimal cut at leaf x, and i the level at which this class is located. In the cone $\mathcal{S}^*(x)$ all classes below level $i+1$ are replaced by $S^*_i(x)$, and the other ones are those of H itself.

Instead of $H \vee H^*$, we can as well start from $H \wedge H^*$, and consider the operation $\zeta^*_\omega(H) = H \wedge H^*$, which also turns out to be an opening. In the cone at leaf x of $\zeta^*_\omega(H)$ all classes above level $i+1$ are replaced by $S^*_i(x)$, and the other ones are those of H itself.

3.1 Semi-groups of Climbing Energies on $\mathcal{H}(\mathcal{S})$

We now consider a climbing family $\{\omega(\lambda), \lambda \in \Lambda\}$ of energies, i.e. a family of h-increasing and single energies, as previously, to which we add the axiom of

scale increasingness [5]. This axiom states that if the energy $\omega(\lambda; S)$ of node is lesser than the energies $\omega(\lambda; \pi)$ for all p.p. π of support S, then the inequality remains true for the energies $\omega(\mu)$, $\lambda \leq \mu$:

$$\lambda \leq \mu \text{ and } \omega(\lambda; S) \leq \omega(\lambda; \pi) \Rightarrow \omega(\mu; S) \leq \omega(\mu; \pi), \qquad S \in \mathcal{S}. \tag{6}$$

The climbing family $\{\omega(\lambda), \lambda \in \Lambda\}$ generates a semi-group of operators. Denote by H^*_λ and H^*_μ the smallest elements of $\mathcal{H}(\mathcal{S})$ for the two $\omega(\lambda)$-lattice and $\omega(\mu)$-lattice respectively. The scale increasingness Rel.(6) implies that $H^*_\lambda \leq H^*_\mu$, or equivalently:

$$H^*_\lambda \vee H^*_\mu = H^*_\mu \qquad H^*_\lambda \wedge H^*_\mu = H^*_\lambda \tag{7}$$

for the refinement supremum and infimum. It follows that:

$$\gamma^*_{\omega(\mu)}[\gamma^*_{\omega(\lambda)}(H)] = (H \vee H^*_\lambda) \vee H^*_\mu = \gamma^*_{\omega(\mu)}(H).$$

As the two suprema commute, the optimal cut openings γ^*_ω turn out to satisfy the Matheron semi-group[1]:

$$\gamma^*_{\omega(\lambda)} \circ \gamma^*_{\omega(\mu)} = \gamma^*_{\omega(\mu)} \circ \gamma^*_{\omega(\lambda)} = \gamma^*_{\max\{\omega(\lambda),\omega(\mu)\}} \qquad \lambda, \mu > 0.$$

Concerning the dual form ζ^*_ω one finds similarly/

$$\zeta^*_{\omega(\lambda)} \circ \zeta^*_{\omega(\mu)} = \zeta^*_{\omega(\mu)} \circ \zeta^*_{\omega(\lambda)} = \zeta^*_{\min\{\omega(\lambda),\omega(\mu)\}} \qquad \lambda, \mu > 0.$$

This time, the *lower* energy imposes its law. Finally, the whole collection of the optimal cuts can appear in the synthetic hierarchy

$$H_{syn} = (...((H \vee_{\omega 1} H^*_{\lambda 1}) \vee_{\omega 2} H^*_{\lambda 2})...) \vee_{\omega p} H^*_{\lambda p}$$

which is a succession of the increasing optimal cuts of the energies $\omega 1, \omega 2, ...\omega p$.

4 Saliency Transformation

We now address the second question set in the introduction: how to merge hierarchy and ground-truth ? This time, hierarchy H is represented by its *saliency*; i.e. by a weighting function associated with the edges between classes of H [8]. For a given edge, this function, constant along the edge, is the level of H when the edge disappears. If we associate also one or more numerical functions g with the ground-truth, the merging question comes back to that of combining numerical functions for generating a new saliency.

In order to make saliencies and hierarchies equivalent notions, we consider the latter as sequences of partitions that appear at different levels, and not

[1] There are two broad classifications of scale spaces semigroups based on the underlying algebraic structure, used in scale space applications. First is the linear semigroup, based on a vector space. Second is the semigroup of Matheron's granulometries [7] which uses an underlying lattice for analysis, and where the most active transformation imposes its law.

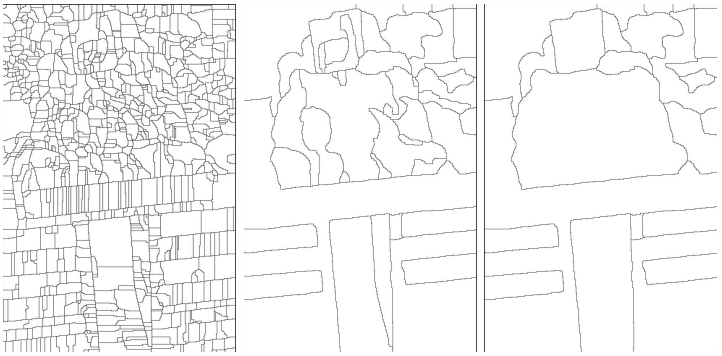


Fig. 5. A set of Optimal cuts form a Matheron semigroup: Three partitions of 25098 Image at $\lambda = 0$(leaves), 5000 and 8000

just ordered. Any strictly increasing mapping α of the levels, e.g. square root, log, etc., transforms a saliency into another one, as well as the addition by a constant value. However, a distribution of arbitrary weights on the edges may not be saliency. It is also required that by removing one edge one still maintains a partition, i.e. that one does not create pending edges. This condition is formalized below by the operation of *class opening*.

4.1 The Class Opening

This operation appeared in literature on the same date, in two independent contexts. The first is the *ultrametric opening* [6] which concerns discrete classifications by ultrametrics. The second is the *pruning* [14], which is a morphological thinning, and transforms a skeleton into a skeleton by zones of influence. More recently, in [9] the same opening allows to identify hierarchical segmentation with ultrametric watershed in digital spaces (see also [3]). Here, we start from the same notion, but more simply, without any ultrametric, or graphs or any digital background.

The difference between what follows and the three above references concerns the consequences of the class opening, namely the corollary 1, and above all the key theorem of structure 4, ignored in [6], [14], [9], and which answers the question set in the first sentence of this section. Given a finite set E of simple arcs in the $2-D$ space $\mathbb{R}^2$ or $\mathbb{Z}^2$, which can meet at their extremities only, consider the binary operation $\gamma : \mathcal{P}(\ E) \to \mathcal{P}(\ E)$ which reduces each set of arcs $X \in \mathcal{P}(\ E)$ to the closed contours it may produce.

Theorem 4. *the operation $\gamma : \mathcal{P}(E) \to \mathcal{P}(E)$ is an opening.*

Proof. Let be $X, Y \in \mathcal{P}(\ E)$. Then each closed contour of X is also a closed contour of Y, and $\gamma(X) \subseteq \gamma(Y)$. On the other hand, as $\gamma(X)$ is reduced to its contours, $\gamma\gamma(X) = \gamma(X)$. Finally, $\gamma(X) \subseteq X$, which achieves the proof. □

Fig. 6. A class opening demonstrated: Initial set of arcs, Class opening providing a partition

We call "binary class opening" the operation γ, since it selects the arcs that delineate the classes of a partition of E.

The numerical extension of γ, for which we keep the same symbol γ, holds on a numerical function g on the $2-D$ underlying space $\mathbb{R}^2$ or $\mathbb{Z}^2$. The edges of the leaves are thus formed by elements of E, points or pixels. Denote by $X_t(\ g)$ the set of pixels of the leaves where g is $\geq t$, and define the numerical opening $\gamma(\ g)$ by its level sets $X_t[\gamma(\ g)]$ by putting

$$X_t[\gamma(g)] = \gamma[X_t(g)], \quad t > 0.$$

As the number of edges is finite, the number of changes between level sets is also finite. Let S_{i+1} be a class which appears at level t_{i+1}. When t decreases, the next new class S_i appears at t_i. Since there is no change in the interval $]\ t_i, t_{i+1}]$, we have

$$t_i = \inf\{g(x) \mid x \in \partial S_i\}. \tag{8}$$

We assume that g is discrete, or lower semi-continuous, so that the value t_i occurs at one point of some edge e_i of S_i. This value is nothing but the weight of the edge e_i in the saliency transform $\gamma(\ g)$ which in turn generates hierarchy H, and t_i is the highest level of class S_i in H. If several classes appear at t_i, generated by several closing edges, then their intersections are empty and the description remains valid. Therefore, an opening being characterized by its invariants, we can state.

Corollary 1. *Let $\mathcal{G}$ be the family of all integer functions $g : \mathbb{R}^2 \to \mathbb{Z}^+$, or $\mathbb{Z}^2 \to \mathbb{Z}^+$. The image $\mathcal{I} = \gamma(\mathcal{G})$ of $\mathcal{G}$ under the class opening γ is exactly the family of all possible saliencies on the set E of the leaves edges.*

4.2 Composition of Class Openings

The composition problems are the following:

1- A first saliency, s say, already weights the set of edges E. When a non negative function g over space the underlying space $\mathbb{R}^2$ or $\mathbb{Z}^2$ is introduced, how to compose it with s?

2- When in turn a second function, g_2, acts on the saliency s_1 resulting of g_1, how the two effects are composed?

The combination of saliencies and functions is not straightforward. Given s and g, the sum, the difference, the product, the ratio, the supremum, or the infimum between s and g, may not be saliencies. The only exception arises when both s and g are saliencies. Then their supremum results in a saliency, but not the other operations. However, a few nice properties can be stated:

Theorem 5. *Let g_1 and g_2 be two non negative functions on $\mathbb{R}^2$ or $\mathbb{Z}^2$, then:*

i) $\gamma(g_1)$ (resp. $\gamma(g_2)$) is the largest saliency smaller than g_1 (resp. g_2);

ii) $\gamma(g_1) \vee \gamma(g_2)$ is the largest saliency whose value at each edge is smaller or equal to that of $\gamma(g_1)$ or $\gamma(g_2)$;

iii) if $g_1 \circledast g_2$ denotes an operation from $\mathcal{G} \times \mathcal{G} \to \mathcal{G}$, such as $+, -, \times, \div, \vee$,or $\wedge$, then $\gamma(g_1 \circledast g_2)$ is the largest saliency smaller than $g_1 \circledast g_2$, and $\gamma(g_1 \vee g_2) \leq \gamma(g_1 + g_2)$.

In all cases the resulting saliency is unique.

The proposition suggests two paths for combining saliencies. Given a primary saliency s and the ground truths $g_1, g_2, ...g_n$, the sequence $s, s \vee \gamma(g_1), s \vee \gamma(g_1) \vee \gamma(g_2)$, etc..provides an increasing family of saliencies, and the ground truths commute in the various $s \vee \gamma(g_1) \vee ...\gamma(g_i)$. Alternative families are given when we compose various g_i and then perform the class opening, namely $\gamma(s \vee g_1)$, $\gamma(s \vee g_1 \vee g_2)$, etc.. and $\gamma(s + g_1)$, $\gamma(s + g_1 + g_2)$, etc..In all cases the series is increasing, and simplify more the hierarchy $H(s)$ when suprema are involved.

Owing to the equivalence "saliencies $\Leftrightarrow$ hierarchies" all the above compositions map the whole space $\mathcal{H}$ of the hierarchies into itself. We have the succession

$$H \to \text{saliency } s \to \text{saliency } \gamma(s, g) \to \text{ new hierarchy } H'$$

We are no longer in the situation of the semi-groups of section 3, where the framework was restricted to $\mathcal{H}(\mathcal{S})$. Here new classes, absent in H, can appear in H'. The adopted approach, via the class opening, provides also the space $\mathcal{H}$ with a lattice structure isomorphic to that of the openings.

5 Experiments and Analysis

Here we demonstrate an example of the class opening on the Ultrametric contour map (UCM) from the Berkeley database [1].

5.1 Saliency Transformation by Ground Truth

Conventionally the ground truth information is intended to assess the quality of a segmentation, here a hierarchy H of segmentations. Here in the place of evaluating the hierarchy, we analyse it with respect to the given ground truth. The saliency transformation by a ground truth is an amelioration of

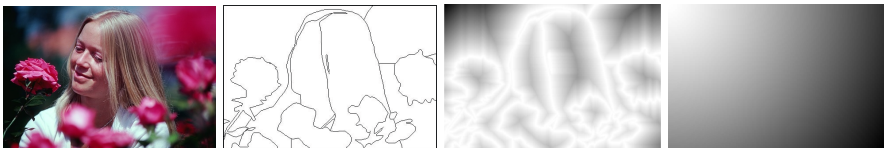


Fig. 7. 239096 Image, One of the Ground truth partitions(G_1), Inverse distance function for g_1, Point ground truth inverse distance function g_p(point at top right), where g_1 and g_p are the corresponding euclidean distance functions

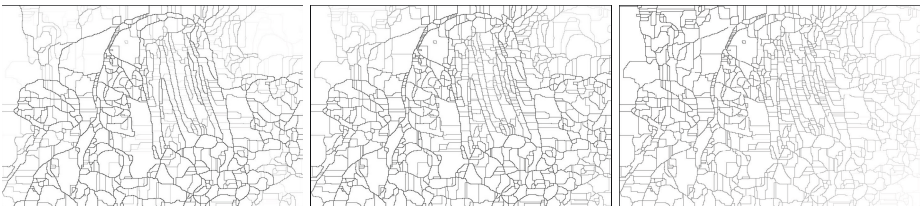


Fig. 8. Original Saliency s(Image 239096), new transformed saliency by class opening $\gamma(g_1 + s)$ with ground truth G_1. Saliency by class opening with point ground truth $\gamma(g_p + s)$ to demonstrate the effect of the inverse distance function. we see the profile of the transformed saliency $\gamma(g_p + s)$ follows the inverse distance function g_p.

the partitions in the hierarchy to generate new partitions with the same edges ordered by combined effect of: 1. proximity to the ground truth 2. high saliency. More clearly, how do we combine a ground truth and a hierarchy of partitions ?

The inputs given to us are the saliency function s representing the initial hierarchy H and the ground truth partition of edges G. Here we use the distance function of ground truth d, to define the inverse distance function $g = 1 - d$. The output is a new saliency $\gamma(s + g)$ and thus a new hierarchy H_g which contains partial partitions from H that are closest in distance to the ground truth partition G and the saliency (see figure 7).

Figure 8 summarizes the input and output saliencies. The input saliency is shown for input image 239096 from the Berkeley database. The ground truth G_1 is more or less representative of the image structure in the saliency s, and thus the resulting transformed saliency s_{G_1} is not too different, except that in general edges very far from the ground truth are reduced or weakened, while the ones in close proximity are reinforced. For the sake of pedagogy we demonstrate with a inverse distance function of a point shown in Figure 7 (g_p) and its corresponding saliency $\gamma(g_p + s)$. We see the radial attenuation in the transformed saliency.

6 Conclusion

This paper discussed two main contributions, namely: 1. The different scale space semigroups on hierarchies of partitions. 2. A saliency transform that introduces

external information into some initial hierarchy. The synthesis was obtained by means of a class opening that reduces a set of arcs containing loops into just its loops, and its numerical equivalent. An application of fusing the ground truth and saliency function was demonstrated, which reordered arcs in the hierarchy based jointly on the saliency and ground truth proximity. The distance function here can be replaced by other external information, like color and depth information, [2], thus enabling the evaluation of the hierarchy using many different functions. Following this algebraic structure, applications in multi-variable fusion and feature extraction will be explored.

Acknowledgements. The authors are grateful to Prof. L. Najman for his valuable comments on the class opening.

References

1. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour Detection and Hierarchical Image Segmentation. IEEE PAMI 33 (2011)
2. Calderero, F., Marques, F.: Hierarchical fusion of color and depth information at partition level by cooperative region merging. In: ICASSP 2009 Proceedings (2009)
3. Cousty, J., Najman, L., Serra, J.: Raising in Watershed Lattices. In: 2008 IEEE Intern. Conf. on Image Processing, ICIP 2008, San Diego, October 13-17 (2008)
4. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-Sets Image Analysis. Int. Journal of Computer Vision 68(3), 289–317 (2006)
5. Kiran, B.R., Serra, J.: Global-Local optimization on hierarchies. Pattern Recognition Letters, Special Issue (2013)
6. Leclerc, B.: Description combinatoire des ultramétriques. Mathématiques et Sciences Humaines 73, 5–37 (1981)
7. Matheron, G.: Random sets and integral geometry. John Wiley and Sons (1975) ISBN 978-0-471-57621-1
8. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Transactions on PAMI (1996)
9. Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. JMIV 40(3), 231–247 (2011)
10. Ronse, C.: Partial Partitions, Partial Connections and Connective Segmentation. JMIV 32(2), 97–125 (2008)
11. Salembier, P., Garrido, L.: Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval. IEEE Trans. on Image Processing 9(4), 561–576 (2000)
12. Serra, J.: Hierarchies and Optima. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 35–46. Springer, Heidelberg (2011)
13. Serra, J., Kiran, B.R., Cousty, J.: Hierarchies and Climbing Energies. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 821–828. Springer, Heidelberg (2012)
14. Serra, J.: Image Analysis and Mathematical Morphology, p. 397. Academic Press (1983) ISBN:0126372403

Discrete Deep Structure

Martin Tschirsich[1] and Arjan Kuijper[1,2]

[1] Technische Universität Darmstadt, Germany
[2] Fraunhofer IGD, Darmstadt, Germany

Abstract. The discrete scale space representation L of f is continuous in scale t. A computational investigation of L however must rely on a finite number of sampled scales. There are multiple approaches to sampling L differing in accuracy, runtime complexity and memory usage. One apparent approach is given by the definition of L via discrete convolution with a scale space kernel. The scale space kernel is of infinite domain and must be truncated in order to compute an individual scale, thus introducing truncation errors. A periodic boundary condition for f further complicates the computation. In this case, circular convolution with a Laplacian kernel provides for an elegant but still computationally complex solution. Applied in its eigenspace however, the circular convolution operator reduces to a simple and much less complex scaling transformation. This paper details how to efficiently decompose a scale of L and its derivative $\partial_t L$ into a sum of eigenimages of the Laplacian circular convolution operator and provides a simple solution of the discretized diffusion equation, enabling for fast and accurate sampling of L.

1 Introduction

The concept of deep structure – the way critical points and structures change under influence of scale – in continuous Gaussian scale space was introduced by Koenderink [1] and has been developed later on [2, 3]. It has proven to reveal information useful for various tasks such as image matching and retrieving, reconstruction and topological partitioning [4]. Practical applications however are rare. An implementation of important scale space based algorithms can be found in the software tool ScaleSpaceViz [5]. ScaleSpaceViz has, according to the authors, "proven to be useful in exploring the deep structure of images and constructing applications involving scale space interest points, such as reconstruction and matching" [4, 6, 7]. Although this holds true under certain conditions, ScaleSpaceViz suffers from robustness problems [8]. As it is the case with many such scale space applications, its implementation is based on a discretized continuous scale space. The instability of tracts at higher scales is due to fluxes and interaction in-between critical paths. This interaction problem became clearly apparent when aiming for image-editing via scale space singularities [4]. In [7] the top-point reconstructions were kept on the bounded domain suffering much less from these instabilities, but still the mutual fluxes in between critical paths were not taken into account in the implementations.

The discrete scale space proposed by Lindeberg [9] takes the discrete nature of computer processed signals into account. It is based on equivalent assumptions and axioms

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 343–354, 2013.

that have been used to derive the continuous Gaussian scale space adapted to discrete signals. Transferring scale space algorithms from a discretized continuous to the discrete scale space will eventually lead to more accurate, robust and possibly faster implementations.

Using so-called Laplacian Eigenimages [10], we compute a discrete scale space. We show how one can do this fast and discuss the consequences in deep structure: the movement of critical points as scale changes. The discrete scale space formalized by Lindeberg does not respect important topological invariants such as the Euler number. Since most algorithms that operate on the deep structure of the Gaussian scale space require this topological invariant to hold, we use a six-neighborhood respecting the Euler number [11]. A subsequent investigation of various properties of this discrete scale space then results in a fast and robust sampling algorithm. We finally propose the application of topological graphs [12] together with adaptive sampling in order to reliably extract the deep structure of the discrete scale space.

2 Discrete Scale Space

For periodic discrete signals f with discrete domain $D(f) = [1, M] \times [1, N]$, the diffusion equation $\partial_t L = \nabla_5^2 L$ can be written as a circular convolution with finite Laplacian kernel

$$\partial_t L = \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} \circledast L = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \circledast L + \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \circledast L$$

where $\circledast$ denotes the circular convolution operator. In matrix form, this translates to a direct summation of two substantially smaller matrices. The discrete circular convolution is a linear operator and can be expressed in matrix form if we consider $L(\cdot, \cdot; t)$ to designate a vector. Scale $L(\cdot, \cdot; t)$ of the scale space representation L can be represented as a vector $\mathbf{L}(t) \in \mathbb{R}^{MN}$ with $\mathbf{f} = \mathbf{L}(0)$.

$$\mathbf{L}(t) = \begin{bmatrix} L(1, 1; t) \\ L(1, 2; t) \\ \vdots \\ L(M, N; t) \end{bmatrix} \in \mathbb{R}^{MN}$$

For periodic f, the diffusion equation can be written in matrix form as $\partial_t L = \nabla_5^2 L \Leftrightarrow \partial_t \mathbf{L} = \Delta_{M,N} \mathbf{L}$, where $\Delta_{M,N} \in \mathbb{R}^{MN \times MN}$ denotes a circulant block matrix corresponding to the Laplacian operator ∇_5^2. It can be written as the direct sum of two ∇_3^2 operators $\Delta_M \in \mathbb{R}^{M \times M}$ and $\Delta_N \in \mathbb{R}^{N \times N}$ by $\Delta_{M,N} = \Delta_M \oplus \Delta_N = (\Delta_M \otimes \mathbf{I}_N) + (\mathbf{I}_M \otimes \Delta_N)$, where Δ_M and Δ_N are the matrix representations of the row wise applied central difference operator of second order. They differ only in their dimensions. $\otimes$ denotes the Kronecker product. For $M \geq 3$, Δ_M has the form of a Toeplitz matrix.

$$\Delta_M = \begin{bmatrix} -2 & 1 & & 1 \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ 1 & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{M \times M}$$

Each eigenvector $\mathbf{u}_{i,j}$ of $\Delta_{M,N}$ can be expressed as the outer product of two eigenvectors $\mathbf{v}_i$ and $\mathbf{w}_j$ of Δ_M and Δ_N. The corresponding eigenvalue $\lambda_{i,j}$ is then the sum of the corresponding eigenvalues υ_i and ω_j of Δ_M and Δ_N, i.e. $\Delta_{M,N}\mathbf{u}_{i,j} = \lambda_{i,j}\mathbf{u}_{i,j} \Leftrightarrow (\Delta_M \oplus \Delta_N)(\mathbf{v}_i \otimes \mathbf{w}_j) = (\upsilon_i + \omega_j)(\mathbf{v}_i \otimes \mathbf{w}_j)$: Let $\Delta_M \mathbf{v}_i = \upsilon_i \mathbf{v}_i$ and $\Delta_N \mathbf{w}_j = \omega_j \mathbf{w}_j$. Then $\mathbf{v}_i \otimes \mathbf{w}_j$ is an eigenvector of $\Delta_{M,N}$ with eigenvalue $\upsilon_i + \omega_j$, since

$$\begin{aligned} \Delta_{M,N}(\mathbf{v}_i \otimes \mathbf{w}_j) &= (\Delta_M \oplus \Delta_N)(\mathbf{v}_i \otimes \mathbf{w}_j) \\ &= (\Delta_M \otimes \mathbf{I}_N)(\mathbf{v}_i \otimes \mathbf{w}_j) + (\mathbf{I}_M \otimes \Delta_N)(\mathbf{v}_i \otimes \mathbf{w}_j) \\ &= (\Delta_M \mathbf{v}_i \otimes \mathbf{w}_j) + (\mathbf{v}_i \otimes \Delta_N \mathbf{w}_j) \\ &= (\upsilon_i \mathbf{v}_i \otimes \mathbf{w}_j) + (\mathbf{v}_i \otimes \omega_j \mathbf{w}_j) \\ &= (\upsilon_i + \omega_j)(\mathbf{v}_i \otimes \mathbf{w}_j). \end{aligned}$$

Δ_M and Δ_N are symmetric and have M respective N orthogonal eigenvectors. Then there are $M \cdot N$ orthogonal combinations $\mathbf{v}_i \otimes \mathbf{w}_j$ which are eigenvectors of $\Delta_{M,N}$. Since $\Delta_{M,N}$ has only $M \cdot N$ eigenvectors, all eigenvectors have been found, that is, each eigenvector $\mathbf{u}_{i,j}$ of $\Delta_{M,N}$ can be written as $\mathbf{u}_{i,j} = \mathbf{v}_i \otimes \mathbf{w}_j$ with corresponding eigenvalue $\lambda_{i,j} = \upsilon_i + \omega_j$ for a suitable numbering of $\upsilon_i + \omega_j$.

Δ_M and Δ_N are still sparse and symmetric matrices. In analogy to the previous step, the Laplacian kernel can be further split into a combination of discrete forward and backward difference kernels.

The matrix $\Delta_{M,N} \in \mathbb{R}^{MN \times MN}$ can be rewritten via $(\Delta_M \oplus \Delta_N)\,\mathbf{L}$ as the direct sum of the discrete forward and backward difference operators ∂_M^F and ∂_M^B in $\mathbb{R}^{M \times M}$ respecively ∂_N^F and ∂_N^B in $\mathbb{R}^{N \times N}$:

$$\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \circledast L - \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \circledast L + \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \circledast L - \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} \circledast L = \left(\partial_M^F \partial_M^B \oplus \partial_N^F \partial_N^B\right) \mathbf{L}$$

For $M \geq 3$, ∂_M^F and ∂_M^B take the form of circulant matrices.

$$\partial_M^F = \begin{bmatrix} -1 & 1 & & \\ & -1 & \ddots & \\ & & \ddots & 1 \\ 1 & & & -1 \end{bmatrix} \in \mathbb{R}^{M \times M}, \partial_M^B = \begin{bmatrix} 1 & & & -1 \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{M \times M}$$

The forward difference matrix ∂_M^F can also be written as a column- or rowwise cyclic shift of the backward difference matrix $\partial_M^F = \mathbf{T}_M \partial_M^B = \partial_M^B \mathbf{T}_M$ with

$$\mathbf{T}_M = \begin{bmatrix} & 1 & \\ & \ddots & \\ & & 1 \\ 1 & & \end{bmatrix} \in \mathbb{R}^{M \times M}$$

being a cyclic permutation matrix. Alternatively, we can write $\partial_M^F = \mathbf{T}_M - \mathbf{I}_M$ which then leads to

$$\Delta_M = \partial_M^F \partial_M^B = (\mathbf{T}_M - \mathbf{I}_M)\, \partial_M^B = \partial_M^F - \partial_M^B. \tag{1}$$

Using Eq. (1) we get

$$\begin{aligned} &\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \circledast L - \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \circledast L + \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \circledast L - \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} \circledast L \\ \Leftrightarrow\ &\left(\partial_M^F \otimes \mathbf{I}_N\right) \mathbf{L} - \left(\partial_M^B \otimes \mathbf{I}_N\right) \mathbf{L} + \left(\mathbf{I}_M \otimes \partial_N^F\right) - \left(\mathbf{I}_M \otimes \partial_N^B\right) \\ =\ &\left(\left(\partial_M^F - \partial_M^B\right) \otimes \mathbf{I}_N\right) \mathbf{L} + \left(\mathbf{I}_M \otimes \left(\partial_N^F - \partial_N^B\right)\right) \mathbf{L} \\ =\ &\left(\partial_M^F \partial_M^B \otimes \mathbf{I}_N\right) \mathbf{L} + \left(\mathbf{I}_M \otimes \partial_N^F \partial_N^B\right) \mathbf{L} \\ =\ &\left(\partial_M^F \partial_M^B \oplus \partial_N^F \partial_N^B\right) \mathbf{L}. \end{aligned}$$

We will later see that ∂_M^F and ∂_M^B have the same eigenvectors. The eigenvectors of $\Delta_M = \partial_M^F \partial_M^B$ then are identical to those of ∂_M^F and its eigenvalues are $\upsilon_i = \delta_{M,i}^F \delta_{M,i}^B$. The matrices ∂_M^F and ∂_M^B are both real symmetric and thus diagonalisable. They commute and are therefore simultaneously diagonalisable

$$\begin{aligned} U \partial_M^F U^\dagger &= \operatorname{diag}\left(\lambda_{M,1}^F, \ldots, \lambda_{M,M}^F\right) \\ U \partial_M^B U^\dagger &= \operatorname{diag}\left(\lambda_{M,1}^B, \ldots, \lambda_{M,M}^B\right) \end{aligned}$$

with $\lambda_{M,i}^F$ and $\lambda_{M,i}^B$ denoting the eigenvalues of ∂_M^F and ∂_M^B. From Eq. (1) it follows that $U \partial_M^F \partial_M^B U^\dagger = U \left(\partial_M^F - \partial_M^B\right) U^\dagger = \operatorname{diag}\left(\lambda_{M,1}^F - \lambda_{M,1}^B, \ldots, \lambda_{M,M}^F - \lambda_{M,M}^B\right)$. For the eigenvalues then holds

$$\lambda_{M,i}^F - \lambda_{M,i}^B = \lambda_{M,i}^F \lambda_{M,i}^B \Rightarrow \lambda_{M,i}^B = \frac{\lambda_{M,i}^F}{\left(\lambda_{M,i}^F + 1\right)}$$

Using the properties of the simultaneous diagonalisation, we can now express the eigenvalues υ_i of $\Delta_M = \partial_M^F \partial_M^B$ uniquely in terms of $\lambda_{M,i}^F$.

$$\upsilon_i = \lambda_{M,i}^F \lambda_{M,i}^B = \frac{\left(\lambda_{M,i}^F\right)^2}{\left(\lambda_{M,i}^F + 1\right)}$$

It still remains to calculate the eigenvalues $\lambda_{M,i}^F$ using Eq. (1) and $\det\left(\partial_M^F - \lambda_{M,i}^F \mathbf{I}_M\right) = \det\left(\mathbf{T}_M - \mathbf{I}_M - \lambda_{M,i}^F \mathbf{I}_M\right) = \det\left(\mathbf{T}_M - \left(\lambda_{M,i}^F + 1\right) \mathbf{I}_M\right) = \det\left(\mathbf{T}_M - \theta_i \mathbf{I}_M\right)$ where

$\lambda^F_{M,i} = \theta_i - 1$ with θ_i denoting the eigenvalues of $\mathbf{T}_M$. $\mathbf{T}_M$ is a circulant matrix and the eigenvectors and eigenvalues of these matrices are well known. The i-th eigenvalue of a circulant matrix

$$C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_1 \\ c_1 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ c_{M-1} & \cdots & \cdots & c_0 \end{bmatrix} \in \mathbb{R}^{M \times M}$$

is known to be of value $c_0 + c_{M.-1}\varphi_i^1 + \ldots + c_1\varphi_i^{M-1}$ with $\varphi = e^{\left(2\pi\iota\frac{i}{M}\right)}$ where ι denotes the imaginary unit. The i-th eigenvector of C is given by $\left[\varphi_i^0, \ldots, \varphi_i^{M-1}\right]$. The eigenvectors of C and C^T are equivalent. The eigenvalues $\lambda^F_{M,i}$ are then given by $\lambda^F_{M,i} = \theta_i - 1 = \varphi - 1$. The eigenvectors $d^F_{M,i}$ are identical to those of T_M and given by $d^F_{M,i} = \left[\varphi^0, \ldots, \varphi^{M-1}\right]$. The eigenvectors $d^B_{M,i}$ are identical $d^F_{M,i}$ and $\mathbf{u}_{i,j} = v_i \otimes w_j = d^F_{M,i} \otimes d^F_{N,j}$: Using orthonormality of T_M, we have $\partial^B = \partial^F \left(T_M\right)^T = \left(T_M - I_M\right) \cdot \left(T_M\right)^T = T_M \left(T_M\right)^T - \left(T_M\right)^T = I_M - \left(T_M\right)^T = -\left(\partial^F\right)^T$. Since the eigenvectors of $\partial^B = I_M - \left(T_M\right)^T$ are equivalent to the eigenvectors of $-\left(T_M\right)^T$ and thus T_M , they are equivalent to the eigenvectors of ∂^F.

As a result we have an analytic formula expressing the eigenvalues $\lambda_{i,j}$ and eigenvectors $\mathbf{u}_{i,j}$ of the Laplacian matrix $\Delta_{M,N}$.

$$\lambda_{i,j} = v_i + \omega_j = \frac{\left(\lambda^F_{M,i}\right)^2}{\left(\lambda^F_{M,i} + 1\right)} + \frac{\left(\lambda^F_{N,j}\right)^2}{\left(\lambda^F_{N,j} + 1\right)} = \frac{\left(e^{\left(\frac{2\pi\iota}{M}i\right)} - 1\right)^2}{e^{\left(\frac{2\pi\iota}{M}i\right)}} + \frac{\left(e^{\left(\frac{2\pi\iota}{N}j\right)} - 1\right)^2}{e^{\left(\frac{2\pi\iota}{N}j\right)}}$$

$$\mathbf{u}_{i,j} = \left[e^{\left(\frac{2\pi\iota}{M}i\right)^0}, \ldots, e^{\left(\frac{2\pi\iota}{M}i\right)^{M-1}}\right] \otimes \left[e^{\left(\frac{2\pi\iota}{N}j\right)^0}, \ldots, e^{\left(\frac{2\pi\iota}{N}j\right)^{N-1}}\right]$$

2.1 Efficient Time Evolution

The restriction of a scale space representation $\mathbf{L}(t)$ to a fixed scale t can be written as a weighted sum of eigenimages of the Laplacian operator, i.e. as a scalar product of the orthonormal eigenvectors $\mathbf{u}_{i,j}$ of $\Delta_{M,N}$ and the scalar coefficients $c_{i,j}(t) = \langle \mathbf{L}(t), \mathbf{u}_{i,j} \rangle$ resulting from the projection of $\mathbf{L}(t)$ to $\mathbf{u}_{i,j}$: $\mathbf{L}(t) = \sum_{i,j} c_{i,j}(t)\, \mathbf{u}_{i,j}$.

Its partial derivative $\partial_t \mathbf{L}(t)$ can then be computed from scaling each projected component separately by the corresponding eigenvalue.

$$\partial_t \mathbf{L}(t) = \mathbf{U}\Lambda\mathbf{U}^T\mathbf{L}(t) = \sum_{i,j} c_{i,j}(t)\, \lambda_{i,j}\mathbf{u}_{i,j}$$

This implicit change of base allows us to give a simple solution for the discretized diffusion equation. using $\partial_t L(t) = \Delta_5^2 L(t) \Leftrightarrow \partial_t \mathbf{L}(t) = \Delta_{MN}\mathbf{L}(t)$:

$$\sum_{i,j} \partial_t c_{i,j}(t)\, \mathbf{u}_{i,j} = \sum_{i,j} c_{i,j}(t)\, \lambda_{ij}\mathbf{u}_{i,j}$$

Multiplying both sides with $\mathbf{u}_{k,l}$ and exploiting the orthonormality $\langle \mathbf{u}_{i,j}, \mathbf{u}_{k,l} \rangle = \delta_{i,k}\delta_{j,l}$, where δ represents the Kronecker delta, gives us the partial derivate $\partial_t f$ projected onto eigenvector $\mathbf{u}_{k,l}$. This differential equation can easily be solved for $c(t)$: $\left\langle \mathbf{u}_{k,l}, \sum_{i,j} \partial_t c_{i,j}(t) \mathbf{u}_{i,j} \right\rangle = \left\langle \mathbf{u}_{k,l}, \sum_{i,j} c_{i,j}(t) \lambda_{i,j} \mathbf{u}_{i,j} \right\rangle$, thus $\partial_t c_{k,l}(t) = c_{k,l}(t) \lambda_{k,l}$ and consequently $c_{k,l}(t) = \exp(\lambda_{kl} t) c_{k,l}(0)$. The scale space representation $\mathbf{L}$ is the solution of the discretized diffusion equation and has the form

$$\mathbf{L}(t) = \sum_{i,j} c_{i,j}(t) \mathbf{u}_{i,j} = \sum_{i,j} \exp(\lambda_{i,j} t) c_{i,j}(0) \mathbf{u}_{i,j}$$

with scalar coefficients $c_{i,j}(t) = \langle \mathbf{L}(t), \mathbf{u}_{i,j} \rangle$. In matrix representation, the solution simplifies to $\mathbf{L}(t) = \mathbf{Q} \exp(\mathbf{\Lambda} t) \mathbf{Q}^T \mathbf{L}(0)$, with $\mathbf{Q}$ the orthogonal matrix mapping the standard basis to the orthonormal basis of eigenvectors.

The computational complexity needed to compute scale $\mathbf{L}(t)$ using naive matrix multiplication is in $O\left((MN)^3\right)$, since each matrix involved is of dimension $MN \times MN$. Using more efficient matrix multiplication algorithms, the complexity can be reduced to $O\left((MN)^k\right)$ for some k between $2 \ll k \leq 3$, but it cannot be faster than $\Omega\left((MN)^2 \log MN\right)$ [13].

However, using the results from the previous section, namely the relation $\mathbf{u}_{ij} = \mathbf{v}_i \otimes \mathbf{w}_j$, the computational complexity can be reduced by separating $\sum_{i,j}$ with $i = 1, \ldots, M$ and $j = 1, \ldots, N$ into two individual sums.

$$\mathbf{L}(t) = \sum_{i,j} e^{(\upsilon_i t + \omega_j t)} c_{i,j}(0) (\mathbf{v}_i \otimes \mathbf{w}_j) = \sum_i e^{(\upsilon_i t)} \mathbf{v}_i \otimes \sum_j e^{(\omega_j t)} c_{i,j}(0) \mathbf{w}_j.$$

First we have to compute the coefficients $c_{i,j}(0)$, given $\mathbf{v}_i$ and $\mathbf{w}_j$:

$$c_{i,j}(0) = \langle \mathbf{f}, \mathbf{u}_{i,j} \rangle = \sum_{k=1}^{MN} \mathbf{f}(k) (\mathbf{v}_i \otimes \mathbf{w}_j)(k).$$

Now we can separate the computation of $\mathbf{L}(t)$ into two steps:

$$\mathbf{g}_i(t) = \sum_j \exp(\omega_j t) c_{i,j}(0) \mathbf{w}_j$$
$$\mathbf{L}(t) = \sum_i \exp(\upsilon_i t) \mathbf{v}_i \otimes \mathbf{g}_i(t).$$

The scalar coefficients $c_{i,j}(0)$ are independent of scale t. Since we require usually more than one sampled scale, it is useful to precompute these coefficients in a preliminary step and then compute the individual scales using the cached results. The number of steps required to compute all $c_{i,j}$ for $i = 1, \ldots, M$ and $j = 1, \ldots, N$ is bounded above by $O\left((MN)^2\right)$. Given $c_{i,j}(0)$, computing $\mathbf{g}_i(t)$ is of complexity $O\left(N^2\right)$ and the combined complexity of $\sum_i \mathbf{g}_i(t)$ is $O\left(MN^2\right)$. Given $\mathbf{g}_i(t)$, the computation of

$\mathbf{L}(t)$ is then bounded by $O\left(MN^2\right)$. Thus, the overall complexity of computing a scale $\mathbf{L}(t)$ given $c_{i,j}(0)$ is bounded by $O\left(MN^2\right)$ which is significantly better compared to $\Omega\left((MN)^2 \log MN\right)$.

As stated in the introduction to the previous section, the intuitive way to compute a scale $L(\cdot,\cdot;t)$ is by convolution with the scale space kernel. A naive implementation however is inefficient. Far better results are obtained using the Fast Fourier Transformation. Truncating the scale space kernel and using the circular convolution theorem, it is possible to reduce the computational complexity for each scale to $O(MN \log_2 MN)$ [14]. However, regardless of its effectiveness, this method has the apparent disadvantage of introducing artificial truncation errors in addition to the inevitable rounding errors. The 2D-FFT of f itself can be computed in a preliminary step and is of complexity $O\left(M^2N^2 \log_2 MN\right)$ [15].

2.2 Additional Properties

The eigenvalues $\lambda_{i,j}$ of $\Delta_{M,N}$ are real, $\lambda_{i,j} \leq 0$ and there is exactly one $\lambda_{i,j} = 0$. $\Delta_{M,N}$ equals the negative Laplacian matrix Δ_G of the simple graph representation G_f of signal f, obtained by using a symmetric 6-neighborhood on the discrete grid. These matrices are known to have some interesting properties:

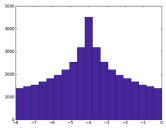


Fig. 1. Eigenvalue distribution for $M, N = 200$

- Δ_G is always positive-semidefinite, thus all eigenvalues of $\Delta_{M,N}$ are negative or 0.
- The number of times 0 appears as an eigenvalue in Δ_G is the number of connected components in G_f. Thus, there is exactly one eigenvalue having value 0.
- The second smallest eigenvalue of Δ_G is the algebraic connectivity of G_f bounded above by the traditional graph connectivity. It equals the biggest non-zero eigenvalue of $\Delta_{M,N}$.

The analytic formula for the eigenvalues confirms these facts. All $\lambda_{i,j}$ are symmetrically and unimodally distributed over $[-8, 0] \in \mathbb{R}$ (Fig. 1) while the smallest and biggest eigenvalue occur exactly once, as in the bounded domain case [7].

3 Deep Structure of the Discrete Scale Space

The kernel of the linear Laplacian operator $\Delta_{M,N}$, $\ker(\Delta_{M,N}) = \{\mathbf{L} : \Delta_{M,N}\mathbf{L} = 0\}$, consists of scales or arbitrary signals $\mathbf{L}$ with $\partial_t \mathbf{L}(t) = 0$ for every point in its domain. Those signals are also called harmonic. $\ker(\Delta_{M,N})$ equals the one dimensional 0-eigenspace of $\Delta_{M,N}$, its base is the eigenimage to the eigenvalue 0. On a finite connected graph such as the graph representation G_f of $L(\cdot,\cdot;t_0)$, harmonic functions and thus the forementioned eigenimage must be constant [16]. Repeated averaging or application of the Laplacian operator does not affect the average intensity $f_{avg} = \frac{1}{MN}\sum_{(x,y)\in D(\mathbf{f})} f(x,y)$. For increasing scale t, $\mathbf{L}(t)$ converges in every dimension against f_{avg} (Fig. 2 top).

$$\begin{aligned}\lim_{t\to\infty}\mathbf{L}\left(t\right) &= \lim_{t\to\infty}\sum_{i,j}\left(e^{\lambda_{i,j}t}c_{i,j}\left(0\right)\mathbf{u}_{i,j}\right)\\ &= c_{M,N}\left(0\right)\mathbf{u}_{M,N} + \lim_{t\to\infty}\sum_{i,j\neq M,N}\left(e^{\lambda_{i,j}t}c_{i,j}\left(0\right)\mathbf{u}_{i,j}\right)\\ &= \left\langle\mathbf{L}\left(0\right),\mathbf{u}_{M,N}\right\rangle\mathbf{u}_{M,N} = \mathbf{f}_{avg}\end{aligned}$$

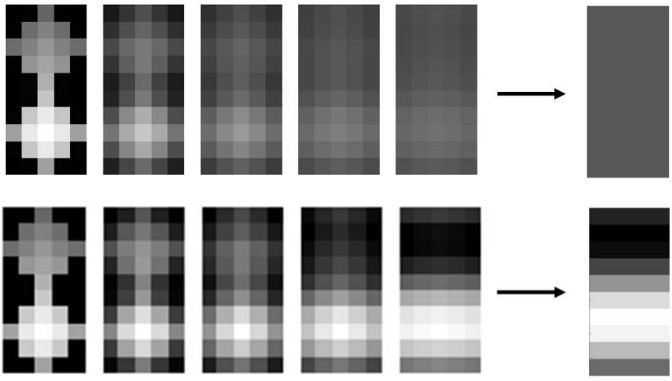


Fig. 2. Top: For increasing scale t, $\mathbf{L}\left(t\right)$ converges to the average $\mathbf{f}_{avg}$. Bottom: For increasing scale t, the topology of $\mathbf{L}\left(t\right)$ converges to the topology of the most significant eigenimages (right). The range of each scale is normalized to $[0,1]$.

For increasing scale t, the topology of $\mathbf{L}\left(t\right)$ is dominated by the eigenimages $\mathbf{u}_{i,j}$ with the biggest non-zero eigenvalues having non-zero coefficient $c_{i,j}\left(0\right)$ (Fig. 2 bottom).

3.1 Critical Point Drift and Critical Curves

Analysing the evolution of non-degenerate spatial critical points as scale changes leads to trajectories of critical points in scale space. In the continuous scale space, critical points can be traced over scale. They form so called critical curves, one-dimensional manifolds satisfying $\nabla L = 0$. Using the implicit function theorem, it can easily be shown that for a non-degenerate critical point, there exists an open interval over scale in which the point can be uniquely identified. Additional properties such as the drift velocity of critical points can then be derived from the trajectory.

In discrete scale space however, the meaning of drift velocity is unclear [9]. Spatial movement of critical points can only happen in discrete steps. The drift will have a tendency to move along the Cartesian grid lines violating rotation covariance. Also, the implicit function theorem cannot be applied to discrete functions, even though there apparently is a relation between between critical points in discrete scale space.

Using the results from the previous section however, we can show that for a spatial critical point (x, y) at scale t_0 there exists an open interval over scale in which the point

can be uniquely identified as long as its initial neighborhood at t_0 does not contain plateaus, i.e. connected points of similar value.

At scale t_0, (x, y) is a valid critical point with with value $\mathbf{L}_{x,y}(t_0)$. Sufficiently small changes in scale do not invalidate the neighborhood based critical point definition. For the neighborhood based critical point criterion to become invalid at (x, y), topological changes must occur. Therefore, we can guarantee the critical point criterion to hold as long as we do not observe a zerocrossing in the pair-wise differences

$$\mathbf{L}_{x_1,y_1}(t) - \mathbf{L}_{x_2,y_2}(t)$$

of two connected points (x_1, y_1) and (x_2, y_2) in the 6-neighborhood around (x, y) with

$$\mathbf{L}_{x,y}(t) = \sum_{i,j} \left(e^{\lambda_{i,j} t} c_{i,j}(0) (\mathbf{u}_{i,j})_{x,y} \right)$$

denoting the value of point (x, y) at scale t, see Figs. 3-4.

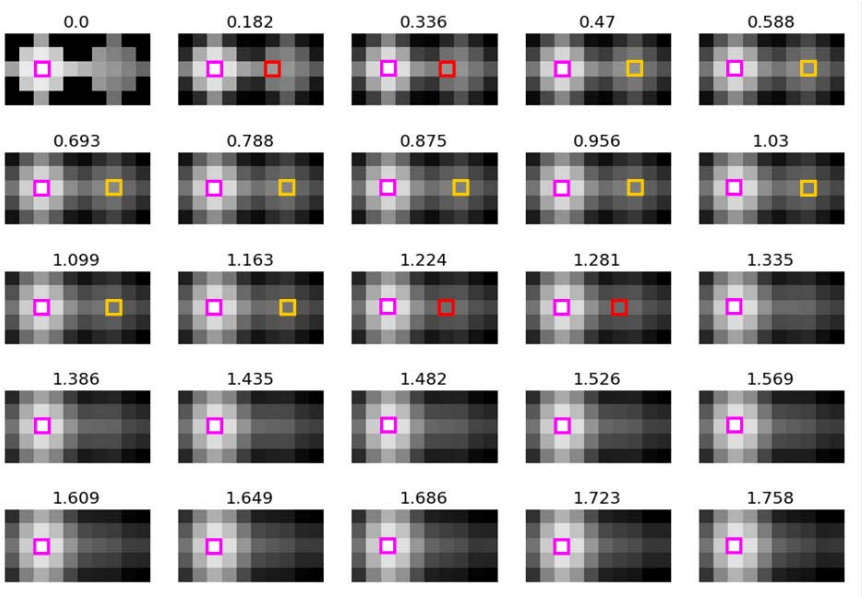


Fig. 3. Sampled scale space over a 5×10 image f. The number above each image denotes the scale t. The colored boxes denote positions of maxima found using the N_6 neighborhood. While the purple maximum is stationary over scale, the red/orange maximum changes its position twice before it disappears.

3.2 Topological Changes over Scale

In order to extract critical curves, the discrete scale space has to be sampled along its continuous scale parameter, since we do not know how to compute the exact occurrences of the zerocrossings in the neighborhood as stated above. The known way to

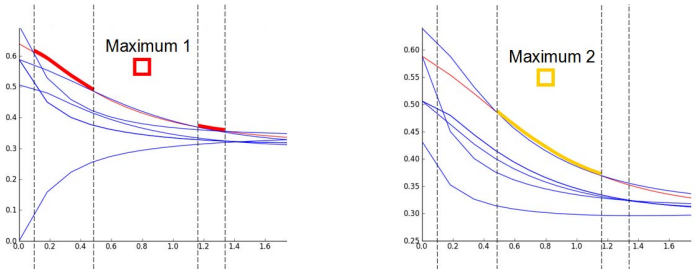


Fig. 4. Tracing maxima over scale. Left: The value of the red maximum at $L\,(3, 7; \cdot)$ and its neighbors (blue) plotted over scale. Right: The values of the orange maximum at $L\,(3, 8; \cdot)$ and its neighbors (blue) plotted over scale.

extract critical curves is to compute several scale images of L at preselected scales, then detecting spatially critical points on these scales and finally linking spatially close critical points on subsequent scales into critical curves. The critical curves extraction algorithm implemented in ScaleSpaceViz computes the first spatial derivatives of the sampled scales and then makes use of the marching cubes algorithm to extract zero crossing surfaces in L. Their intersections form critical paths. These intersections may be inaccurate due to undersampling [8]. Increasing the sampling density might increase the accuracy, however we can never guarantee that, even for very high sampling densities, the resulting critical curves are correct - see Fig. 5.

Therefore we need a criterion that tells us when further subsampling is required and when the sampling density is high enough to guarantee a correct result. Such a criterion might be found in topological graphs or more precisely in the difference of topological graphs of subsequent scales. Tracking changes in the surface network over scale is a promising approach, since the set of possible changes between scale space events such as creations or annihilations of critical points is strictly limited.

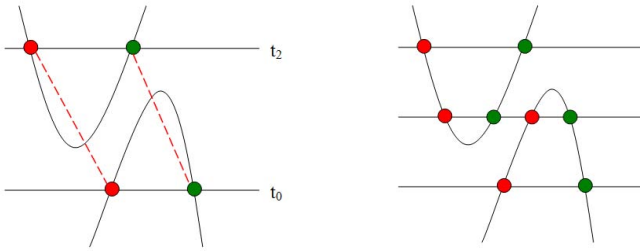


Fig. 5. Left: Undersampling and subsequent annihilation and creation events lead to inaccuracies. Right: Finer sampling increases the accuracy. However, there is no lower bound on the sampling density that guarantees a correct linking of critical curves.

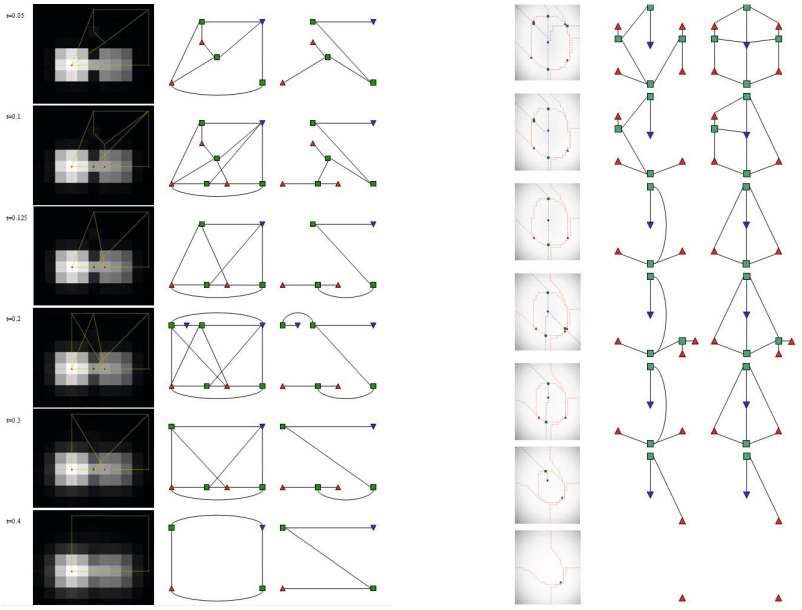


Fig. 6. Left: A sampled discrete scale space. From the critical points alone (left), the subsequent annihilation and creation between scale $t_1 = 0.005$ and $t_2 = 0.125$ is not visible and might lead to incorrect linking. However, the surface networks (middle, incomplete) of these scales differ, thus providing a criterion whether further subsampling might be necessary. The Reeb graphs (right, incomplete) are identical. Right: Topological changes over scale manifest themselves in changes of the Reeb graph (middle) and the surface network (right).

A multi-graph surface network that allows for multiple edges between related vertices depicts changes in subsequent scales even more accurate [12]. The number of edges is then proportional to the number of saddle points. Annihilations and creations of extremum saddle pairs reduce or increase the number of vertices and edges in a predictable manner (Fig. 6).

4 Conclusion and Future Work

The discrete scale space as an equivalent to the two-dimensional Gaussian scale space has been discussed and some important properties have been derived. A computationally practicable implementation of the discrete scale space framework has been outlined. A computationally efficient sampling method, based on properties of the discrete finite difference Laplacian kernel, has been proposed and compared to competing approaches. A first investigation of the deep structure of the discrete scale space has illustrated the need for a more robust algorithm for critical curve extraction. Topological graphs have show promising properties under the influence of changes in scale. However, further and more formal investigation of the deep structure of the discrete scale space is necessary. Also, the positions of critical points detected on the homogeneous

6-neighborhood are not invariant under certain transformations such as rotations and thus formal differential geometrical analysis of generic scale space singularities where co-dimension 1 is filled in with the scale derivative. Depending on the choices made during the triangulation of the square lattice, positions of saddle points can change up to one pixel [17].

References

1. Koenderink, J.J.: The structure of images. Biological Cybernetics 50, 363–370 (1984)
2. Kuijper, A., Florack, L.M.J.: Understanding and modeling the evolution of critical points under Gaussian blurring. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 143–157. Springer, Heidelberg (2002)
3. Kuijper, A., Florack, L.: The relevance of non-generic events in scale space models. International Journal of Computer Vision 1(57), 67–84 (2004)
4. Kanters, F., Lillholm, M., Duits, R., Janssen, B., Platel, B., Florack, L., ter Haar Romeny, B.: On image reconstruction from multiscale top points. In: Kimmel, R., Sochen, N.A., Weickert, J. (eds.) Scale-Space 2005. LNCS, vol. 3459, pp. 431–442. Springer, Heidelberg (2005)
5. Kanters, F., Florack, L., Duits, R., Platel, B., ter Haar Romeny, B.: Scalespaceviz: a-scale spaces in practice. Pattern Recognition and Image Analysis 17, 106–116 (2007)
6. Felsberg, M., Duits, R., Florack, L.: The monogenic scale space on a bounded domain and its applications. In: Griffin, L.D., Lillholm, M. (eds.) Scale-Space 2003. LNCS, vol. 2695, pp. 209–224. Springer, Heidelberg (2003)
7. Janssen, B., Duits, R., ter Haar Romeny, B.M.: Linear image reconstruction by sobolev norms on the bounded domain. In: Sgallari, F., Murli, A., Paragios, N. (eds.) SSVM 2007. LNCS, vol. 4485, pp. 55–67. Springer, Heidelberg (2007)
8. Kanters, F., Florack, L.: Deep structure, singularities, and computer vision. Technical report, Eindhoven University of Technology (September 2003)
9. Lindeberg, T.: Scale-Space Theory in Computer Vision. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers (1994)
10. Tschirsich, M., Kuijper, A.: Laplacian eigenimages in discrete scale space. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR&SPR 2012. LNCS, vol. 7626, pp. 162–170. Springer, Heidelberg (2012)
11. Kuijper, A.: On detecting all saddle points in 2d images. Pattern Recognition Letters 25(15), 1665–1672 (2004)
12. Tschirsich, M., Kuijper, A.: A discrete scale space neighborhood for robust deep structure extraction. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR&SPR 2012. LNCS, vol. 7626, pp. 126–134. Springer, Heidelberg (2012)
13. Raz, R.: On the complexity of matrix product. In: ACM Symposium on Theory of Computing, pp. 144–151 (2002)
14. Lizhi, C., Zengrong, J.: An efficient algorithm for cyclic convolution based on fast-polynomial and fast-w transforms. Circuits, Systems, and Signal Processing 20, 77–88 (2001)
15. Nowak, R.: 2D DFT (July 2005), `http://cnx.org/content/m10987/2.4/`
16. Kenyon, R.: The Laplacian on planar graphs and graphs on surfaces. ArXiv e-prints (March 2012)
17. Scott, P.J.: An algorithm to extract critical points from lattice height data. International Journal of Machine Tools and Manufacture 41(13-14), 1889–1897 (2001)

Image Matching Using Generalized Scale-Space Interest Points

Tony Lindeberg*

School of Computer Science and Communication
KTH Royal Institute of Technology, Stockholm, Sweden

Abstract. The performance of matching and object recognition methods based on interest points depends on both the properties of the underlying interest points and the associated image descriptors. This paper demonstrates the advantages of using generalized scale-space interest point detectors when computing image descriptors for image-based matching. These generalized scale-space interest points are based on linking of image features over scale and scale selection by weighted averaging along feature trajectories over scale and allow for a higher ratio of correct matches and a lower ratio of false matches compared to previously known interest point detectors within the same class. Specifically, it is shown how a significant increase in matching performance can be obtained in relation to the underlying interest point detectors in the SIFT and the SURF operators. We propose that these generalized scale-space interest points when accompanied by associated scale-invariant image descriptors should allow for better performance of interest point based methods for image-based matching, object recognition and related vision tasks.

Keywords: interest points, scale selection, scale linking, matching, object recognition, feature detection, scale invariance, scale space.

1 Introduction

A common approach to image-based matching consists of detecting interest points with associated image descriptors from image data and then establishing a correspondence between the image descriptors. Specifically, the SIFT operator [1] and the SURF operator [2] have been demonstrated to be highly useful for this purpose with many successful applications, including object recognition, 3-D object and scene modelling, video tracking, gesture recognition, panorama stitching as well as robot localization and mapping.

In the SIFT operator, the intitial detection of interest points is based on differences-of-Gaussians from which local extrema over space and scale are computed. Such points are referred to as scale-space extrema. The difference of Gaussian operator can be seen as an approximation of the Laplacian operator,

* The support from the Swedish Research Council (contract 2010-4766), the Royal Swedish Academy of Sciences and the Knut and Alice Wallenberg Foundation is gratefully acknowledged.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 355–367, 2013.

and it follows from general results in [3] that the scale-space extrema of the Laplacian have scale-invariant properties that can be used for normalizing local image patches or image descriptors with respect to scaling transformations. The SURF operator is on the other hand based on initial detection of image features that can be seen as approximations of the determinant of the Hessian operator with the underlying Gaussian derivatives replaced by an approximation in terms of Haar wavelets. From the general results in [3] it follows that scale-space extrema of the determinant of the Hessian do also lead to scale-invariant behaviour, which can be used for explaining the good performance of the SIFT and SURF operators under scaling transformations.

The subject of this article is to show how the performance of image matching can be improved by using a generalized framework for detecting interest points from scale-space features involving (i) new Hessian feature strength measures at a fixed scale, (ii) linking of image features over scale into feature trajectories to allow for a better selection of significant image features and (iii) scale selection by weighted averaging along feature trajectories to allow for more robust scale estimates. By replacing the interest points in the regular SIFT and SURF operators by generalized scale-space interest points to be described below, it is possible to define new scale-invariant image descriptors that lead to better matching performance compared to the performance obtained by corresponding interest point detection mechanisms as used in the SIFT and SURF operators.

2 Generalized Scale-Space Interest Points

Basic requirements on the interest points on which image matching is to be performed are that they should (i) have a clear, preferably mathematically well-founded, *definition*, (ii) have a well-defined *position* in image space, (iii) have local image structures around the interest point that are *rich in information content* such that the interest points carry important information to later stages and (iv) be stable under local and global deformations of the image domain, including perspective image deformations and illumination variations such that the interest points can be reliably computed with a high degree of *repeatability*.

2.1 Differential Entities for Detecting Scale-Space Interest Points

As basis for performing local image measurements on a two-dimensional image f, we will consider a *scale-space representation* [4–10]

$$L(x, y;\ t) = \int_{(u,v)\in \mathbb{R}^2} f(x-u, y-v)\, g(u, v;\ t)\, du\, dv \tag{1}$$

generated by convolution with Gaussian kernels $g(x, y;\ t) = \frac{1}{2\pi t}\, e^{-(x^2+y^2)/2t}$ of increasing width, where the variance t is referred to as the scale parameter, and with *scale-normalized derivatives* with $\gamma = 1$ defined according to a $\partial_\xi = t^{\gamma/2}\, \partial_x$ and $\partial_\eta = t^{\gamma/2}\, \partial_y$ [3]. To detect interest points within this scale-space framework, we will consider:

(i) either the *scale-normalized Laplacian operator* [3]

$$\nabla^2_{norm} L = t\,(L_{xx} + L_{yy}) \tag{2}$$

or the *scale-normalized determinant of the Hessian* [3]

$$\det \mathcal{H}_{norm} L = t^2\,(L_{xx} L_{yy} - L^2_{xy}), \tag{3}$$

(ii) either of the following differential analogues/extensions of the Harris operator [11] proposed in [12, 13]; the *unsigned Hessian feature strength measure I*

$$\mathcal{D}_{1,norm} L = \begin{cases} t^2\,(\det \mathcal{H}L - k\ \mathrm{trace}^2\,\mathcal{H}L) & \text{if } \det \mathcal{H}L - k\ \mathrm{trace}^2\,\mathcal{H}L > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

or the *signed Hessian feature strength measure I*

$$\tilde{\mathcal{D}}_{1,norm} L = \begin{cases} t^2\,(\det \mathcal{H}L - k\ \mathrm{trace}^2\,\mathcal{H}L) & \text{if } \det \mathcal{H}L - k\ \mathrm{trace}^2\,\mathcal{H}L > 0 \\ t^2\,(\det \mathcal{H}L + k\ \mathrm{trace}^2\,\mathcal{H}L) & \text{if } \det \mathcal{H}L + k\ \mathrm{trace}^2\,\mathcal{H}L < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $k \in [0, \frac{1}{4}[$ as derived in [12] with the preferred choice $k \approx 0.06$, or

(iii) either of the following differential analogues and extensions of the Shi and Tomasi operator [14] proposed in [12, 13]; the *unsigned Hessian feature strength measure II*

$$\mathcal{D}_{2,norm} L = t\ \min(|\lambda_1|, |\lambda_2|) = t\ \min(|L_{pp}|, |L_{qq}|) \tag{6}$$

or the *signed Hessian feature strength measure II*

$$\tilde{\mathcal{D}}_{2,norm} L = \begin{cases} t\,L_{pp} & \text{if } |L_{pp}| < |L_{qq}| \\ t\,L_{qq} & \text{if } |L_{qq}| < |L_{pp}| \\ t\,(L_{pp} + L_{qq})/2 & \text{otherwise} \end{cases} \tag{7}$$

with L_{pp} and L_{qq} denoting the eigenvalues of the Hessian matrix ordered such that $L_{pp} \leq L_{qq}$ [10].

2.2 Scale Selection Mechanisms

To perform scale selection for the abovementioned differential feature detectors, we will consider two different approaches:

- Detection of *scale-space extrema* $(\hat{x}, \hat{y}, \hat{t})$ where the scale normalized differential entities assume local extrema with respect to space and scale [3], and with image features ranked by the magnitude of the scale-normalized response $|\mathcal{D}_{norm} L|$ at the scale-space extremum.
- Linking image features at different scales into feature trajectories over scale and performing scale selection by weighted averaging of scale values along each feature trajectory T delimited by bifurcation events [12, 13]

$$\hat{\tau}_T = \frac{\int_{\tau \in T} \tau\, \psi((\mathcal{D}_{\gamma-norm} L)(p(\tau);\ \tau))\, d\tau}{\int_{\tau \in T} \psi((\mathcal{D}_{\gamma-norm} L)(p(\tau);\ \tau))\, d\tau} \tag{8}$$

with the integral expressed in terms of effective scale $\tau = \log t$ to give a scale covariant construction of the corresponding scale estimates $\hat{t}_T = \exp \hat{\tau}_T$, and with significance measure taken as the integral of the scale-normalized feature responses along the feature trajectory [12, 13]

$$W_T = \int_{\tau \in T} \psi(|(\mathcal{D}_{norm} L)(p(\tau);\ \tau)|)\, d\tau \tag{9}$$

where $\psi(|\mathcal{D}_{norm} L|) = w_{\mathcal{D}L}\, |\mathcal{D}_{norm} L|^a$ represents a monotonically increasing self-similar transformation and $w_{\mathcal{D}L} = (L_{\xi\xi}^2 + 2L_{\xi\eta}^2 + L_{\eta\eta}^2)/(A(L_\xi^2 + L_\eta^2) + L_{\xi\xi}^2 + 2L_{\xi\eta}^2 + L_{\eta\eta}^2 + \varepsilon^2)$ with $A = 4/e$ representing the relative weighting between first- and second-order derivatives [15] and with $\varepsilon \approx 0.1$ representing an estimated noise level for image data in the range $[0, 256]$.

In [13] it is shown that when applied to a rotationally symmetric Gaussian blob model $f(x, y) = g(x, y;\ t_0)$, both scale-space extrema detection and weighed scale selection lead to similar scale estimates $\hat{t} = t_0$ for all the above interest point detectors. When, subjected to non-uniform affine image deformations outside the similarity group, the determinant of the Hessian $\det \mathcal{H}_{norm} L$ and the Hessian feature strength measures $\mathcal{D}_{1,norm} L$ and $\tilde{\mathcal{D}}_{1,norm} L$ do, however, have theoretical advantages in terms of affine covariance or approximations thereof [12, 13].

3 Scale-Invariant Image Descriptors for Matching

For each interest point, we will compute a complementary image descriptor in analogous ways as done in the SIFT and SURF operators, with the difference that the feature vectors will be computed from Gaussian derivative responses in a scale-space representation instead of using a pyramid as done in the original SIFT operator [1] or a Haar wavelet basis as used in the SURF operator [2].

For our SIFT-like image descriptor, we compute image gradients ∇L at the detection scale $\hat{t}$ of the interest point. An orientation estimate is computed in a similar way as by Lowe [1], by accumulating a histogram of gradient directions $\arg \nabla L$ quantized into 36 bins with the area of the accumulation window proportional to the detection scale $\hat{t}$, and then detecting peaks in the smoothed orientation histograms. Multiple peaks are accepted if the height of the secondary peak(s) are above 80 % of the highest peak. Then, for each point on a 4×4 grid with the grid spacing proportional to the detection scale measured in units of $\hat{\sigma} = \sqrt{\hat{t}}$, a weighed local histogram of gradient directions $\arg \nabla L$ quantized into 8 bins is accumulated around each grid point, with the weights proportional to the gradient magnitude $|\nabla L|$ and a Gaussian window function with its area proportional to the detection scale $\hat{t}$ with trilinear interpolation for distributing the weighted increments for the sampled image measurements into adjacent histogram bins. The resulting 128-dimensional descriptor is normalized to unit sum to achieve contrast invariance, with the relative contibution of a single bin limited to a maximum value of 0.20.

For our SURF-like image descriptor, we compute the following sums of derivative responses $\sum L_x$, $\sum |L_x|$, $\sum L_y$, $\sum |L_y|$ at the scale $\hat{t}$ of the interest point, for each one of 4×4 subwindows around the interest point as Bay et al [2] and with similar orientation normalization as for the SIFT operator. The resulting 64-D descriptor is then normalized to unit length for contrast invariance.

4 Matching Properties under Perspective Transformations

To evaluate the quality of the interest points with their associated local image descriptors, we will apply bi-directional nearest-neighbour matching of the image descriptors in Euclidean norm. In other words, given a pair of images f_A and f_B with corresponding sets of interest points $A = \{A_i\}$ and $B = \{B_j\}$, a match between the pair of interest points (A_i, B_j) is accepted only if (i) A_i is the best match for B_j in relation to all the other points in A and, in addition, (ii) B_j is the best match for A_i in relation to all the other points in B.

To suppress matching candidates for which the correspondence may be regarded as ambiguous, we will furthermore require the ratio between the distances to the nearest and the next nearest image descriptor to be less than $r = 0.9$.

Next, we will evaluate the matching performance of such interest points with local image descriptors over a dataset of poster images with calibrated homographies over different amounts of perspective scaling and foreshortening.

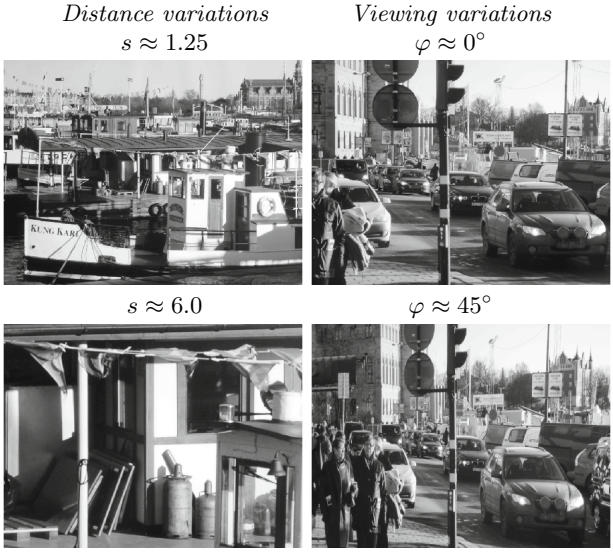


Fig. 1. Illustration of images of posters from multiple views (left) by varying the distance between the camera and the object for different frontal views, and (right) by varying the viewing direction relative to the direction of the surface normal. (Image size: 768×576 pixels.).

4.1 Poster Image Dataset

High-resolution photographs of approximately 4900×3200 pixels were taken of 12 outdoor and indoor scenes in natural city and office environments, from which poster printouts of size 100×70 cm were produced by a professional laboratory. Each such poster was then photographed from 14 different positions:

(i) 11 normal views leading to approximate scaling transformations with relative scale factors s approximately equal to 1.25, 1.5, 1.75, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0 and 6.0, and

(ii) 3 additional oblique views leading to foreshortening transformations with slant angles 22.5°, 30° and 45° relative to the frontal view with $s \approx 2.0$.

For the 11 normal views of each objects, homographies were computed between each pair using the ESM method [16] with initial estimates of the relative scaling factors obtained from manual measurements of the distance between the poster surface and the camera. For the oblique views, for which the ESM method did not produce sufficiently accurate results, homographies were computed by first manually marking correspondences between the four images of each poster, computing an initial estimate of the homography using the linear method in [17, algorithm 3.2, page 92] and then computing a refined estimate by minimizing the Sampson approximation of the geometric error [17, algorithm 3.3, page 98].

The motivation for using such poster image for evaluation is to reflect natural image structures while allowing for easy calibration without 3-D reconstruction.

4.2 Matching Criteria and Performance Measures

Figure 2 shows an illustration of point matches obtained between two pairs of images corresponding to a scaling transformation and a foreshortening transformation based on interest points detected using the $\tilde{\mathcal{D}}_{1,norm}L$ operator.

To judge whether two image features A_i and B_j matched in this way should be regarded as belonging to the same feature or not, we associate a scale dependent circle C_A and C_B to each feature, with the radius of each circle equal to the detection scale of the corresponding feature measured in units of the standard deviation $\sigma = \sqrt{t}$. Then, each such feature is transformed to the other image domain, using the homography and with the scale value transformed by a scale factor of the homography. The relative amount of overlap between any pair of circles is defined by forming the ratio between the intersection and the union of the two circles in a similar way as Mikolajczyk et al [18] define a corresponding ratio for ellipses

$$m(C_A, C_B) = \frac{|\bigcap(C_A, C_B)|}{|\bigcup(C_A, C_B)|}. \tag{10}$$

Then, we measure the performance of the interest point detector by:

$$\text{efficiency} = \frac{\#(\text{interest points that lead to accepted matches})}{\#(\text{interest points})}$$

$$1 \text{ - precision} = \frac{\#(\text{rejected matches})}{\#(\text{accepted matches}) + \#(\text{rejected matches})}$$

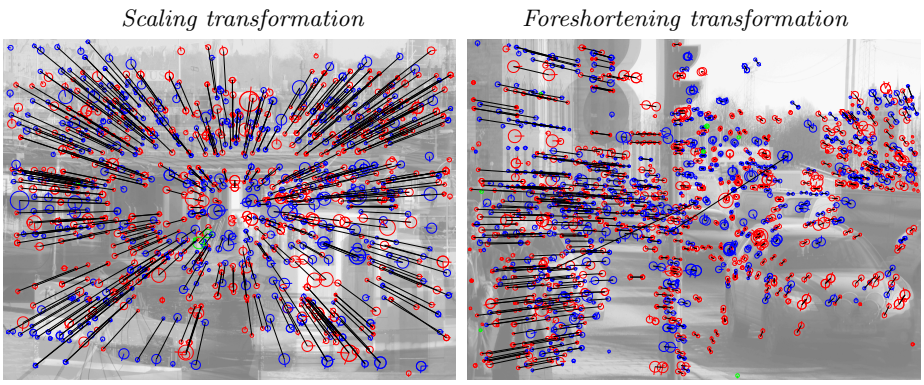


Fig. 2. Illustration of matching relations obtained by bidirectional matching of SIFT-like image descriptors computed at interest points of the signed Hessian feature strength measure $\widetilde{\mathcal{D}}_{1,norm}L$ for (left) a scaling transformation and (right) a foreshortening transformation between pairs of poster images of the harbour and city scenes shown in Figure 1. These illustrations have been generated by first superimposing bright copies of the two images to be matched by adding them. Then, the interest points detected in the two domains have been overlayed on the image data, and a black line has been drawn between each pair of image points that has been matched. Red circles indicate that the Hessian matrix is negative definite (bright features), blue circles that the Hessian matrix is positive definite (dark features), whereas green circles indicate that the Hessian matrix is indefinite (saddle-like features).

The evaluation of the matching score is only performed for image features that are within the image domain for both images before and after the transformation. Moreover, only features within corresponding scale ranges are evaluated. In other words, if the scale range for the image f_A is $[t_{min}, t_{max}]$, then image features are searched for in the transformed image f_B within the scale range $[t'_{min}, t'_{max}] = [s^2\, t_{min}, s^2\, t_{max}]$, where s denotes an overall scaling factor of the homography. In the experiments below, we used $[t_{min}, t_{max}] = [4, 256]$.

4.3 Experimental Results

Table 1 shows the result of evaluating 2×9 different types of scale-space interest point detectors with respect to the problem of establishing point correspondences between pairs of images on the poster dataset. Each interest point detector is applied in two versions (i) with scale selection from local extrema of scale-normalized derivatives over scale, or (ii) using scale linking with scale selection from weighted averaging of scale-normalized feature responses along feature trajectories.

In addition to the 2×7 differential interest point detectors described in section 2, we have also included 2×2 additional interest point detectors derived from

Table 1. Performance measures obtained by matching different types of scale-space interest points with associated SIFT- and SURF-like image descriptors for the poster image dataset. The columns show from left to right: (i) the average efficiency over all pairs of scaling transformations, (ii) the average efficiency over all pairs of foreshortening transformations and (iii) the average total computed as the mean of the scaling and foreshortening scores. The columns labelled "extr" and "link" indicate whether the features have been detected with scale selection from extrema over scale or by scale linking.

Efficiency: SIFT-like image descriptor

	scaling		foreshortening		average	
Interest points	extr	link	extr	link	extr	link
$\nabla^2_{norm} L \quad (\mathcal{D}_1 L > 0)$	0.7484	0.7994	0.7512	0.7574	0.7498	0.7784
$\det \mathcal{H}_{norm} L \ (\mathcal{D}_1 L > 0)$	0.7721	*0.8225*	0.7635	*0.7932*	0.7678	*0.8079*
$\det \mathcal{H}_{norm} L \ (\tilde{\mathcal{D}}_1 L > 0)$	0.7691	0.8163	0.7602	0.7841	0.7647	0.8002
$\mathcal{D}_{1,norm} L$	0.7719	**0.8280**	0.7596	**0.7977**	0.7658	**0.8128**
$\tilde{\mathcal{D}}_{1,norm} L$	0.7698	0.8241	0.7578	*0.7916*	0.7638	*0.8079*
$\mathcal{D}_{2,norm} L \quad (\mathcal{D}_1 L > 0)$	0.7203	0.8187	0.7111	0.7776	0.7157	0.7981
$\tilde{\mathcal{D}}_{2,norm} L \quad (\mathcal{D}_1 L > 0)$	0.7204	*0.8261*	0.7113	0.7766	0.7159	0.8014
Harris-Laplace	0.7002	0.7855	0.7046	0.7535	0.7024	0.7695
Harris-detHessian	0.7406	0.7608	0.7561	0.7319	0.7406	0.7463

Efficiency: SURF-like image descriptor

	scaling		foreshortening		average	
Interest points	extr	link	extr	link	extr	link
$\nabla^2_{norm} L \quad (\mathcal{D}_1 L > 0)$	0.7424	0.7832	0.7280	0.7140	0.7352	0.7486
$\det \mathcal{H}_{norm} L \ (\mathcal{D}_1 L > 0)$	0.7656	0.8072	0.7402	*0.7504*	0.7529	*0.7788*
$\det \mathcal{H}_{norm} L \ (\tilde{\mathcal{D}}_1 L > 0)$	0.7628	0.8015	0.7372	0.7430	0.7500	0.7723
$\mathcal{D}_{1,norm} L$	0.7661	**0.8126**	0.7354	**0.7537**	0.7507	**0.7831**
$\tilde{\mathcal{D}}_{1,norm} L$	0.7640	*0.8081*	0.7334	*0.7478*	0.7487	*0.7779*
$\mathcal{D}_{2,norm} L \quad (\mathcal{D}_1 L > 0)$	0.7157	0.8014	0.6870	0.7284	0.7013	0.7649
$\tilde{\mathcal{D}}_{2,norm} L \quad (\mathcal{D}_1 L > 0)$	0.7158	*0.8100*	0.6873	0.7328	0.7015	0.7714
Harris-Laplace	0.6948	0.7620	0.6724	0.6944	0.6836	0.7282
Harris-detHessian	0.7345	0.7381	0.7192	0.6705	0.7268	0.7043

the Harris operator [11]: (i) the Harris-Laplace operator [19] based on spatial extrema of the Harris measure and scale selection from local extrema over scale of the scale-normalized Laplacian, (ii) a scale-linked version of the Harris-Laplace operator with scale selection by weighted averaging over feature trajectories of Harris features [12], and (iii-iv) two Harris-detHessian operators analogous to the Harris-Laplace operators, with the difference that scale selection is performed based on the scale-normalized determinant of the Hessian instead of the scale-normalized Laplacian [12].

Table 2. The five best combinations of interest points and image descriptors among the $2 \times 2 \times 9 = 36$ combinations considered in this experimental evaluation as ranked on the ratio of interest points that lead to correct matches. For comparison, results are also shown for the SIFT descriptor based on scale-space extrema of the Laplacian, the SIFT or SURF descriptors based on scale-space extrema of the determinant of the Hessian and the SIFT descriptor based on Harris-Laplace interest points.

Interest points and image descriptors ranked on matching efficiency

Interest points	Scale selection	Descriptor	Efficiency
$\mathcal{D}_{1,norm}L$	link	SIFT	0.8128
$\tilde{\mathcal{D}}_{1,norm}L$	link	SIFT	0.8079
$\det \mathcal{H}_{norm}L$ $(\mathcal{D}_1 L > 0)$	link	SIFT	0.8079
$\tilde{\mathcal{D}}_{2,norm}L$ $(\mathcal{D}_1 L > 0)$	link	SIFT	0.8014
$\det \mathcal{H}_{norm}L$ $(\tilde{\mathcal{D}}_1 L > 0)$	link	SIFT	0.8002
$\vdots$			$\vdots$
$\det \mathcal{H}_{norm}L$ $(\mathcal{D}_1 L > 0)$	extr	SIFT	0.7721
$\det \mathcal{H}_{norm}L$ $(\mathcal{D}_1 L > 0)$	extr	SURF	0.7656
$\nabla^2_{norm}L$ $(\mathcal{D}_1 L > 0)$	extr	SIFT	0.7484
Harris-Laplace	extr	SIFT	0.7002

The experiments are based on detecting the $N = 800$ strongest interest points extracted from the first image, regarded as reference image for the homography. To obtain an approximate uniform density of interest points under scaling transformations, an adapted number $N' = N/s^2$ of interest points is searched for (i) within the subwindow of the reference image that is mapped to the interior of the transformed image and (ii) in the transformed image, with s denoting relative scaling factor between the two images.

This procedure is repeated for all pairs of images within the groups of distance variations or viewing variations respectively, implying up to 55 image pairs for the scaling transformations and 6 image pairs for the foreshortening transformations, *i.e.* up to 61 matching experiments for each one of the 12 posters, thus up to 732 experiments for each one of 2×9 interest point detectors.

As can be seen from the results of matching SIFT- or SURF-like image descriptors in Table 1, the interest point detectors based on scale linking and with scale selection by weighted averaging along feature trajectories generally lead to significantly higher efficiency rates compared to the corresponding interest point detectors based on scale selection from local extrema over scale. Specifically, the highest efficiency rates are obtained with the scale linked version of the unsigned Hessian feature strength measure $\mathcal{D}_{1,norm}L$, followed by scale-linked versions of the unsigned signed Hessian feature strength measure $\tilde{\mathcal{D}}_{1,norm}L$ and the determinant of the Hessian operator $\det \mathcal{H}_{norm}L$ with complementary thresholding on $\mathcal{D}_{1,norm}L > 0$.

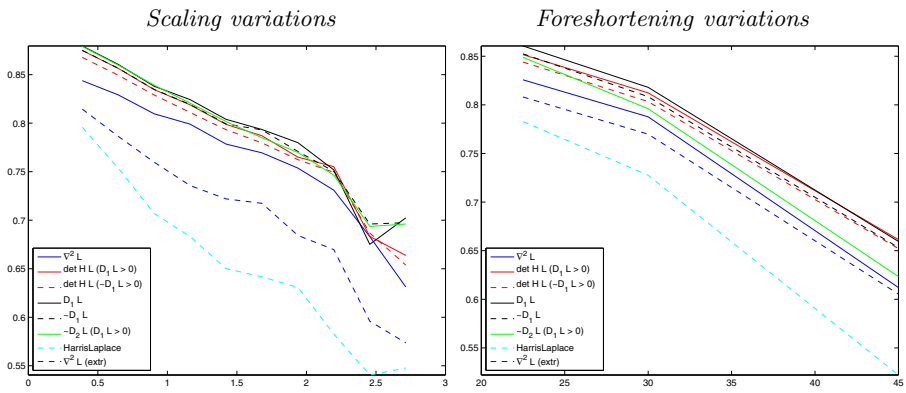


Fig. 3. Graphs showing how the matching efficiency depends upon (left) the amount of scaling $s \in [1.25, 6.0]$ for scaling transformations (with $\log_2 s$ on the horizontal axis) and (right) the difference in viewing angle $\varphi \in [22.5°, 45°]$ for the foreshortening transformations for interest point matching based on SIFT-like image descriptors.

Corresponding experimental results that cannot be included here because of lack of space show that the lowest and thus the best 1-precision score is obtained with the determinant of the Hessian operator $\det \mathcal{H}_{norm} L$ with complementary thresholding on $\tilde{\mathcal{D}}_{1,norm} L > 0$, followed by the determinant of the Hessian operator $\det \mathcal{H}_{norm} L$ with complementary thresholding on $\mathcal{D}_{1,norm} L > 0$.

Among the more traditional feature detectors based on scale selection from local extrema over scale, we can also note that the determinant of the Hessian operator $\det \mathcal{H}_{norm} L$ performs significantly better than both the Laplacian operator $\nabla^2_{norm} L$ and the Harris-Laplace operator. We can also note that the Harris-Laplace operator can be improved by either scale linking or by replacing scale selection based on the scale-normalized Laplacian by scale selection based on the scale-normalized determinant of the Hessian.

When comparing the results obtained for SIFT-like and SURF-like image descriptors, we can see that the SIFT-like image descriptors lead to both higher efficiency rates and lower 1-precision scores than the SURF-like image descriptors. This qualitative relationship holds over all types of interest point detectors. In this respect, the pure image descriptor in the SIFT operator is clearly better than the pure image descriptor in the SURF operator. Specifically, more reliable image matches can be obtained by replacing pure image descriptor in the SURF operator by the pure image descriptor in the SIFT operator.

Table 2 lists the five best combinations of interest point detectors and image descriptors in this evaluation as ranked on their efficiency values. For comparison, the results of our corresponding analogues of the SIFT operator with interest point detection from scale-space extrema of the Laplacian and our analogue of the SURF operator based on scale-space extrema of the determinant of the Hessian are also shown. As can be seen from this ranking, the best combinations

of generalized points with SIFT-like image descriptors perform significantly better than the corresponding analogues of regular SIFT or regular SIFT based on scale-space extrema of the Laplacian or the determinant of the Hessian.

Figure 3 shows graphs of how the efficiency rate depends upon the amount of scaling for the scaling transformations and the difference in viewing angle for the foreshortening transformations. As can be seen from these graphs, the interest point detectors $\det \mathcal{H}_{norm} L$, $\mathcal{D}_{1,norm} L$ and $\tilde{\mathcal{D}}_{1,norm} L$ that possess affine covariance properties or approximations thereof [12, 13] do also have the best matching properties under foreshortening transformations. Specifically, the generalized interest point detectors based on scale linking perform significantly better than scale-space extrema of the Laplacian or the determinant of the Hessian as well as better than the Harris-Laplace operator.

5 Summary and Conclusions

We have presented a set of extensions of the SIFT and SURF operators, by replacing the underlying interest point detectors used for computing the SIFT or SURF descriptors by a family of generalized scale-space interest points.

These generalized scale-space interest points are based on (i) new differential entities for interest point detection at a fixed scale in terms of new Hessian feature strength measures, (ii) linking of image structures into feature trajectories over scale and (ii) performing scale selection by weighted averaging of scale-normalized feature responses along these feature trajectories [12].

The generalized scale-space interest points are all *scale-invariant* in the sense that (i) the interest points are preserved under scaling transformation and that (ii) the detection scales obtained from the scale selection step are transformed in a scale covariant way. Thereby, the detection scale can be used for defining a local scale normalized reference frame around the interest point, which means that image descriptors that are defined relative to such a scale-normalized reference frame will also be scale invariant.

By complementing these generalized scale-space interest points with local image descriptors defined in a conceptually similar way as the pure image descriptor parts in SIFT or SURF, while being based on image measurements in terms of Gaussian derivatives instead of image pyramids or Haar wavelets, we have shown that the generalized interest points with their associated scale-invariant image descriptors lead to a higher ratio of correct matches and a lower ratio of false matches compared to corresponding results obtained with interest point detectors based on more traditional scale-space extrema of the Laplacian, scale-space extrema of the determinant of the Hessian or the Harris-Laplace operator.

In the literature, there has been some debate concerning which one of the SIFT or SURF descriptors leads to the best performance. In our experimental evaluations, we have throughtout found that our SIFT-like image descriptor based on Gaussian derivatives generally performs much better than our SURF-like image descriptor, also expressed in terms of Gaussian derivatives. In this respect, the pure image descriptor in the SIFT operator can be seen as significantly better than the pure image descriptor in the SURF operator.

Concerning the underlying interest points, we have on the other hand found that the determinant of the Hessian operator to generally perform significantly better than the Laplacian operator, both for scale selection based on scale-space extrema and scale selection based on weighted averaging of feature responses along feature trajectories obtained by scale linking. Since the difference-of-Gaussians interest point detector in the regular SIFT operator can be seen as an approximation of the scale-normalized Laplacian, we can therefore regard the underlying interest point detector in the SURF operator as significantly better than the interest point detector in the SIFT operator. Specifically, we could expect a significant increase in the performance of SIFT by just replacing the scale-space extrema of the difference-of-Gaussians operator by scale-space extrema of the determinant of the Hessian.

In addition, our experimental evaluations show that further improvements are possible by replacing the interest points obtained from scale-space extrema in the SIFT and SURF operators by generalized scale-space interest points obtained by scale linking, with the best results obtained with the Hessian feature strength measures $\mathcal{D}_{1,norm}L$ and $\tilde{\mathcal{D}}_{1,norm}L$ followed by the determinant of the Hessian $\det \mathcal{H}_{norm}L$ and the Hessian feature strength measure $\tilde{\mathcal{D}}_{2,norm}L$.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comp. Vis. 60, 91–110 (2004)
2. Bay, H., Ess, A., Tuytelaars, T.: van Gool: Speeded up robust features (SURF). CVIU 110, 346–359 (2008)
3. Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comp. Vis. 30, 77–116 (1998)
4. Witkin, A.P.: Scale-space filtering. In: 8th IJCAI, pp. 1019–1022 (1983)
5. Koenderink, J.J.: The structure of images. Biol. Cyb. 50, 363–370 (1984)
6. Koenderink, J.J., van Doorn, A.J.: Generic neighborhood operators. IEEE-PAMI 14, 597–605 (1992)
7. Lindeberg, T.: Scale-Space Theory in Computer Vision. Springer (1994)
8. Florack, L.M.J.: Image Structure. Springer (1997)
9. ter Haar Romeny, B.: Front-End Vision and Multi-Scale Image Analysis. Springer (2003)
10. Lindeberg, T.: Scale-space. In: Wah, B. (ed.) Encyclopedia of Computer Science and Engineering, pp. 2495–2504. Wiley (2008)
11. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, pp. 147–152 (1988)
12. Lindeberg, T.: Generalized scale-space interest points: Scale-space primal sketch for differential descriptors (2010) (under revision for International Journal of Computer Vision, original version submitted in June 2010)
13. Lindeberg, T.: Scale selection properties of generalized scale-space interest point detectors. J. Math. Im. Vis (September 2012), doi:10.1007/s10851-012-0378-3
14. Shi, J., Tomasi, C.: Good features to track. In: CVPR, pp. 593–600 (1994)
15. Lindeberg, T.: On automatic selection of temporal scales in time-casual scale-space. In: Sommer, G. (ed.) AFPAC 1997. LNCS, vol. 1315, pp. 94–113. Springer, Heidelberg (1997)

16. Benhimane, S., Malis, E.: Real-time image-based tracking of planes using efficient second-order minimization. In: Intelligent Robots and Systems, pp. 943–948 (2004)
17. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 1st edn. Cambridge University Press (2000)
18. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., van Gool, L.: A comparison of affine region detectors. Int. J. Comp. Vis. 65, 43–72 (2005)
19. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comp. Vis. 60, 63–86 (2004)

A Fully Discrete Theory for Linear Osmosis Filtering

Oliver Vogel, Kai Hagenburg, Joachim Weickert, and Simon Setzer

Mathematical Image Analysis Group
Dept. of Mathematics and Computer Science, Campus E1.7
Saarland University, 66041 Saarbrücken, Germany
{vogel,hagenburg,weickert,setzer}@mia.uni-saarland.de
http://www.mia.uni-saarland.de

Abstract. Osmosis filters are based on drift–diffusion processes. They offer nontrivial steady states with a number of interesting applications. In this paper we present a fully discrete theory for linear osmosis filtering that follows the structure of Weickert's discrete framework for diffusion filters. It regards the positive initial image as a vector and expresses its evolution in terms of iterative matrix–vector multiplications. The matrix differs from its diffusion counterpart by the fact that it is unsymmetric. We assume that it satisfies four properties: vanishing column sums, nonnegativity, irreducibility, and positive diagonal elements. Then the resulting filter class preserves the average grey value and the positivity of the solution. Using the Perron–Frobenius theory we prove that the process converges to the unique eigenvector of the iteration matrix that is positive and has the same average grey value as the initial image. We show that our theory is directly applicable to explicit and implicit finite difference discretisations. We establish a stability condition for the explicit scheme, and we prove that the implicit scheme is absolutely stable. Both schemes converge to a steady state that solves the discrete elliptic equation. This steady state can be reached efficiently when the implicit scheme is equipped with a BiCGStab solver.

Keywords: osmosis filtering, drift–diffusion, finite difference methods, BiCGStab.

1 Introduction

Osmosis filtering relies on the idea of making diffusion filters unsymmetric. This is achieved by supplementing it with a drift term that allows nontrivial steady states. While specific applications of this idea to the fields of digital halftoning and numerical methods for hyperbolic conservation laws can be found in two earlier publications [1,2], the first comprehensive description of osmosis models for a variety of visual computing applications is presented in our companion paper [3]. In [3] we demonstrate that osmosis models are powerful tools for compact data representation, for editing an existing image, and for fusing information from different images. Most of these applications go far beyond of what can

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 368–379, 2013.

be achieved with nonlinear diffusion filters, in spite of the fact that the osmosis models in [3] are linear. Osmosis filters have some similarities to gradient domain methods from computer graphics [4,5], but offer additional advantages such as invariance under multiplicative illumination changes.

Since osmosis can be interpreted as a modification of diffusion filtering and there is a well-established theory for diffusion filters, it is natural to study which results can be generalised from diffusion to osmosis. The goal of the present paper is to provide a fully discrete theory for linear osmosis filtering that has a similar structure as Weickert's discrete framework for diffusion filters [6]. We will see that this theory offers some fundamental differences to diffusion filters, and that it is applicable to the design of osmosis algorithms that are not only reliable, but also efficient.

Our paper is organised as follows. In Section 2 we review the basic structure of continuous osmosis filters, and we consider finite difference discretisations in space and time. This leads us to fully discrete osmosis filters that can be expressed as iterative matrix–vector multiplications. Section 3 provides our theoretical framework for this filter class, in which we establish useful properties such as preservation of positivity and convergence results. In Section 4 we apply this theory to two popular finite difference discretisations: an explicit and an implicit scheme. The performance of these schemes is evaluated in Section 5, and a summary in Section 6 concludes our paper.

2 From Continuous to Discrete Osmosis

Before we can introduce a theory for discrete linear osmosis processes in visual computing, we have to discuss the continuous concept first and show how it can be turned into a discrete filter representation. This is the topic of the present section.

2.1 Continuous Linear Osmosis Filtering

Let us consider a rectangular image domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$, and a positive greyscale image $f : \Omega \to \mathbb{R}_+$. Moreover, assume we are given some *drift vector field* $\boldsymbol{d} : \Omega \mapsto \mathbb{R}^2$. Then a (linear) *osmosis filter* computes a processed version $u(\boldsymbol{x}, t)$ of $f(\boldsymbol{x})$ by solving the drift-diffusion PDE

$$\partial_t u \;=\; \Delta u \;-\; \operatorname{div}(\boldsymbol{d}u) \qquad \text{on } \Omega \times (0, T], \tag{1}$$

with f as initial condition,

$$u(\boldsymbol{x}, 0) \;=\; f(\boldsymbol{x}) \qquad \text{on } \Omega, \tag{2}$$

and homogeneous Neumann boundary conditions. They specify a vanishing flux in normal direction $\boldsymbol{n}$ to the image boundary $\partial\Omega$:

$$\langle \boldsymbol{\nabla} u - \boldsymbol{d}u,\, \boldsymbol{n} \rangle \;=\; 0 \qquad \text{on } \partial\Omega \times (0, T]. \tag{3}$$

Let us now sketch three key properties of our osmosis model [3]:

(a) **Preservation of the Average Grey Value:**
Since the osmosis process is in divergence form, its solution preserves the average grey value of the initial image:

$$\frac{1}{|\Omega|}\int_{\Omega} u(\boldsymbol{x},t)\,d\boldsymbol{x} \;=\; \frac{1}{|\Omega|}\int_{\Omega} f(\boldsymbol{x})\,d\boldsymbol{x} \qquad \forall\, t>0\,. \tag{4}$$

This property can also be found for diffusion filters.

(b) **Preservation of Positivity:**
One can show that the solution remains positive for all times:

$$u(\boldsymbol{x},t)>0 \qquad \forall \boldsymbol{x}\in\Omega, \quad \forall t>0. \tag{5}$$

This is a weaker property than the maximum–minimum principle for diffusion [6]. Osmosis may violate a maximum–minimum principle.

(c) **Convergence to a Nontrivial Steady State:**
The continuous linear osmosis model differs from a homogeneous diffusion filter only by its drift term. However, the drift vector field $\boldsymbol{d}$ is a powerful tool to steer its convergence: If $\boldsymbol{d}$ satisfies

$$\boldsymbol{d} \;=\; \boldsymbol{\nabla}(\ln v) \;=\; \frac{\boldsymbol{\nabla} v}{v} \tag{6}$$

with some positive image v, one can show that the osmosis process converges to v up to a multiplicative constant which ensures preservation of the average grey value of f. Thus, osmosis creates nontrivial steady states. This is a fundamental difference to diffusion that allows only flat steady states [6].

Since $\boldsymbol{d}$ contains the gradient information of $\ln v$, we may regard osmosis as a process for data integration. In that sense it resembles so-called gradient domain methods that are popular in computer graphics [4,5]. Therefore, it is not surprising that it can also be used for similar applications such as image editing and image fusion. We refer to our companion paper [3] for such applications. Other applications are concerned with alternative numerical schemes for hyperbolic conservation laws [2]. Moreover, also the PDE limit of a lattice Boltzmann model for halftoning [1] is an osmosis equation.

Applying osmosis to colour images is as simple as applying it to greyscale images: One proceeds separately in each RGB channel using the individual drift vector fields of each channel.

2.2 Finite Difference Discretisation

Let us now consider a finite difference space discretisation of the drift–diffusion equation (1). We consider a grid size h in x- and y-direction, and we denote by $u_{i,j}$ an approximation to u in the grid point $((i-\frac{1}{2})h,(j-\frac{1}{2})h))^\top$. Setting $\boldsymbol{d}=(d_1,d_2)^\top$, we approximate (1) by

$$u'_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2} - \frac{1}{h}\left(d_{1,i+\frac{1}{2},j}\frac{u_{i+1,j}+u_{i,j}}{2} - d_{1,i-\frac{1}{2},j}\frac{u_{i,j}+u_{i-1,j}}{2}\right) - \frac{1}{h}\left(d_{2,i,j+\frac{1}{2}}\frac{u_{i,j+1}+u_{i,j}}{2} - d_{2,i,j-\frac{1}{2}}\frac{u_{i,j}+u_{i,j-1}}{2}\right) \quad (7)$$

This also holds for boundary points, if we mirror the image at its boundaries and assume a zero drift vector across boundaries. Rearranging (7) gives

$$\begin{aligned} u'_{i,j} = {} & u_{i+1,j}\left(\frac{1}{h^2} - \frac{d_{1,i+\frac{1}{2},j}}{2h}\right) + u_{i-1,j}\left(\frac{1}{h^2} + \frac{d_{1,i-\frac{1}{2},j}}{2h}\right) \\ & + u_{i,j+1}\left(\frac{1}{h^2} - \frac{d_{2,i,j+\frac{1}{2}}}{2h}\right) + u_{i,j-1}\left(\frac{1}{h^2} + \frac{d_{2,i,j-\frac{1}{2}}}{2h}\right) \\ & + u_{i,j}\left(-\frac{4}{h^2} - \frac{d_{1,i+\frac{1}{2},j}}{2h} + \frac{d_{1,i-\frac{1}{2},j}}{2h} - \frac{d_{2,i,j+\frac{1}{2}}}{2h} + \frac{d_{2,i,j-\frac{1}{2}}}{2h}\right). \end{aligned} \quad (8)$$

From now on we restrict ourselves to drift vector fields $(d_1(\boldsymbol{x}), d_2(\boldsymbol{x}))^\top$ with

$$|d_1(\boldsymbol{x})| < \frac{2}{h}, \qquad |d_2(\boldsymbol{x})| < \frac{2}{h} \qquad \forall\, \boldsymbol{x} \in \Omega. \quad (9)$$

This ensures that in (8) the weights of all four neighbours of $u_{i,j}$ are positive. We want to write this discretisation in a more compact notation. To this end, we replace the double indexing in each pixel by a single index and assemble all unknown grey values in a single vector $\boldsymbol{u} \in \mathbb{R}^N$ where N denotes the number of pixels. Then we end up with the following dynamical system:

$$\boldsymbol{u}(0) = \boldsymbol{f}, \quad (10)$$

$$\boldsymbol{u}'(t) = \boldsymbol{A}\,\boldsymbol{u}(t) \quad (11)$$

where the matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is unsymmetric. This differs from the diffusion scenario that leads to symmetric matrices [6]. Since the weights of the neighbours in (8) are positive, it follows that $\boldsymbol{A}$ has nonnegative off-diagonals. Moreover, one can show that all column sums of $\boldsymbol{A}$ are zero and $\boldsymbol{A}$ is irreducible.

We have different options to discretise this ODE system in time. In the simplest case one can consider the *explicit scheme*:

$$\frac{\boldsymbol{u}^{k+1} - \boldsymbol{u}^k}{\tau} = \boldsymbol{A}\boldsymbol{u}^k \quad (12)$$

where $\tau > 0$ denotes the time step size, and the upper index k refers to an approximation at time $k\tau$. With $\boldsymbol{P} := \boldsymbol{I} + \tau\boldsymbol{A}$, we can rearrange this scheme to

$$\boldsymbol{u}^{k+1} = \boldsymbol{P}\boldsymbol{u}^k \quad (13)$$

An alternative time discretisation is given by the *implicit scheme*

$$\frac{\boldsymbol{u}^{k+1} - \boldsymbol{u}^k}{\tau} = \boldsymbol{A}\boldsymbol{u}^{k+1}. \quad (14)$$

It requires to solve a linear system in the unknown vector $\boldsymbol{u}^{k+1}$. If the system matrix is invertible, the problem can also be formally written as a matrix–vector multiplication of type (13) with $\boldsymbol{P} := (\boldsymbol{I} - \tau \boldsymbol{A})^{-1}$.

3 A Discrete Osmosis Theory

We have seen that both the explicit and the implicit scheme are examples of numerical methods that can be written in the general form (13). This motivates us to derive a general theory for discrete osmosis processes of this type. Here is our main result.

Proposition 1. [Theory for Discrete Linear Osmosis]
Let $\boldsymbol{f} \in \mathbb{R}_+^N$ and consider a process

$$\boldsymbol{u}^0 = \boldsymbol{f}, \tag{15}$$

$$\boldsymbol{u}^{k+1} = \boldsymbol{P}\boldsymbol{u}^k \qquad (k = 0, 1, ...) \tag{16}$$

where the (unsymmetric) matrix $\boldsymbol{P} \in \mathbb{R}^{N \times N}$ satisfies the following properties:

(DLO1) All column sums of $\boldsymbol{P}$ are 1.
(DLO2) $\boldsymbol{P}$ is nonnegative.
(DLO3) $\boldsymbol{P}$ is irreducible.
(DLO4) $\boldsymbol{P}$ has only positive diagonal entries.

Then the following results hold:

(a) The average grey value is preserved:

$$\frac{1}{N}\sum_{i=1}^{N} u_i^k \;=\; \frac{1}{N}\sum_{i=1}^{N} f_i \qquad \forall\, k > 0\,. \tag{17}$$

(b) The evolution preserves positivity:

$$u_i^k > 0 \qquad \forall i \in \{1, .., N\}, \quad \forall\, k > 0\,. \tag{18}$$

(c) There exists a unique steady state for $k \to \infty$. It is given by the eigenvector $\boldsymbol{v} \in \mathbb{R}_+^N$ of $\boldsymbol{P}$ to the eigenvalue 1, that has the same average grey value as $\boldsymbol{f}$.

Proof. Average grey value invariance and preservation of positivity are very easily seen, while the convergence result requires some more technicalities.

(a) Average grey value invariance for osmosis has already been shown in [2], where the reasoning is identical to the diffusion case [6, Proposition 4]:
(b) In order to verify preservation of positivity, we observe that applying one osmosis step to the positive initial image $\boldsymbol{f}$ gives

$$u_i^1 \;=\; \underbrace{p_{i,i}}_{>0}\,\underbrace{f_i}_{>0} \;+\; \sum_{\substack{j=1\\ j\neq i}}^{N} \underbrace{p_{i,j}}_{\geq 0}\,\underbrace{f_j}_{>0} \;>\; 0 \qquad \forall\, i \in \{1, ..., N\}\,. \tag{19}$$

Applying this reasoning iteratively ensures that $\boldsymbol{u}^k$ is positive for all $k > 0$.

(c) To establish our convergence result, first we show that 1 is an eigenvalue of $\boldsymbol{P}$. As the eigenvalues of $\boldsymbol{P}$ and $\boldsymbol{P}^\top$ are identical, we can exploit the unit row sum of $\boldsymbol{P}^\top$ instead of the unit column sum of $\boldsymbol{P}$. Hence, we can compute

$$\boldsymbol{P}^\top \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^N p_{1,j} \\ \sum_{j=1}^N p_{2,j} \\ \vdots \\ \sum_{j=1}^N p_{N,j} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{20}$$

Thus, 1 is an eigenvalue of $\boldsymbol{P}^\top$ and therefore also of $\boldsymbol{P}$. Note that $(1, 1, \ldots, 1)^\top$ is an eigenvector for $\boldsymbol{P}^\top$, but not for $\boldsymbol{P}$.

Next we prove that all eigenvalues $\lambda \in \mathbb{C}$ with $\lambda \neq 1$ satisfy $|\lambda| < 1$. Since $\boldsymbol{P}$ has unit column sums, its column sum norm satisfies $\|\boldsymbol{P}\|_1 = 1$. Thus, we have $|\lambda| \leq 1$. By Gershgorin's theorem, all eigenvalues of $\boldsymbol{P}$ lie within disks in the complex domain whose centres are given by the diagonal entries, respectively. As the spectrum of eigenvalues of a matrix is the same as the spectrum of eigenvalues of a transposed matrix, we can compute the set of all Gershgorin disks as

$$\Lambda := \bigcup_{j=1}^N \underbrace{\left\{ z \in \mathbb{C} \,\middle|\, |z - p_{j,j}| \leq \sum_{i=1, i\neq j}^N |p_{i,j}| \right\}}_{=:B_j}. \tag{21}$$

Since $\boldsymbol{P}$ is nonnegative with unit column sums and positive diagonal elements, we conclude that

$$\sum_{i=1, i\neq j}^N |p_{i,j}| \;=\; \sum_{i=1, i\neq j}^N p_{i,j} \;=\; 1 - p_{j,j} \;<\; 1. \tag{22}$$

As it holds for all j that $B_j \cap \{z \in \mathbb{C} \mid |z| = 1\} = \{1\}$, we can describe Λ as

$$\Lambda \subset \{z \in \mathbb{C} \mid |z| < 1\} \cup \{1\}. \tag{23}$$

By the assumptions $\lambda \in \Lambda$ and $\lambda \neq 1$, we have $|\lambda| < 1$.

For the final step of our convergence analysis, we need the following results from the Perron-Frobenius theory (see e.g. Theorem 8.4.4 in [7]):
If $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is irreducible and nonnegative, then its spectral radius $\rho(\boldsymbol{A})$ is a simple eigenvalue of $\boldsymbol{A}$. Moreover, there exists a positive eigenvector to $\rho(\boldsymbol{A})$.
Since $\rho(\boldsymbol{P}) = 1$, this theorem states that $\lambda = 1$ is a simple eigenvalue and has a positive eigenvector. Hence, the iteration (15)–(16) attenuates all components outside the eigenspace of $\lambda = 1$ to zero. Therefore, the process converges to a vector $\boldsymbol{v}$ in the eigenspace of $\lambda = 1$. Since $\boldsymbol{f} \in \mathbb{R}^N_+$ and the iteration preserves the positive average grey value, it converges to a vector $\boldsymbol{v} \in \mathbb{R}^N$ with the same positive average grey value as $\boldsymbol{f}$. Because of the cited Perron-Frobenius result we know that $\boldsymbol{v}$ is positive. □

Our framework for discrete linear osmosis allows to analyse osmosis algorithms in a very simple way: All one has to do is to check the four properties (DLO1)–(DLO4). If they are satisfied, we can be sure that the filter preserves the average grey value and the positivity of the original image, and we have full control over its steady state.

It should be mentioned that this theory is very general: It does not rely on any specific space discretisation on a regular grid. Without any alterations, it is applicable to osmosis processes acting on graphs, on surface data, or on higher dimensional data sets.

4 Application to Finite Difference Discretisations

Let us now apply our discrete osmosis theory to two important finite difference discretisations that we have already mentioned: the explicit and the implicit scheme. We will see that they are not only useful for computing the parabolic time evolution, but also for the elliptic steady state.

4.1 The Parabolic Time Evolution

Applying Proposition 1 to the explicit and the implicit scheme gives the following result.

Proposition 2. [Finite Difference Discretisations]
Let $\boldsymbol{f} \in \mathbb{R}^N_+$ and consider the semidiscrete linear osmosis evolution

$$\boldsymbol{u}(0) = \boldsymbol{f}, \tag{24}$$
$$\boldsymbol{u}'(t) = \boldsymbol{A}\,\boldsymbol{u}(t) \tag{25}$$

where the (unsymmetric) matrix $\boldsymbol{A} = (a_{i,j}) \in \mathbb{R}^{N\times N}$ fulfils the following properties:

(SLO1) All column sums of $\boldsymbol{A}$ are 0.
(SLO2) $\boldsymbol{A}$ has only nonnegative off-diagonal entries.
(SLO3) $\boldsymbol{A}$ is irreducible.

Then the following results hold:

(a) The explicit scheme

$$\boldsymbol{u}^{k+1} \;=\; (\boldsymbol{I} + \tau \boldsymbol{A})\,\boldsymbol{u}^k \tag{26}$$

satisfies the requirements (DLO1)–(DLO4) for discrete linear osmosis processes provided that

$$\tau < \frac{1}{|a_{i,i}|} \qquad \forall\, i \in \{1, ..., N\}. \tag{27}$$

(b) The implicit scheme

$$(\boldsymbol{I} - \tau \boldsymbol{A})\, \boldsymbol{u}^{k+1} \;=\; \boldsymbol{u}^{k} \tag{28}$$

satisfies (DLO1)–(DLO4) for all time step sizes $\tau > 0$.

Proof. We check (DLO1)–(DLO4) by applying classical matrix analysis.

(a) It holds that $a_{i,i} \neq 0$ because otherwise (SLO1) implies that the whole column i of $\boldsymbol{A}$ is 0, and thus the digraph associated with $\boldsymbol{A}$ is not strongly connected. This contradicts the irreducibility of $\boldsymbol{A}$. The unit column sum property (DLO1) follows directly from the zero column sums of $\boldsymbol{A}$. Moreover, $\boldsymbol{I}+\tau\boldsymbol{A}$ is nonnegative (DLO2) with positive diagonal elements (DLO4), since (SLO2) holds true and τ fulfils (27). Clearly, (27) guarantees that $\boldsymbol{I} + \tau\boldsymbol{A}$ and $\boldsymbol{A}$ have the same digraph. Thus, $\boldsymbol{I} + \tau\boldsymbol{A}$ is also irreducible (DLO3).

(b) We start by observing that $\boldsymbol{I} - \tau\boldsymbol{A}$ is strictly column diagonally dominant: From the zero column sum property (SLO1) it follows that

$$-a_{j,j} = \sum_{i=1, i\neq j}^{N} a_{i,j} \qquad \forall\, j \in \{1, \ldots, N\} \tag{29}$$

and thus

$$1 - \tau a_{j,j} > \tau \sum_{i=1, i\neq j}^{N} a_{i,j} \qquad \forall\, \tau > 0. \tag{30}$$

By (SLO2) the off-diagonals of $\boldsymbol{A}$ are nonnegative. Hence, we can apply Gershgorin's theorem to the columns of $\boldsymbol{I}-\tau\boldsymbol{A}$ and conclude that this matrix is nonsingular. Let us consider the row vector $\boldsymbol{e} := (1, 1, \ldots, 1)$ with N components. Clearly, (SLO1) means that $\boldsymbol{I}-\tau\boldsymbol{A}$ has unit column sums. The same holds true for its inverse since

$$\boldsymbol{e}\,(\boldsymbol{I} - \tau\boldsymbol{A}) \;=\; \boldsymbol{e} \quad\Longleftrightarrow\quad \boldsymbol{e} \;=\; \boldsymbol{e}\,(\boldsymbol{I} - \tau\boldsymbol{A})^{-1}. \tag{31}$$

This proves (DLO1). The nonpositivity of the off-diagonals of $\boldsymbol{I} - \tau\boldsymbol{A}$ and its strict column diagonal dominance imply that $\boldsymbol{I} - \tau\boldsymbol{A}$ is a nonsingular M-matrix, cf. [8, Theorem 6.2.3 (C_{10})]. For any nonsingular M-matrix, it holds that its inverse has only strictly positive entries, see [8, Theorem 6.2.7]. This shows (DLO2)–(DLO4). □

Proposition 2 gives stability results with respect to preservation of positivity. Since also the average grey value is preserved, it follows that $0 < u_j^k < \sum_i f_i$ for all $j \in \{1, ..., N\}$ and for all $k > 0$. This ensures that also the ℓ_p norms of the solution remain bounded for $p > 1$. Note that osmosis does not allow to give stability results in terms of decreasing ℓ_p norms for $p > 1$, since comparable properties for the L_p norms do not hold for the continuous equation: An osmosis process that starts with a flat image and converges to a nonflat one with identical average grey value may serve as counterexample. This shows that preservation of positivity is a very natural stability criterion for osmosis.

For a spatial grid size of $h = 1$, the condition (9) becomes

$$|d_1(\boldsymbol{x})| < 2, \qquad |d_2(\boldsymbol{x})| < 2 \qquad \forall\, \boldsymbol{x} \in \Omega, \tag{32}$$

and inspecting the central weight in (8) shows that $|a_{i,i}| < 8$. Thus, the stability condition (27) for the explicit scheme becomes $\tau < \frac{1}{8}$. This stability bound is half as large as the well-known stability limit of an explicit scheme for the homogeneous 2-D diffusion equation $\partial_t u = \Delta u$.

The absolute stability of the implicit scheme is in full accordance with the corresponding diffusion result from [6, Theorem 8]. The implicit scheme yields nonsymmetric pentadiagonal systems of linear equations that are strictly diagonally dominant in their columns. Using the classical theory of regular splittings [9], one can show that the Gauß-Seidel algorithm converges under these circumstances. More efficient alternatives include Krylov subspace methods such as the BiCGStab method [10] and its preconditioned variants [11]. Implementing these iterative methods is fairly straightforward. Also multigrid methods [12] appear promising, but are more cumbersome to implement.

4.2 The Elliptic Steady State

For many applications of osmosis – such as the ones discussed in [3] – one is mainly interested in the osmotic steady state. Thus, it appears tempting to approximate the elliptic PDE

$$\Delta u - \operatorname{div}(\boldsymbol{d} u) \;=\; 0 \tag{33}$$

and its homogeneous Neumann boundary conditions directly with numerical solvers. However, this can become unpleasant since the elliptic problem has infinitely many solutions: For any solution $w(\mathbf{x})$, also $c\,w(\mathbf{x})$ with some arbitrary constant c is a solution.

This suggests to use also our parabolic time evolution schemes to obtain the desired solution that is positive and has the same average grey value as the initial image f. For the explicit scheme (26) the steady state $\boldsymbol{w}$ is characterised by

$$\boldsymbol{w} \;=\; (\boldsymbol{I} + \tau \boldsymbol{A})\, \boldsymbol{w} \tag{34}$$

and for the implicit scheme (28), it satisfies

$$(\boldsymbol{I} - \tau \boldsymbol{A})\, \boldsymbol{w} \;=\; \boldsymbol{w}\,. \tag{35}$$

Interestingly both equations (34) and (35) are equivalent to

$$\boldsymbol{A}\, \boldsymbol{w} \;=\; \boldsymbol{0} \tag{36}$$

which is a space discretisation of the elliptic PDE (33). Thus, we have the remarkable situation that any stable time step size τ gives the correct elliptic steady state $\boldsymbol{w}$. This makes the implicit scheme with large τ attractive for this task, if one has an efficient solver for the resulting linear systems of equations.

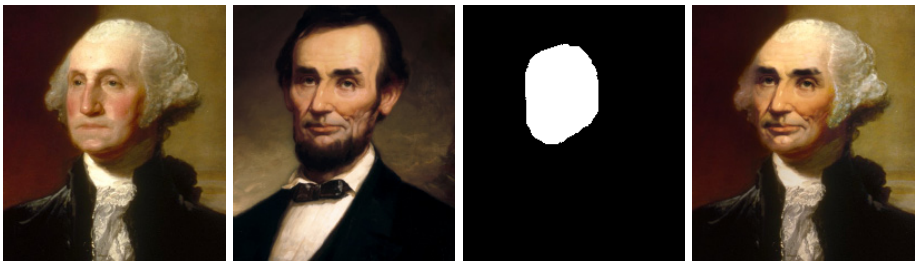


Fig. 1. Seamless image cloning with osmosis. From left to right: **(a)** Painting of George Washington by Gilbert Stuart (Source: Wikimedia Commons, public domain work). **(b)** Painting of Abraham Lincoln by George Story (Source: Wikimedia Commons, public domain work). **(c)** Mask for the seamless image cloning. **(d)** Osmotic steady state using combined drift vector fields.

5 Experimental Evaluation

The preceding discrete osmosis framework provides general criteria that guarantee the *reliability* of osmosis schemes. However, it tells us nothing about their *speed*. To evaluate the practical performance of the explicit and the implicit scheme, let us now consider a typical image editing problem where one is interested in the osmotic steady state.

For our experiment we want to combine the two images from Figure 1(a) and (b). They depict contemporary paintings of famous US presidents. The task is to replace the face of George Washington with the face of Abraham Lincoln in a seamless way. The image of Washington serves as initialisation of our osmosis process. In order to apply osmosis, we first have to specify its drift vectors. We choose the drift vectors of the Washington image where the binary mask image of Fig. 1(c) is black, and the drift vectors of the Lincoln image where the mask image is white. At the interface we perform arithmetic averaging of both drift vector fields. With this combined drift vector field we compute the osmosis evolution. Its steady state gives the seamlessly cloned image in Fig. 1(d).

Now let us discuss some numerical details. For a positive image f, we use the following discretisation of its canonical drift vector field $(d_1, d_2)^\top = \frac{\nabla f}{f}$ in the sense of (6):

$$d_{1,i+\frac{1}{2},j} = \frac{2\,(f_{i+1,j} - f_{i,j})}{h\,(f_{i+1,j} + f_{i,j})}\,, \qquad d_{2,i,j+\frac{1}{2}} = \frac{2\,(f_{i,j+1} - f_{i,j})}{h\,(f_{i,j+1} + f_{i,j})}\,. \tag{37}$$

These vectors are fed into our space discretisation (7), and as time discretisation we use the explicit and the implicit scheme. In the implicit case, we have tested different solvers for the linear system of equations, including Gauß-Seidel, SOR, BiCGStab, and two preconditioned BiCGStab variants. Because BiCGStab without preconditioning offered the best performance, we only report results for this solver here. Since we approach our steady state solution iteratively, we need a stopping criterion: We compute the average ℓ_1 distance per pixel between our

Table 1. CPU times [s] and number of iterations for different image sizes and different osmosis schemes. For the explicit scheme we use $\tau = 0.12$, and in the implicit case $\tau = 10^5$.

image size	explicit:	time[s]	iterations	implicit:	time[s]	iterations
100×115		14.689	61184		0.3179	2
200×230		359.49	240115		4.5454	2
400×460		4487.6	948484		61.909	3

numerical solution and a precomputed ground truth. The iterations are stopped if this error is less than 0.1, where the initial range of each colour channel is $[1, 256]$.

Table 1 shows a comparison of the CPU times for the explicit and the implicit scheme for three different image sizes. The run times are obtained with a double precision C implementation on a standard desktop PC with an Intel Xeon processor, clocked at 3.2 GHz with single threading and without GPU support. We observe that the implicit scheme with BiCGStab allows to reach the desired steady state solution up to 79 times faster than the explicit scheme.

6 Summary and Conclusions

We have introduced a fully discrete theory for osmosis filters that can be expressed in terms of linear drift-diffusion equations. Its prerequisites differ from the ones for discrete diffusion filtering by the fact that the iteration matrix is not symmetric. We have seen that this seemingly small difference has a substantial impact on properties such as maximum-minimum principles and nontrivial steady states. The possibility to design interesting steady states is a key feature of osmosis filtering, and our paper has provided a discrete characterisation of the osmotic steady state. Moreover, we have established stability results in terms of preservation of positivity which is a very natural stability concept for osmosis. We have shown that our theory is applicable to important finite difference approximations such as explicit and implicit schemes. Finally, we have demonstrated that an implicit scheme with a BiCGStab solver also constitutes an efficient method for obtaining the osmotic steady state. This method is not very difficult to implement and can be two orders of magnitude faster than the explicit scheme.

In our ongoing work we are exploring alternative numerical options such as multigrid solvers [12] and additive operator splittings (AOS) [13,14]. Moreover, we are establishing semidiscrete and continuous theories for osmosis filtering that have a similar structure as their diffusion counterparts.

Acknowledgments. This work has been performed while Kai Hagenburg and Oliver Vogel were members of the Mathematical Image Analysis Group. Financial support by the *Deutsche Forschungsgemeinschaft (DFG)* is gratefully acknowledged.

References

1. Hagenburg, K., Breuß, M., Vogel, O., Weickert, J., Welk, M.: A lattice Boltzmann model for rotationally invariant dithering. In: Bebis, G., et al. (eds.) ISVC 2009, Part II. LNCS, vol. 5876, pp. 949–959. Springer, Heidelberg (2009)
2. Hagenburg, K., Breuß, M., Weickert, J., Vogel, O.: Novel schemes for hyperbolic PDEs using osmosis filters from visual computing. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 532–543. Springer, Heidelberg (2012)
3. Weickert, J., Hagenburg, K., Breuß, M., Vogel, O.: Linear osmosis models for visual computing (submitted, 2013)
4. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. In: Proc. SIGGRAPH 2002, San Antonio, TX, July 2002, pp. 249–256 (2002)
5. Pérez, P., Gagnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics 22(3), 313–318 (2003)
6. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner, Stuttgart (1998)
7. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1990)
8. Berman, A., Plemmons, R.J.: Nonnegative Matrices in the Mathematical Sciences. Academic Press, New York (1979)
9. Varga, R.A.: Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs (1962)
10. van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing 13(2), 631–644 (1992)
11. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
12. Briggs, W.L., Henson, V.E., McCormick, S.F.: A Multigrid Tutorial, 2nd edn. SIAM, Philadelphia (2000)
13. Lu, T., Neittaanmäki, P., Tai, X.C.: A parallel splitting up method and its application to Navier–Stokes equations. Applied Mathematics Letters 4(2), 25–29 (1991)
14. Weickert, J., ter Haar Romeny, B.M., Viergever, M.A.: Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Transactions on Image Processing 7(3), 398–410 (1998)

L^2-Stable Nonstandard Finite Differences for Anisotropic Diffusion

Joachim Weickert[1], Martin Welk[2], and Marco Wickert[1]

[1] Mathematical Image Analysis Group, Campus E1.7
Saarland University, 66041 Saarbrücken, Germany
{weickert,wickert}@mia.uni-saarland.de
[2] University for Health Sciences, Medical Informatics and Technology
Eduard-Wallnöfer-Zentrum 1, 6060 Hall/Tyrol, Austria
martin.welk@umit.at

Abstract. Anisotropic diffusion filters with a diffusion tensor are successfully used in many image processing and computer vision applications, ranging from image denoising over compression to optic flow computation. However, finding adequate numerical schemes is difficult: Implementations may suffer from dissipative artifacts, poor approximation of rotation invariance, and they may lack provable stability guarantees. In our paper we propose a general framework for finite difference discretisations of anisotropic diffusion filters on a 3×3 stencil. It is based on a gradient descent of a discrete quadratic energy where the occurring derivatives are replaced by classical as well as the widely unknown nonstandard finite differences in the sense of Mickens. This allows a large class of space discretisations with two free parameters. Combining it with an explicit or semi-implicit time discretisation, we establish a general and easily applicable stability theory in terms of a decreasing Euclidean norm. Our framework comprises as many as seven existing space discretisations from the literature. However, we show that also novel schemes are possible that offer a better performance than existing ones. Our experimental evaluation confirms that the space discretisation can have a very substantial and often underestimated impact on the quality of anisotropic diffusion filters.

Keywords: diffusion filtering, finite difference methods, stability, rotation invariance, dissipativity.

1 Introduction

Anisotropic diffusion filters with a diffusion tensor instead of a scalar-valued diffusivity are flexible tools that permit to steer the diffusion process in a desired direction [1]: This can be very useful for image processing tasks ranging from image denoising and enhancement (see e.g. [1, 2]) to lossy image compression [3]. Anisotropic diffusion terms also appear in computer vision applications, e.g. in the Euler-Lagrange equations of variational methods for optic flow computations [4, 5], for stereo reconstruction [6], and for range image integration [7].

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 380–391, 2013.

An anisotropic diffusion filter in the sense of [1] computes a filtered version $u(\boldsymbol{x}, t)$ of some initial image $f(\boldsymbol{x})$ by solving the diffusion equation

$$u_t = \operatorname{div}(\boldsymbol{D}\,\boldsymbol{\nabla} u) \tag{1}$$

with f as initial condition,

$$u(\boldsymbol{x}, 0) = f(\boldsymbol{x})\,, \tag{2}$$

and homogeneous Neumann boundary conditions. Here the lower index t denotes a time derivative, and the divergence and the nabla operators involve spatial derivatives only. The diffusion tensor $\boldsymbol{D}$ is a positive definite (and thus also symmetric) 2×2 matrix that is space-variant and may even depend on derivatives of the evolving image $u(\boldsymbol{x}, t)$. For our discussions below, one can use positive semidefinite diffusion tensors as well.

A large number of numerical schemes has been proposed for anisotropic diffusion processes, including finite elements [8], finite volume methods [9], and lattice Boltzmann techniques [10]. However, mostly finite difference methods are used [1, 2, 11–14], sometimes realised as wavelet shrinkage [15, 16]. Apart from [13], all finite difference schemes approximate the divergence term on a 3×3 stencil.

Unfortunately, finding good finite difference schemes for anisotropic diffusion filters is much more challenging than for their isotropic counterparts with a scalar-valued diffusivity. While the time discretisation mainly influences the efficiency of the method and does not create specific difficulties, the major problem comes from the space discretisation: If the diffusion process is strongly anisotropic, the corresponding direction has to be approximated with very high accuracy in order to avoid undesired dissipative blurring effects. The approximation quality of the rotationally invariant model can also vary a lot even among schemes with identical order of consistency. Last but not least, it is difficult to establish a stability theory for anisotropic diffusion filters: Weickert [1] presents a discrete theory that analyses stability in terms of a maximum-minimum principle. However, he shows that on a 3×3 stencil, this can only be guaranteed if the spectral condition number of $\boldsymbol{D}$ does not exceed 5.82 (see also [17]). In practice, one is usually interested in using more pronounced anisotropies. In this case, there is no L^∞-stability guarantee for the nonnegativity scheme from [1] and its generalisations by Mrázek and Navara [12]. Thus, it would be desirable to have at least an L^2-stability theory such as for the wavelet-inspired schemes from [15, 16]. However, none of the finite difference discretisations in [1, 2, 11–13] gives L^2-stability results.

Our Contributions. The goal of the present paper is to provide a general framework for L^2-stable discretisations of anisotropic diffusion filters on a 3×3 stencil. It is derived as gradient descent of a discretised energy functional with a positive definite quadratic form. By considering also the widely unknown nonstandard discretisations in the sense of Mickens [18], we end up with a two-parameter family of space discretisations on a 3×3 stencil. Interestingly this

family covers seven existing finite difference discretisations that have been proposed for such a stencil [1, 2, 11, 12, 15, 16]. Moreover, we establish stability results in the Euclidean norm for explicit and semi-implicit time discretisations, providing a theoretical foundation for many of these schemes that has not been available so far. Last but not least we present an experiment that illustrates the large impact that the free parameters can have. With a suitable parameter choice, one can design novel schemes with low dissipativity and an excellent approximation of rotation invariance.

Organisation of the Paper. In Section 2 we derive our general finite difference stencil from a discrete energy. Its theoretical properties are analysed in Section 3. In the fourth section, we evaluate various discretisations that arise as special cases, and we conclude our paper with a summary in Section 5.

2 General Discretisation

Quadratic Energy Model. To discretise the anisotropic diffusion process (1) in time, we will use a sequence $t_0 < t_1 < t_2 < \dots$ of discrete time nodes. Freezing the space-variant diffusion tensor $\boldsymbol{D}$ within each time interval $[t_k, t_{k+1})$, $k \in \mathbb{N}_0$ then creates a sequence of linearised processes. In each interval, the evolution equation is a gradient descent of the quadratic energy

$$E(u) \;=\; \frac{1}{2} \int_\Omega \boldsymbol{\nabla}^\top u \, \boldsymbol{D} \, \boldsymbol{\nabla} u \, dx \, dy \tag{3}$$

with a space-variant but time-invariant positive definite diffusion tensor

$$\boldsymbol{D} \;=\; \begin{pmatrix} a(x,y) & b(x,y) \\ b(x,y) & c(x,y) \end{pmatrix}. \tag{4}$$

For discretisation in space, we adopt for u a regular grid $\{1,\dots,N\} \times \{1,\dots,M\}$ with mesh size h in both x and y direction, i.e. the index (i,j) refers to the location (x_i, y_j) with $x_i = x_0 + i\,h$, $y_j = y_0 + j\,h$. Following the proceeding in [15, 16], we assume that approximations for a, b, and c are available in the locations $(i+\frac{1}{2}, j+\frac{1}{2})$. Provided that also u_x^2, $u_x u_y$, and u_y^2 are approximated in $(i+\frac{1}{2}, j+\frac{1}{2})$, a discrete version of the energy (3) is then given by

$$E(\boldsymbol{u}) \;=\; \frac{1}{2} \sum_{i=0}^{N} \sum_{j=0}^{M} \left(a u_x^2 + 2 b u_x u_y + c u_y^2 \right)_{i+\frac{1}{2}, j+\frac{1}{2}} . \tag{5}$$

Suitable approximations should be *local,* i.e. involve only the four pixels in the cell $\{i, i+1\} \times \{j, j+1\}$. In terms of *accuracy,* we require their consistency to be of second order. At boundary locations (rows $j \in \{0, M\}$, columns $i \in \{0, N\}$), values of a, b, c, and u_x, u_y must satisfy appropriate constraints to be compatible with Neumann boundary conditions.

Derivative Approximations. To derive approximations with these properties, we start by discretising u_x and u_y. To this end, we consider combinations of the forward differences

$$\boxed{\_} := D_x u_{i,j} := \frac{u_{i+1,j} - u_{i,j}}{h}, \tag{6}$$

$$\boxed{\overline{}} := D_x u_{i,j+1} := \frac{u_{i+1,j+1} - u_{i,j+1}}{h}, \tag{7}$$

$$\boxed{|} := D_y u_{i,j} := \frac{u_{i,j+1} - u_{i,j}}{h}, \tag{8}$$

$$\boxed{|} := D_y u_{i+1,j} := \frac{u_{i+1,j+1} - u_{i+1,j}}{h}. \tag{9}$$

Nonstandard Finite Difference Approximations. Since our quadratic energy involves expressions in u_x^2, u_y^2, and $u_x u_y$, let us study approximations of these terms with second order of consistency using the discretisations (6)–(9).

We approximate u_x^2 by affine combinations of the arithmetic mean and the geometric mean of the finite differences in x-direction:

$$\begin{aligned} u_x^2\Big|_{i+\frac{1}{2},j+\frac{1}{2}} \approx\ & (1-\alpha_{i+\frac{1}{2},j+\frac{1}{2}}) \cdot \frac{1}{2}\left(\boxed{\_}\cdot\boxed{\_} + \boxed{\overline{}}\cdot\boxed{\overline{}}\right) \\ &+ \alpha_{i+\frac{1}{2},j+\frac{1}{2}} \cdot \boxed{\_}\cdot\boxed{\overline{}}, \end{aligned} \tag{10}$$

where $\alpha_{i+\frac{1}{2},j+\frac{1}{2}}$ in an arbitrary weight that may be space-variant.

Analogously, u_y^2 is approximated by affine combinations of the arithmetic mean and the geometric mean of the finite differences in y-direction:

$$\begin{aligned} u_y^2\Big|_{i+\frac{1}{2},j+\frac{1}{2}} \approx\ & (1-\alpha_{i+\frac{1}{2},j+\frac{1}{2}}) \cdot \frac{1}{2}\left(\boxed{|}\cdot\boxed{|} + \boxed{|}\cdot\boxed{|}\right) \\ &+ \alpha_{i+\frac{1}{2},j+\frac{1}{2}} \cdot \boxed{|}\cdot\boxed{|}. \end{aligned} \tag{11}$$

To treat u_x^2 and u_y^2 equally, we have chosen the same weight $\alpha_{i+\frac{1}{2},j+\frac{1}{2}}$.

Eventually, $u_x u_y$ involves all four combinations of the two finite differences in x-direction and the two finite differences in y-direction:

$$\begin{aligned} u_x u_y\Big|_{i+\frac{1}{2},j+\frac{1}{2}} \approx\ & \frac{1-\beta_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \cdot \frac{1}{2}\left(\boxed{\_}\cdot\boxed{|} + \boxed{\overline{}}\cdot\boxed{|}\right) \\ &+ \frac{1+\beta_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \cdot \frac{1}{2}\left(\boxed{\_}\cdot\boxed{|} + \boxed{\overline{}}\cdot\boxed{|}\right) \end{aligned} \tag{12}$$

with a space-variant weight $\beta_{i+\frac{1}{2},j+\frac{1}{2}}$.

The approximations (10)–(12) deserve some further discussion. Note that for $\alpha_{i+\frac{1}{2},j+\frac{1}{2}} \neq 0$, the second summand in (10) approximates u_x^2 at the location $(i+\frac{1}{2}, j+\frac{1}{2})$ by multiplying two different approximations for u_x, namely $D_x u_{i,j}$

and $D_x u_{i,j+1}$. This is in accordance with one of Mickens' principles for so-called *nonstandard finite difference* schemes [18]: "Nonlinear terms must, in general, be modelled nonlocally on the computational grid or lattice". Here, the term nonlocal means that both approximations refer to different grid points: $D_x u_{i,j}$ is a central difference approximation in $(i+\frac{1}{2}, j)$, while $D_x u_{i,j+1}$ is centred in $(i+\frac{1}{2}, j+1)$.

In a similar way, one sees that also (11) uses nonstandard finite differences for $\alpha_{i+\frac{1}{2},j+\frac{1}{2}} \neq 0$, and so does (12) for $\beta_{i+\frac{1}{2},j+\frac{1}{2}} \neq 0$. Note that for $\beta_{i+\frac{1}{2},j+\frac{1}{2}} = 0$, approximation (12) is equivalent to

$$u_x u_y \Big|_{i+\frac{1}{2},j+\frac{1}{2}} \approx \frac{1}{2}\left(\boxed{\_} + \boxed{\overline{\ }}\right) \cdot \frac{1}{2}\left(\boxed{|} + \boxed{|}\right) \tag{13}$$

which is a standard approximation, since both factors are centred in $(i+\frac{1}{2}, j+\frac{1}{2})$.

Mickens advocates his principle of nonlocal approximation of nonlinear terms as an ingredient for obtaining qualitatively correct discrete models of continuous equations. The evaluation in Section 4 will show the benefit of this idea.

Gradient Descent. Our space-discrete approximation of the anisotropic diffusion process in every pixel i is finally given by the gradient descent

$$\frac{du_i}{dt} = -\frac{\partial E(\boldsymbol{u})}{\partial u_i} \tag{14}$$

for the discrete energy (5) with the approximations (10)–(12). The right hand side gives the desired discretisation of $\operatorname{div}(\boldsymbol{D}\,\boldsymbol{\nabla} u)$. It can be represented by the weights in a (3×3)-stencil. For inner pixels $1 < i < N$, $1 < j < M$ one obtains after some tedious but straightforward calculations the stencil

$$\frac{1}{2h^2} \cdot \begin{array}{|c|c|c|} \hline \left[(\beta-1)\,b + \alpha\,(a+c)\right]_{i-\frac{1}{2},j+\frac{1}{2}} & \begin{array}{l}\left[(1-\alpha)\,c - \alpha\,a - \beta\,b\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ +\left[(1-\alpha)\,c - \alpha\,a - \beta\,b\right]_{i-\frac{1}{2},j+\frac{1}{2}}\end{array} & \left[(\beta+1)\,b + \alpha\,(a+c)\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ \hline \begin{array}{l}\left[(1-\alpha)\,a - \alpha\,c - \beta\,b\right]_{i-\frac{1}{2},j+\frac{1}{2}} \\ +\left[(1-\alpha)\,a - \alpha\,c - \beta\,b\right]_{i-\frac{1}{2},j-\frac{1}{2}}\end{array} & \begin{array}{l} -\left[(1-\alpha)\,(a+c) - (\beta-1)\,b\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ -\left[(1-\alpha)\,(a+c) - (\beta+1)\,b\right]_{i+\frac{1}{2},j-\frac{1}{2}} \\ -\left[(1-\alpha)\,(a+c) - (\beta+1)\,b\right]_{i-\frac{1}{2},j+\frac{1}{2}} \\ -\left[(1-\alpha)\,(a+c) - (\beta-1)\,b\right]_{i-\frac{1}{2},j-\frac{1}{2}}\end{array} & \begin{array}{l}\left[(1-\alpha)\,a - \alpha\,c - \beta\,b\right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ +\left[(1-\alpha)\,a - \alpha\,c - \beta\,b\right]_{i+\frac{1}{2},j-\frac{1}{2}}\end{array} \\ \hline \left[(\beta+1)\,b + \alpha\,(a+c)\right]_{i-\frac{1}{2},j-\frac{1}{2}} & \begin{array}{l}\left[(1-\alpha)\,c - \alpha\,a - \beta\,b\right]_{i+\frac{1}{2},j-\frac{1}{2}} \\ +\left[(1-\alpha)\,c - \alpha\,a - \beta\,b\right]_{i-\frac{1}{2},j-\frac{1}{2}}\end{array} & \left[(\beta-1)\,b + \alpha\,(a+c)\right]_{i+\frac{1}{2},j-\frac{1}{2}} \\ \hline \end{array} \tag{15}$$

where the y-axis is oriented upwards. This stencil approximates $\operatorname{div}(\boldsymbol{D}\,\boldsymbol{\nabla} u)$ with consistency order 2. In boundary pixels, homogeneous Neumann boundary conditions can be taken into account just by mirroring the first and last rows and columns of u.

Table 1. Seven existing space discretisations as special cases of our general stencil

discretisation	α	β
standard discretisation [11]	0	0
Cottet and El-Ayyadi [2]	0	-1
nonnegativity discretisation [1]	0	$\operatorname{sign}(b)$
Mrázek and Navara II [12]	$\frac{\min(a,c)}{a+c}$	0
Mrázek and Navara III [12]	$\frac{\min(a,c)}{2(a+c)}$	$\frac{1}{2}\operatorname{sign}(b)$
wavelet-inspired scheme I [15]	$\frac{1}{2}$	0
wavelet-inspired scheme II [16]	$[0, \frac{1}{2}]$	0

A General Framework for Existing Schemes. Interestingly our space discretisation subsumes a number of anisotropic diffusion stencils from the literature. Table 1 lists seven representatives with the corresponding weight parameters α, β of our general stencil. In three of the listed schemes the weights are chosen space-variant. All but the last two schemes have originally been stated with the diffusion tensor discretised either at locations (i,j) or $(i+\frac{1}{2},j)$, $(i,j+\frac{1}{2})$. In these cases, full correspondence with our scheme is achieved by a suitable grid resampling with linear interpolation. In Section 4 we will see that our general stencil also contains new parameter settings with favourable performance.

3 Theoretical Properties

In the anisotropic diffusion process (1), the diffusion tensor field $\boldsymbol{D}$ is required to consist of positive definite tensors. As a consequence, the quadratic form within the continuous energy (3) is nonnegative. It is therefore natural to ask whether also the discrete energy (5) retains this property. This will help to determine stability properties of the gradient descent.

3.1 Positive Semidefiniteness of the Discrete Energy

Introducing the notations

$$\boldsymbol{w}_{i+\frac{1}{2},j+\frac{1}{2}} := \left(\boxed{\llcorner}, \boxed{\urcorner}, \boxed{\rfloor}, \boxed{\lfloor}\right)^\top, \tag{16}$$

$$\boldsymbol{H}_{i+\frac{1}{2},j+\frac{1}{2}} := \left(\begin{array}{cc|cc} \frac{1-\alpha}{2}a_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{\alpha}{2}a_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1-\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1+\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} \\ \frac{\alpha}{2}a_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1-\alpha}{2}a_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1+\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1-\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} \\ \hline \frac{1-\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1+\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1-\alpha}{2}c_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{\alpha}{2}c_{i+\frac{1}{2},j+\frac{1}{2}} \\ \frac{1+\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1-\beta}{4}b_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{\alpha}{2}c_{i+\frac{1}{2},j+\frac{1}{2}} & \frac{1-\alpha}{2}c_{i+\frac{1}{2},j+\frac{1}{2}} \end{array}\right), \tag{17}$$

we can rewrite our discrete energy (5) as

$$E(\boldsymbol{u}) \;=\; \frac{1}{2}\sum_{i=0}^{N}\sum_{j=0}^{M} \boldsymbol{w}_{i+\frac{1}{2},j+\frac{1}{2}}^{\top}\, \boldsymbol{H}_{i+\frac{1}{2},j+\frac{1}{2}}\, \boldsymbol{w}_{i+\frac{1}{2},j+\frac{1}{2}}\,. \tag{18}$$

Now we state our main result on the discrete energy functional.

Proposition 1 (Positive Semidefiniteness of $\boldsymbol{H}_{i+\frac{1}{2},j+\frac{1}{2}}$). *The matrix $\boldsymbol{H}_{i+\frac{1}{2},j+\frac{1}{2}}$ is positive semidefinite for any positive definite diffusion tensor $\boldsymbol{D}_{i+\frac{1}{2},j+\frac{1}{2}}$ if and only if $|\beta| \leq 1-2\alpha$.*

Sketch of the proof. We decompose $\mathbb{R}^4$ into the subspaces

$$V := \operatorname{span}\{(1,1,0,0)^{\top},(0,0,1,1)^{\top}\} \quad \text{and} \tag{19}$$

$$V^{\perp} = \operatorname{span}\{(1,-1,0,0)^{\top},(0,0,1,-1)^{\top}\}\,. \tag{20}$$

On V, the matrix $\boldsymbol{H}_{i+\frac{1}{2},j+\frac{1}{2}}$ acts like $\frac{1}{2}\boldsymbol{D}_{i+\frac{1}{2},j+\frac{1}{2}}$: To see this, let $(x,y)^{\top}$ be an eigenvector of $\boldsymbol{D}_{i+\frac{1}{2},j+\frac{1}{2}}$ with eigenvalue λ. Then $(x,x,y,y)^{\top}$ is an eigenvector of $\boldsymbol{H}_{i+\frac{1}{2},j+\frac{1}{2}}$ with eigenvalue $\frac{\lambda}{2}$. This guarantees positive definiteness on V.

On $V^{\perp}$, it is easy to check that the action of $\boldsymbol{H}_{i+\frac{1}{2},j+\frac{1}{2}}$ is given by the matrix

$$\boldsymbol{T} := \frac{1}{2}\begin{pmatrix} (1-2\alpha)a & \beta b \\ \beta b & (1-2\alpha)c \end{pmatrix} \tag{21}$$

with respect to the basis vectors stated above. To ensure nonnegativity of the eigenvalues of $\boldsymbol{T}$ for any positive definite $\boldsymbol{D}$, the inequality $|\beta| \leq 1-2\alpha$ is necessary and sufficient. □

Remark. As a consequence, the largest value of α for which positive semidefiniteness can be established is $\alpha = 0.5$. However, we do not recommend using $\alpha = 0.5$, since the stencil can decouple into two checkerboard-like subgrids then. For $\alpha < 0.5$ one has strict positive definiteness and no decoupling problems.

3.2 Stability Results for Fully Discrete Diffusion Schemes

The positive semidefinite energy (18) can be rewritten as

$$E(\boldsymbol{u}) \;=\; -\frac{1}{2}\,\boldsymbol{u}^{\top}\boldsymbol{A}\boldsymbol{u} \tag{22}$$

with a negative semidefinite matrix $\boldsymbol{A} \in \mathbb{R}^{NM\times NM}$ where each row contains the nine stencil entries of the corresponding spatial node. Its gradient descent

$$\frac{d\boldsymbol{u}}{dt} \;=\; \boldsymbol{A}\,\boldsymbol{u} \tag{23}$$

is a space-discrete and time-continuous anisotropic diffusion process. Let us now consider two common time discretisations of this dynamical system.

Explicit Time Discretisation. An explicit scheme with step size τ is given by

$$\frac{\boldsymbol{u}^{k+1} - \boldsymbol{u}^k}{\tau} \;=\; \boldsymbol{A}^k \, \boldsymbol{u}^k \,, \tag{24}$$

where the upper index denotes the time level. It can be written as

$$\boldsymbol{u}^{k+1} \;=\; (\boldsymbol{I} + \tau \boldsymbol{A}^k) \, \boldsymbol{u}^k \,. \tag{25}$$

Stability in the Euclidean norm requires $\|\boldsymbol{u}^{k+1}\|_2 \;\leq\; \|\boldsymbol{u}^k\|_2$. This is guaranteed for $\rho(\boldsymbol{I} + \tau \boldsymbol{A}^k) \leq 1$, where ρ denotes the spectral norm. For a negative semidefinite $\boldsymbol{A}^k$, this comes down to

$$\tau \;\leq\; \frac{2}{\rho(\boldsymbol{A}^k)} \,. \tag{26}$$

An estimate for $\rho(\boldsymbol{A}^k)$ can be derived via Gershgorin's Theorem. The stability bound (26) also allows to design extremely efficient variants of (24), so-called *fast explicit diffusion (FED)* schemes [19]. They use cycles of varying time steps, preserve the L^2-stability of the underlying scheme, and are well-suited for GPUs.

Semi-implicit Time Discretisation. The semi-implicit scheme

$$\frac{\boldsymbol{u}^{k+1} - \boldsymbol{u}^k}{\tau} \;=\; \boldsymbol{A}^k \, \boldsymbol{u}^{k+1} \tag{27}$$

requires to solve a linear system of equations:

$$\left(\boldsymbol{I} - \tau \boldsymbol{A}^k\right) \, \boldsymbol{u}^{k+1} \;=\; \boldsymbol{u}^k \,. \tag{28}$$

For negative semidefinite $\boldsymbol{A}^k$, the matrix $\boldsymbol{I} - \tau \boldsymbol{A}^k$ has only eigenvalues ≥ 1 and is thus invertible. Since $\rho((\boldsymbol{I} - \tau \boldsymbol{A}^k)^{-1}) \leq 1$, the semi-implicit scheme

$$\boldsymbol{u}^{k+1} \;=\; \left(\boldsymbol{I} - \tau \boldsymbol{A}^k\right)^{-1} \, \boldsymbol{u}^k \tag{29}$$

is absolutely stable in the Euclidean norm.

4 Evaluation of Specific Discretisations

Now that we have derived a general class of L^2-stable discretisations for anisotropic diffusion processes, let us study the performance of different parameter settings. Since more recent applications of anisotropic diffusion focus on its interpolation quality (see e.g. [3, 5]), we consider an idealised demosaicking scenario, where we know the ground truth solution and where subpixel accuracy w.r.t. the interpolation direction plays an important role. For other applications such as denoising and image enhancement we have found similar performance rankings.

Demosaicking addresses a problem of many camera sensors: They use a colour filter array which allows them to measure only one out of three colour channels

Table 2. Performance of different space discretisations

discretisation	α	β	PSNR [dB]
standard discretisation [11]	0	0	24.60
Cottet and El-Ayyadi [2]	0	-1	25.38
nonnegativity discretisation [1]	0	$\operatorname{sign}(b)$	29.86
Mrázek and Navara II [12]	$\frac{\min(a,c)}{a+c}$	0	24.17
Mrázek and Navara III [12]	$\frac{\min(a,c)}{2(a+c)}$	$0.5\operatorname{sign}(b)$	27.42
wavelet-inspired scheme I [15]	0.5	0	29.57
wavelet-inspired scheme II [16]	0.49	0	32.88
our nonstandard stencil	0.44	$0.118\operatorname{sign}(b)$	**33.99**

at each pixel location: either red (R), green (G), or blue (B). Thus, exactly one third of the colour image information is available, and two thirds must be interpolated. Often the colour information is arranged in the order of the so-called *Bayer array* that consists of a periodic repetition of the pattern

R	G
G	B

(30)

For our evaluation we consider the synthetic test image in Figure 1(a). It consists of concentric circular structures of varying frequencies. Thus, it allows us to assess the directional and the frequency behaviour of anisotropic diffusion interpolation. By removing two thirds of the colour information by means of the Bayer mask, we obtain the image in Figure 1(b). Now we interpolate the missing information at the unspecified channels in each pixel by evolving an explicit anisotropic diffusion scheme to its steady state. The data at the specified pixels serve as Dirichlet boundary conditions. In this synthetic example we can design a positive semidefinite diffusion tensor in such a way that only diffusion in tangential direction is allowed. Thus, the directional error of $\boldsymbol{D}$ is zero, and all reconstruction errors are caused by limitations of our space discretisation due to dissipative artifacts or deviations from rotation invariance.

Table 2 compares the peak signal-to-noise ratio (PSNR) between the interpolated image and the original image. In spite of the fact that we interpolate over a distance of at most two pixels and that we prescribe the correct interpolation direction, we observe that the seven stencils from the literature differ strongly in their performance: While the standard discretisation and the Mrázek–Navara scheme II perform fairly bad, the nonnegativity discretisation and the wavelet-inspired scheme II with $\alpha = 0.49$ give rather good results. It should be noted that the wavelet-inspired scheme I always uses $\alpha = 0.5$. Hence, it suffers from the before mentioned checkerboard decoupling which is particularly undesirable for demosaicking. Therefore, we recommend to use only stencils with $\alpha \leq 0.49$.

We see that all seven schemes from the literature can be outperformed by our nonstandard scheme with suitable parameters. Its demosaicking result is

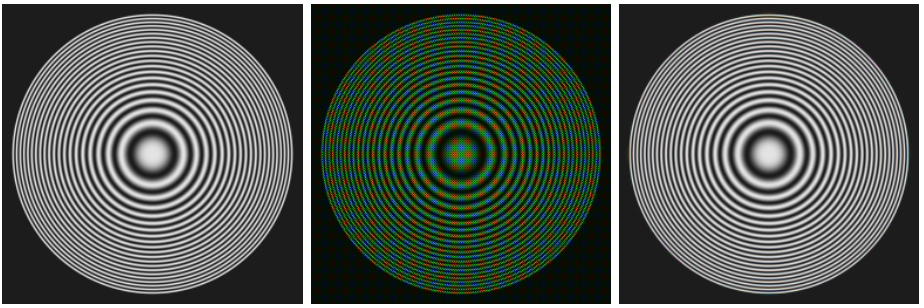


Fig. 1. (a) Left: Test image, 256×256 pixels. **(b) Middle:** After applying the Bayer colour filter array. **(c) Right:** Demosaicking result with our nonstandard scheme with $\alpha = 0.44$ and $\beta = 0.118\,\mathrm{sign}(b)$.

depicted in Figure 1(c). Since all stencils from Table 2 have the same consistency order (namely 2), it is remarkable that their actual performance is so different: The PSNR difference between the best and the worst stencil is 9.82 dB! This confirms the fundamental importance of a good discretisation when one wants to use anisotropic diffusion processes with a diffusion tensor. Similar findings have also been made in [12, 13, 15, 16].

The best 3×3 finite difference stencil in the anisotropic diffusion literature is given by the wavelet-inspired filter class II [16]. It contains α as a free parameter. Our nonstandard stencil class identifies β as a second degree of freedom. According to Proposition 1, β has to fulfil $|\beta| \leq 1 - 2\alpha$. Often it is advisable to use a β-value that has the same sign as b, since this can help to reduce the well-known over- and undershoots due to a lack of nonnegativity in the stencil. Thus, it is convenient to replace the parameter β by a parameter γ that is linked to $\mathrm{sign}(b)$ and can vary in the interval $[-1, 1]$ for all $\alpha < \frac{1}{2}$:

$$\beta = \gamma \cdot (1 - 2\alpha) \cdot \mathrm{sign}(b)\,. \tag{31}$$

For example, $\beta = 0.118\,\mathrm{sign}(b)$ in Table 2 can be expressed by $\gamma = 0.98$.

Table 3 illustrates the advantages of our nonstandard stencil over the wavelet-inspired stencil II. For small values of α, one can easily improve the PSNR in the demosaicking test case by more than 5 dB: All one has to do is to choose $\gamma = 1$ instead of $\gamma = 0$. The latter corresponds to the wavelet-inspired stencil II. As long as α is not too close to the critical value $\frac{1}{2}$ (which should be avoided anyway due to checkerboard artifacts) it turns out that $\gamma = 1$ gives the highest PSNR. However, even for $\alpha \in [0.43, 0.49]$, where $\gamma = 1$ is suboptimal, it still outperforms $\gamma = 0$. Thus, choosing $\gamma = 1$ works well in practice and reduces the parameter space to a single degree of freedom.

We see that within our stencil class, schemes with fairly large values for α and γ perform particularly well. This confirms Mickens' principle of nonlocal approximation of nonlinear terms: The more α and γ differ from 0, the larger is the contribution of the nonstandard finite difference terms within (10)–(12).

Table 3. Comparison between the wavelet-inspired filter class II from [16] and our nonstandard filter class. The table depicts the PSNR for the demosaicking test scenario, and γ_{opt} refers to the γ-value where the nonstandard stencil yields the highest PSNR.

α	wavelet-insp. II	**nonstandard**	γ_{opt}
0	24.60	29.86	1
0.1	25.27	30.81	1
0.2	26.14	31.90	1
0.3	27.32	33.04	1
0.4	29.08	33.83	1
0.42	29.57	33.87	1
0.44	30.16	**33.99**	0.98
0.46	30.90	33.96	0.96
0.48	31.98	33.87	0.90
0.49	**32.88**	33.78	0.80
0.5	29.57	29.57	–

5 Conclusions

We have shown that seven finite difference discretisations for anisotropic diffusion filtering with a diffusion tensor are special cases of a novel, unifying framework. It is derived systematically from a discrete energy formulation, and it exploits the widely unknown nonstandard finite differences of Mickens [18]. We have established general L^2-stability results. Our framework does not only provide a theoretical foundation of existing schemes as L^2-stable discrete energy minimisers, but also comprises novel stencils that outperform existing ones.

Our evaluation has shown that different discretisations of the same continuous model can give PSNR differences of almost 10 dB, even though they have identical consistency order. This confirms the widely underestimated fact that appropriate numerical algorithms are at least as important as good models.

We expect that the ideas in our paper can also be generalised to other anisotropic equations that create similar numerical challenges.

References

1. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner, Stuttgart (1998)
2. Cottet, G.H., El Ayyadi, M.: A Volterra type model for image processing. IEEE Transactions on Image Processing 7(3), 292–303 (1998)
3. 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)
4. 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)
5. Zimmer, H., Bruhn, A., Weickert, J.: Optic flow in harmony. International Journal of Computer Vision 93(3), 368–388 (2011)

6. 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)
7. Schroers, C., Zimmer, H., Valgaerts, L., Bruhn, A., Demetz, O., Weickert, J.: Anisotropic range image integration. In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) DAGM and OAGM 2012. LNCS, vol. 7476, pp. 73–82. Springer, Heidelberg (2012)
8. Preußer, T., Rumpf, M.: An adaptive finite element method for large scale image processing. Journal of Visual Communication and Image Representation 11(2), 183–195 (2000)
9. Drblíková, O., Mikula, K.: Convergence analysis of finite volume scheme for nonlinear tensor anisotropic diffusion in image processing. SIAM Journal on Numerical Analysis 46(1), 37–60 (2007)
10. Jawerth, B., Lin, P., Sinzinger, E.: Lattice Boltzmann models for anisotropic diffusion of images. Journal of Mathematical Imaging and Vision 11, 231–237 (1999)
11. Weickert, J.: Nonlinear diffusion filtering. In: Jähne, B., Haußecker, H., Geißler, P. (eds.) Handbook on Computer Vision and Applications. Signal Processing and Pattern Recognition, pp. 423–450. Academic Press, San Diego (1999)
12. Mrázek, P., Navara, M.: Consistent positive directional splitting of anisotropic diffusion. In: Likar, B. (ed.) Proc. Sixth Computer Vision Winter Workshop, Bled, Slovenia, February 2001, pp. 37–48 (2001)
13. Weickert, J., Scharr, H.: A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. Journal of Visual Communication and Image Representation 13(1/2), 103–118 (2002)
14. Felsberg, M.: Autocorrelation-driven diffusion filtering. IEEE Transactions on Image Processing 20(7), 1797–1806 (2011)
15. Welk, M., Weickert, J., Steidl, G.: From tensor-driven diffusion to anisotropic wavelet shrinkage. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 391–403. Springer, Heidelberg (2006)
16. 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)
17. Dascal, L., Ditkowski, A., Sochen, N.: A study of the discrete maximum principle for the Beltrami color flow. Journal of Mathematical Imaging and Vision 29(1), 63–77 (2007)
18. Mickens, R.E.: Nonstandard Finite Difference Models of Differential Equations. World Scientific, Singapore (1994)
19. Grewenig, S., Weickert, J., Bruhn, A.: From box filtering to fast explicit diffusion. In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) Pattern Recognition. LNCS, vol. 6376, pp. 533–542. Springer, Heidelberg (2010)

Relations between Amoeba Median Algorithms and Curvature-Based PDEs

Martin Welk

University for Health Sciences, Medical Informatics and Technology (UMIT),
Eduard-Wallnöfer-Zentrum 1, 6060 Hall/Tyrol, Austria
martin.welk@umit.at

Abstract. This paper is concerned with the theoretical analysis of structure-adaptive median filter algorithms that approximate curvature-based PDEs for image filtering and segmentation. These so-called morphological amoeba filters, introduced by Lerallut et al. and further developped by Welk et al., achieve similar results as the well-known geodesic active contour and self-snakes PDEs. In the present work, the PDE approximated by amoeba active contours is derived in the general case. This PDE is structurally similar but not identical to the geodesic active contour equation. Implications for the qualitative behaviour of amoeba active contours as well as for the approximation of the pre-smoothed self-snakes equation are investigated.

1 Introduction

Introduced by Lerallut et al. [11,12], morphological amoeba filtering is a class of discrete image filtering procedures based on image-adaptive structuring elements. These structuring elements are defined by a so-called amoeba metric that combines spatial proximity and grey-value similarity. Amoeba filters adapt flexibly to image structures. For example, iterated *amoeba median filtering* (AMF) improves the favourable edge-preserving denoising capabilities of traditional iterated median filtering [17] by removing its tendency to dislocate edges, and introducing even edge-enhancing behaviour.

Extending the author's earlier work with co-authors [18,19], this paper is concerned with comparing AMF methods to two curvature-based PDEs of image processing. Firstly, we consider *geodesic active contours* [3,4,8,9]

$$u_t = |\nabla u| \operatorname{div}\left(g(|\nabla f|^2)\,\frac{\nabla u}{|\nabla u|}\right) \tag{1}$$

which can be used to segment a given image f by evolving a contour towards regions of high contrast in f. The evolving contour is encoded as zero-level set of the function u. The (decreasing, nonnegative) edge-stopping function g can be chosen e.g. as a Perona-Malik-type function [15]

$$g(s^2) = \frac{1}{1+s^2/\lambda^2}\;, \qquad \lambda > 0\;. \tag{2}$$

Secondly, we are interested in *self-snakes* [16], a PDE filter for a single image u that is obtained from (1) by identifying f with the evolving function u.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 392–403, 2013.

As shown in [19], AMF is linked to the self-snakes equation in a way similar to the connection of traditional median filtering to (mean) curvature motion [1] that was proven by Guichard and Morel [6]: One amoeba median filtering step asymptotically approximates a time step of size $\varrho^2/6$ of an explicit time discretisation for the self-snakes PDE when the radius ϱ of the structuring element goes to zero. The exact shape of the (decreasing, nonnegative) edge-stopping function g depends on the specific choice of the amoeba metric, with the Perona-Malik-type function (2) being associated to the L^2 amoeba metric.

Building on this amoeba/self-snakes connection, [18] proposed a morphological amoeba algorithm for active contour segmentation. Experimentally, this process behaves similar to geodesic active contours, with a tendency to refined adaptation to structure details, see [18, Fig. 2]. Analysis in [18] was restricted to a rotationally symmetric situation where asymptotic equivalence to geodesic active contours (1) could be proven. The present paper aims at closing this gap in theoretical analysis.

Writing the self-snakes equation ((1) with $f \equiv u$) as $u_t = g \cdot |\boldsymbol{\nabla} u| \operatorname{div}(\boldsymbol{\nabla} u/|\boldsymbol{\nabla} u|) + \langle \boldsymbol{\nabla} g, \boldsymbol{\nabla} u \rangle$ accentuates an important difference between the (mean) curvature motion equation $u_t = |\boldsymbol{\nabla} u| \operatorname{div}(\boldsymbol{\nabla} u/|\boldsymbol{\nabla} u|)$ and self-snakes: the edge-enhancing component $\langle \boldsymbol{\nabla} g, \boldsymbol{\nabla} u \rangle$ is related to a shock filter [14,16] or backward diffusion [16]. Analytically, this makes the self-snakes PDE ill-posed, and in particular induces staircasing behaviour [20]. Numerically, this shock component needs specific consideration. In finite-difference discretisations, it is usually treated by an upwind discretisation [13]. Still, severe numerical dissipation artifacts appear. As [19] demonstrates, results depend heavily on the grid mesh size, rendering the approximation of the PDE unreliable.

One approach to defeat these undesired phenomena on the PDE level itself, and to construct a PDE that can properly be numerically approximated, is pre-smoothing [5]. To this end, one replaces $\boldsymbol{\nabla} u$ in the argument of g by a smoothed version like $\boldsymbol{\nabla} u_\sigma := K_\sigma * \boldsymbol{\nabla} u$ where K_σ denotes a Gaussian of standard deviation σ.

As [19] suggests, AMF can be considered as an unconventional discretisation of self-snakes. Experiments in [19] indicate that it is less susceptible to the above-mentioned sort of artifacts. This indicates that the AMF procedure also acts in some way regularising. To make a first step towards a better understanding of the regularisation effects of pre-smoothing and amoeba filtering is another objective of this work.

Our Contribution. We extend the analytical investigation of amoeba filters. First, we derive the PDE corresponding to the amoeba active contour method in the general case, which is no longer fully identical to the geodesic active contour equation. To this end, we introduce a proof strategy substantially different from that used in [18,19]. Qualitative differences between geodesic and amoeba active contours are discussed based on the approximation result.

Finally, we apply our extended analysis of amoeba active contours to amoeba approximation of pre-smoothed self-snakes.

Structure of the Paper. We give a short account of the basic concepts of amoeba filtering in Section 2. Our main theoretical result on PDE approximation is proven in

Section 3. It is used for comparing amoeba active contours to geodesic active contours in Section 4. Pre-smoothing in the self-snakes PDE and its approximation in the amoeba framework is discussed in Section 5, followed by a conclusion in Section 6.

2 Amoeba Filters

In this section we recall shortly the definition of amoeba metrics and amoeba filters. We assume that a 2D image is given as a smooth function $f : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^2$ is closed.

Amoeba Metrics. Following the spatially continuous formulation of the amoeba framework in [18,19], we associate with f the image manifold $\Gamma \subset \mathbb{R}^3$ consisting of the points $(x, y, \beta f(x, y))$. As a Riemannian metric on Γ, an *amoeba metric* is given by

$$\mathrm{d}_\nu s = \nu\left(\sqrt{\mathrm{d}x^2 + \mathrm{d}y^2}, \beta\, \mathrm{d}f\right) , \tag{3}$$

where ν is some norm on $\mathbb{R}^2$. The use of the Euclidean norm $\sqrt{\mathrm{d}x^2 + \mathrm{d}y^2}$ in the spatial component ensures rotational invariance of the amoeba metric, while the combination of spatial and tonal distances is governed by ν. The factor β is a scale that balances the spatial and tonal information.

The *amoeba distance* $d(\boldsymbol{p}, \boldsymbol{q})$ between two points $\boldsymbol{p}$, $\boldsymbol{q}$ of the image domain is the minimum of $L(\boldsymbol{c}) = \int_{\boldsymbol{c}} \mathrm{d}_\nu s$ among all curves $\boldsymbol{c}$ connecting $\boldsymbol{p}$ with $\boldsymbol{q}$.

Continuous-Scale Amoeba Filtering Formulation. For amoeba filters, one defines a structuring element $\mathcal{A}_{\boldsymbol{p}}$ for each point $\boldsymbol{p} \in \Omega$ as the set of all $\boldsymbol{q} \in \Omega$ such that $d(\boldsymbol{p}, \boldsymbol{q}) \leq \varrho$, where the global parameter ϱ is the *amoeba radius*. With the so defined structuring elements several morphological filters can be carried out straightforward. In particular, for amoeba median filtering (AMF), the median of the grey-values of the given image f within $\mathcal{A}_{\boldsymbol{p}}$ becomes the filtered grey-value at $\boldsymbol{p}$. Like traditional median filtering, this filter can be applied iteratively. This process was studied in [19].

Amoeba Active Contours. The amoeba active contour method described in [18] acts in a similar way: Structuring elements are determined as before but on the basis of the given image f, and are used for median-filtering the evolving level-set function u.

Discrete Amoeba Filtering Algorithms. Practically, computations are done on discrete images, using a discrete version of the above-mentioned amoeba distance obtained by restricting curves to paths in the neighbourhood graph of the image grid, either with 4-neighbourhoods as in [11,12] or with 8-neighbourhoods as in [18,19]. More sophisticated constructions using geometric distance transforms [2,7] would be possible.

Choice of the Amoeba Metric for the Analysis. In the following, we use the L^2 amoeba metric given by $\nu(s, t) = \sqrt{s^2 + t^2}$. The amoeba metric parameter β can be fixed to 1 since a change of this parameter is equivalent to a simple rescaling of the steering function f.

3 Analysis of Amoeba Active Contours

We study an amoeba median filter for $\varrho \to 0$, in which f is a smooth function from which the amoeba structuring elements are generated, and u is another smooth function, to which the median filter is applied. In our analysis, local orthonormal bases aligned to the gradient and level-line directions of both functions will play an important role. Given a location $\boldsymbol{x}_0$ in the image domain, we will therefore denote by $\boldsymbol{\chi} = (\cos\varphi, \sin\varphi)^{\mathrm{T}}$ the normalised gradient vector of f at $\boldsymbol{x}_0$. The unit vector $\boldsymbol{\zeta} \perp \boldsymbol{\chi}$ then indicates the local level line direction of f. Analogously, we denote by $\boldsymbol{\eta}$ a normalised gradient vector for u, and by $\boldsymbol{\xi} \perp \boldsymbol{\eta}$ the unit vector in the level line direction. The angle between the gradient directions will be called α, such that $\boldsymbol{\eta} = \big(\cos(\varphi+\alpha), \sin(\varphi+\alpha)\big)^{\mathrm{T}}$. We will prove the following fact.

Theorem 1. *One step of amoeba median filtering of a smooth function u governed by amoebas generated from f with an amoeba radius of ϱ asymptotically approximates a time step of size $\tau = \varrho^2/6$ of an explicit time discretisation for the PDE*

$$u_t = \frac{u_{\boldsymbol{\xi}\boldsymbol{\xi}}}{1+|\boldsymbol{\nabla} f|^2 \sin^2\alpha} - \frac{|\boldsymbol{\nabla} f|\,|\boldsymbol{\nabla} u|}{1+|\boldsymbol{\nabla} f|^2 \sin^2\alpha} \cdot \left(\frac{f_{\boldsymbol{\zeta}\boldsymbol{\zeta}} \cos^3\alpha}{1+|\boldsymbol{\nabla} f|^2} + 2\, f_{\boldsymbol{\zeta}\boldsymbol{\chi}} \sin^3\alpha + \frac{f_{\boldsymbol{\chi}\boldsymbol{\chi}} \cos\alpha \left(2+\sin^2\alpha+3\,|\boldsymbol{\nabla} f|^2 \sin^2\alpha\right)}{\left(1+|\boldsymbol{\nabla} f|^2\right)^2} \right). \quad (4)$$

Remark on the Proof Strategy. The proofs in [18,19] were based on measuring level line segments within the amoeba. Throughout the proofs, Taylor coefficients of f and u up to second order were used in the calculations. This strategy could be followed in the more specialised cases treated in those papers. However, the complexity of such calculations would increase a lot in the general case we are about to discuss. In the following proof of the theorem we follow therefore a different strategy that measures areas not segments but sectors of amoebas via a polar coordinate representation. Level lines other than the one through the amoeba centre are not considered directly any more.

Finding the Amoeba Contour. To determine the shape of the amoeba $\mathcal{A} := \mathcal{A}_{\boldsymbol{x}_0}$ around a point $\boldsymbol{x}_0 \in \Omega$, we start by considering the 1D case: given $f : \mathbb{R} \to \mathbb{R}$, we seek $z_\pm \in \mathbb{R}$ such that the arc-length of the image graph of f between x_0 and each of $x_0 + z_+$, $x_0 - z_-$ equals ϱ. Certainly, $z_\pm \le \varrho$.

Using Taylor expansions for f and the square root function, we have for the arc-length from x_0 to $x_0 + z$ (where $z > 0$)

$$\int_{x_0}^{x_0+z} \sqrt{1+f'(x)^2}\,\mathrm{d}x = z\sqrt{1+f'(x_0)^2} + \frac{z^2}{2}\,\frac{f'(x_0)\,f''(x_0)}{\sqrt{1+f'(x_0)^2}} + \mathcal{O}(\varrho^3)\,. \quad (5)$$

Equating this to ϱ yields a quadratic equation in z with the solutions

$$z_{1,2} = \frac{1+f'(x_0)^2}{f'(x_0)\,f''(x_0)} \left(-1 \pm \sqrt{1+\varrho\,\frac{f'(x_0)\,f''(x_0)}{\big(1+f'(x_0)\big)^{3/2}}} \right) + \mathcal{O}(\varrho^3) \quad (6)$$

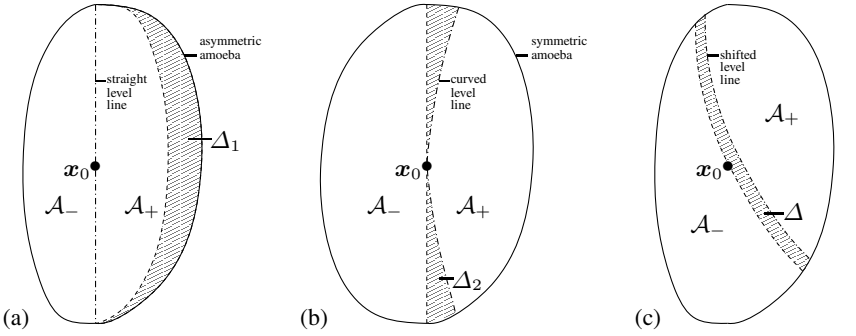


Fig. 1. Left to right: (a) Area difference Δ_1 in an asymmetric amoeba with straight level lines. – **(b)** Area difference Δ_2 in a symmetric amoeba with curved level lines. – **(c)** Compensation of the area difference Δ by shifting the central level line (schematic).

which gives z_+ as the "+" case (because of $z > 0$). Using again the Taylor expansion of the square root function, and doing an analogous derivation for z_-, we arrive at

$$z_\pm = \frac{\varrho}{\sqrt{1 + f'(x_0)^2}} \mp \frac{\varrho^2 \, f'(x_0) \, f''(x_0)}{2\big(1 + f'(x_0)^2\big)^2} + \mathcal{O}(\varrho^3) \; . \tag{7}$$

Turning to the 2D case, we approximate each shortest path in the amoeba metric from $\boldsymbol{x}_0$ to a point on the amoeba contour by a Euclidean straight line in the image plane. This introduces only an $\mathcal{O}(\varrho^3)$ error for the path length. We consider now the straight line through $\boldsymbol{x}_0$ in the direction of a given unit vector $\boldsymbol{v} \in \mathbb{R}^2$. By our previous 1D result, with the directional derivatives $f_{\boldsymbol{v}}(\boldsymbol{x}_0) = \langle \boldsymbol{v}, \boldsymbol{\nabla} f(\boldsymbol{x}_0) \rangle$ and $f_{\boldsymbol{v}\boldsymbol{v}}(\boldsymbol{x}_0) = \boldsymbol{v}^{\mathrm{T}} \, \mathrm{D}^2 f(\boldsymbol{x}_0) \, \boldsymbol{v}$, we see that said straight line intersects the amoeba contour at $\boldsymbol{x}_0 \pm z_\pm(\boldsymbol{v}) \cdot \boldsymbol{v}$ with

$$z_\pm(\boldsymbol{v}) = \frac{\varrho}{\sqrt{1 + \big\langle \boldsymbol{v}, \boldsymbol{\nabla} f(\boldsymbol{x}_0) \big\rangle^2}} \mp \frac{\varrho^2 \, \big\langle \boldsymbol{v}, \boldsymbol{\nabla} f(\boldsymbol{x}_0) \big\rangle \, \boldsymbol{v}^{\mathrm{T}} \, \mathrm{D}^2 f(\boldsymbol{x}_0) \, \boldsymbol{v}}{2 \, \Big(1 + \big\langle \boldsymbol{v}, \boldsymbol{\nabla} f(\boldsymbol{x}_0) \big\rangle^2\Big)^2} + \mathcal{O}(\varrho^3) \; . \tag{8}$$

Contributions to the Amoeba Median. The median of u within the structuring element $\mathcal{A}$ equals $u(\boldsymbol{x}_0)$ if (a) the amoeba is point-symmetric w.r.t. $\boldsymbol{x}_0$, and (b) the level lines of u are straight: The central level line $u(\boldsymbol{x}) = u(\boldsymbol{x}_0)$ of u then bisects $\mathcal{A}$, i.e. $\mathcal{A}_+ := \{\boldsymbol{x} \in \mathcal{A} \mid u(\boldsymbol{x}) \geq u(\boldsymbol{x}_0)\}$ and $\mathcal{A}_- := \{\boldsymbol{x} \in \mathcal{A} \mid u(\boldsymbol{x}) \leq u(\boldsymbol{x}_0)\}$ have equal area. For a similar bisection approach in a gradient descent for segmentation compare [10].

Deviations from conditions (a) and (b) lead to imbalances between $\mathcal{A}_+$ and $\mathcal{A}_-$. The median is determined by the shift of the central level line that is necessary to compensate for the resulting area difference. The separate area effects of asymmetry of the amoeba, and curvature of u's level lines are of order $\mathcal{O}(\varrho^3)$, while any cross-effects are at least of order $\mathcal{O}(\varrho^4)$, and can be neglected for the purpose of our analysis. Therefore, the two effects can be studied independently.

Asymmetry of the Amoeba. We start by analysing the effect of asymmetries of the point set $\mathcal{A}$, compare Figure 1(a). As the amoeba shape is governed by f, we will use the $\boldsymbol{\zeta}$,

$\boldsymbol{\chi}$ local coordinates. For an arbitrary unit vector $\boldsymbol{v} = \big(\cos(\varphi+\vartheta), \sin(\varphi+\vartheta)\big)^{\mathrm{T}}$ we have then

$$f_{\boldsymbol{v}}(\boldsymbol{x}_0) = |\boldsymbol{\nabla} f(\boldsymbol{x}_0)| \cos\vartheta \;, \tag{9}$$

$$\boldsymbol{v}^{\mathrm{T}}\, \mathrm{D}^2 f(\boldsymbol{x}_0)\, \boldsymbol{v} = f_{\boldsymbol{\zeta\zeta}} \sin^2\vartheta + 2\, f_{\boldsymbol{\zeta\chi}} \cos\vartheta\, \sin\vartheta + f_{\boldsymbol{\chi\chi}} \cos^2\vartheta \tag{10}$$

which can be inserted into (8) to obtain $z_{\pm}(\varphi+\vartheta) := z_{\pm}(\boldsymbol{v})$.

Assume now that u has straight level lines; remember that $\varphi+\alpha$ is the direction angle of its gradient direction. Since the amoeba shape is given by $z_{\pm}(\boldsymbol{v})$ in polar coordinates, the sought area difference is then obtained as

$$\Delta_1 := |\mathcal{A}_+| - |\mathcal{A}_-| = \int\limits_{\varphi+\alpha-\pi/2}^{\varphi+\alpha+\pi/2} \big(z_+(\vartheta) - z_-(\vartheta)\big)\, \frac{z_+(\vartheta) + z_-(\vartheta)}{2}\, \mathrm{d}\vartheta + \mathcal{O}(\varrho^4) \;. \tag{11}$$

The integral on the right-hand side equals

$$-\varrho^3\, |\boldsymbol{\nabla} f| \int\limits_{\alpha-\pi/2}^{\alpha+\pi/2} \frac{f_{\boldsymbol{\zeta\zeta}}\, \cos\vartheta\, \sin^2\vartheta + 2\, f_{\boldsymbol{\zeta\chi}}\, \cos^2\vartheta\, \sin\vartheta + f_{\boldsymbol{\chi\chi}}\, \cos^3\vartheta}{\big(1 + |\boldsymbol{\nabla} f|^2 \cos^2\vartheta\big)^{5/2}}\, \mathrm{d}\vartheta \tag{12}$$

which evaluates to

$$-\frac{2}{3}\, \varrho^3\, |\boldsymbol{\nabla} f| \Bigg(\frac{f_{\boldsymbol{\zeta\zeta}}\, \cos^3\alpha}{\big(1+|\boldsymbol{\nabla} f|^2\big)\big(1+|\boldsymbol{\nabla} f|^2\, \sin^2\alpha\big)^{3/2}} + \frac{2\, f_{\boldsymbol{\zeta\chi}}\, \sin^3\alpha}{\big(1+|\boldsymbol{\nabla} f|^2\, \sin^2\alpha\big)^{3/2}} + \frac{f_{\boldsymbol{\chi\chi}}\, \cos\alpha\, \big(2+\sin^2\alpha+3\,|\boldsymbol{\nabla} f|^2\, \sin^2\alpha\big)}{\big(1+|\boldsymbol{\nabla} f|^2\big)^2 \big(1+|\boldsymbol{\nabla} f|^2\, \sin^2\alpha\big)^{3/2}} \Bigg) \;. \tag{13}$$

Curvature of the Level Lines. The second source of area imbalance between $\mathcal{A}_+$ and $\mathcal{A}_-$ is the curvature of the level line of u through $\boldsymbol{x}_0$. Using the $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ local coordinates pertaining to u, this curvature equals $u_{\boldsymbol{\xi\xi}}/(2|\boldsymbol{\nabla} u|)$. The resulting area difference is

$$\begin{aligned} \Delta_2 := |\mathcal{A}_+| - |\mathcal{A}_-| &= -2 \int\limits_{-z_-(\varphi+\alpha+\pi/2)}^{z_+(\varphi+\alpha+\pi/2)} -\frac{u_{\boldsymbol{\xi\xi}}}{2\,|\boldsymbol{\nabla} u|} z^2\, \mathrm{d}z + \mathcal{O}(\varrho^4) \\ &= \frac{2}{3}\, \frac{u_{\boldsymbol{\xi\xi}}}{|\boldsymbol{\nabla} u|}\, \frac{\varrho^3}{\big(1+|\boldsymbol{\nabla} f|^2\, \sin^2\alpha\big)^{3/2}} + \mathcal{O}(\varrho^4) \;. \end{aligned} \tag{14}$$

Median Calculation. As the median μ of u within $\mathcal{A}$ belongs to the level line of u that bisects the area of the amoeba, the difference $\mu - u(\boldsymbol{x}_0)$ corresponds to a shift of the central level line that compensates the area difference $\Delta_1 + \Delta_2$. This compensation is obtained when

$$2\, \frac{\mu - u(\boldsymbol{x}_0)}{|\boldsymbol{\nabla} u|} \cdot \big(z_+(\varphi+\alpha+\pi/2) + z_-(\varphi+\alpha+\pi/2)\big) = \Delta_1 + \Delta_2 + \mathcal{O}(\varrho^4) \;, \tag{15}$$

which finally gives $\mu = u(\boldsymbol{x}_0) + (\varrho^2/6) \cdot u_t$ with u_t given by (4) up to an error $\mathcal{O}(\varrho)$. This concludes the proof of Theorem 1.

Special Cases. The following two statements reproduce the more specialised approximation results from [19] (in the case of the L^2 amoeba metric) and [18], respectively.

Corollary 1. *The amoeba median filter with $f \equiv u/\lambda$ approximates the self-snakes equation*

$$\begin{aligned} u_t &= \frac{u_{\boldsymbol{\xi\xi}}}{1+|\boldsymbol{\nabla} u|^2/\lambda^2} - \frac{2\,u_{\boldsymbol{\eta\eta}}\,|\boldsymbol{\nabla} u|^2}{\lambda^2\left(1+|\boldsymbol{\nabla} u|^2/\lambda^2\right)^2} \\ &= |\boldsymbol{\nabla} u|\,\mathrm{div}\left(\frac{1}{1+|\boldsymbol{\nabla} u|^2/\lambda^2}\,\frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|}\right) \end{aligned} \tag{16}$$

in the sense of Theorem 1.

Corollary 2. *If input image f and evolving level-set image u are rotationally symmetric with respect to the origin, amoeba median filtering approximates the geodesic active contour equation*

$$u_t = \frac{u_{\boldsymbol{\xi\xi}}}{1+|\boldsymbol{\nabla} f|^2} - \frac{2\,f_{\boldsymbol{\eta\eta}}\,|\boldsymbol{\nabla} u|\,|\boldsymbol{\nabla} f|}{\left(1+|\boldsymbol{\nabla} f|^2\right)^2} = |\boldsymbol{\nabla} u|\,\mathrm{div}\left(\frac{1}{1+|\boldsymbol{\nabla} f|^2}\,\frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|}\right) \tag{17}$$

in the sense of Theorem 1.

In the case of Corollary 1, one observes that its hypothesis entails that the identities $\alpha = 0$, $\boldsymbol{\zeta} = \boldsymbol{\xi}$, and $\boldsymbol{\chi} = \boldsymbol{\eta}$ hold everywhere. For Corollary 2, the assumed rotational symmetry yields $\alpha = 0$, $\boldsymbol{\zeta} = \boldsymbol{\xi}$, $\boldsymbol{\chi} = \boldsymbol{\eta}$, $u_{\boldsymbol{\xi\eta}} \equiv f_{\boldsymbol{\xi\eta}} \equiv 0$, and $u_{\boldsymbol{\xi\xi}}/u_{\boldsymbol{\eta}} \equiv f_{\boldsymbol{\xi\xi}}/f_{\boldsymbol{\eta}}$. Substituting the respective sets of identities into (4) implies the corollaries.

4 Comparison to Geodesic Active Contours

In the general amoeba active contour setting, however, it is evident that equation (4) does not exactly coincide with (1). For a better understanding of the differences between both active contour methods, we consider further typical configurations.

Homogeneous Image Gradients. In flat image regions ($\boldsymbol{\nabla} f = \mathbf{0}$), geodesic active contours (1) as well as amoeba active contours evolve the level set function u by curvature motion. Let us consider now an image region with a homogeneous non-zero gradient, $\boldsymbol{\nabla} f = \mathrm{const}$. In such a region, geodesic active contours still perform curvature motion, but with an evolution speed slowed down by the contrast-dependent factor $g(|\boldsymbol{\nabla} f|^2) = 1/(1+|\boldsymbol{\nabla} f|^2)$. The amoeba-based PDE (4) in this case becomes

$$u_t = \frac{u_{\boldsymbol{\xi\xi}}}{1+|\boldsymbol{\nabla} f|^2 \sin^2\alpha}\,, \tag{18}$$

i.e. also a slowed-down curvature motion, but the evolution is slowed down the less, the more the level lines of f and u are aligned. This leads to a faster straightening of aligned contour segments, thereby boosting adaptation of u's level lines to those of f, see the schematic representation in Figure 2(a).

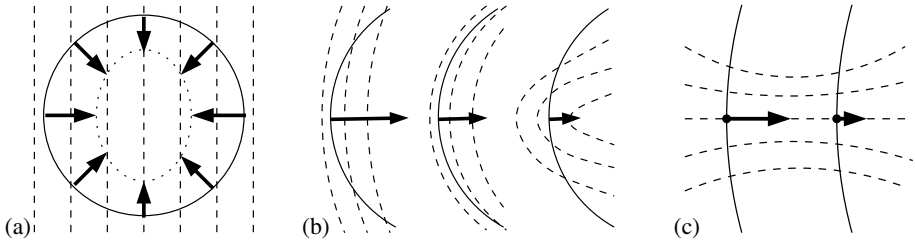


Fig. 2. Evolution of level lines under the PDE (4) in exemplary configurations (schematic). Solid lines: level lines of u, dashed lines: level lines of f. **Left to right: (a)** In a region with homogeneous ∇f, aligned level line segments of u evolve faster. – **(b)** At a location with aligned ∇u and ∇f, the contour evolves inward faster when the curvature of u exceeds that of f. – **(c)** At locations with orthogonal ∇u and ∇f, the curvature-dependent movement of the contour is attracted towards high-contrast regions of f. Assuming that $\boldsymbol{\eta}$ points to the right, $f_{\boldsymbol{\xi\eta}} < 0$ holds in the left, and $f_{\boldsymbol{\xi\eta}} > 0$ in the right part, while $u_{\boldsymbol{\xi\xi}} < 0$ in both cases.

Aligned Gradients. Relaxing the condition of Corollary 2, we assume now that the gradient directions of f and u coincide, $\alpha = 0$, $\boldsymbol{\zeta} = \boldsymbol{\xi}$, $\boldsymbol{\chi} = \boldsymbol{\eta}$, but make no assumption on their curvatures. At such a location, (4) takes the form

$$\begin{aligned} u_t &= u_{\boldsymbol{\xi\xi}} - |\nabla f|\,|\nabla u| \left(\frac{f_{\boldsymbol{\xi\xi}}}{1+|\nabla f|^2} + \frac{2\, f_{\boldsymbol{\eta\eta}}}{\left(1+|\nabla f|^2\right)^2} \right) \\ &= \frac{u_{\boldsymbol{\xi\xi}}}{1+|\nabla f|^2} - \frac{2\,|\nabla f|\,|\nabla u|\, f_{\boldsymbol{\eta\eta}}}{\left(1+|\nabla f|^2\right)^2} + \frac{2\,|\nabla f|^2\,|\nabla u|}{1+|\nabla f|^2} \left(\frac{u_{\boldsymbol{\xi\xi}}}{2\, u_{\boldsymbol{\eta}}} - \frac{f_{\boldsymbol{\xi\xi}}}{2\, f_{\boldsymbol{\eta}}} \right) \end{aligned} \tag{19}$$

which coincides with the corresponding geodesic active contour evolution except for the last summand that speeds up the evolution if the level line curvature $u_{\boldsymbol{\xi\xi}}/(2\, u_{\boldsymbol{\eta}})$ of u exceeds that of f, see Figure 2(b). The same offset is obtained in the anti-aligned case, $\alpha = \pi$, $\boldsymbol{\zeta} = -\boldsymbol{\xi}$, $\boldsymbol{\chi} = -\boldsymbol{\eta}$; note that the curvature of f's level lines is measured with respect to the orientation of u's level lines. Relative to geodesic active contours, this implies an accelerated removal of sharp contour corners that do not match the given image f.

Orthogonal Gradients. Consider now the complementary situation where the gradient directions of u and f are orthogonal, i.e. $\alpha = \pi/2$, $\boldsymbol{\zeta} = \boldsymbol{\eta}$, $\boldsymbol{\chi} = -\boldsymbol{\xi}$. Then (4) becomes

$$u_t = \frac{u_{\boldsymbol{\xi\xi}}}{1+|\nabla f|^2} + \frac{2\,|\nabla f|\,|\nabla u|\, f_{\boldsymbol{\xi\eta}}}{1+|\nabla f|^2} \tag{20}$$

where the last summand is by a factor $\left(1+|\nabla f|^2\right)$ larger than in the corresponding geodesic active contour evolution. This means that attraction of the contour in u towards high-contrast regions in f is strengthened, see Figure 2(c).

In summary, our findings indicate that compared to geodesic active contours (1) the amoeba active contour equation (4) tends to attract the contour u faster to high-contrast

image regions and to strengthen the alignment of level lines of u to those of f. These effects are in line with the somewhat finer adaptation of amoeba active contours to structure details that was observed in [18].

5 Pre-smoothing and Amoeba Filters

The approximation result of [19], compare Corollary 1, refers to the self-snakes PDE (1) with $f \equiv u/\lambda$. As pointed out in the introduction, a disadvantage of this PDE is its ill-posedness that is often countered by pre-smoothing, i.e.

$$u_t = |\boldsymbol{\nabla} u| \operatorname{div}\left(g(|\boldsymbol{\nabla} u_\sigma|^2) \frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|} \right) \tag{21}$$

with $u_\sigma = K_\sigma * u$.

This procedure can be translated in a straightforward way to our amoeba median filter setting. One only needs to carve the amoeba structuring elements based on the pre-smoothed image u_σ instead of u. The resulting filtering step is described by our amoeba active contour model with $f \equiv u_\sigma/\lambda$, such that the approximation result from Theorem 1 applies. Analogous to our discussion in the amoeba active contour setting this means that in the limit $\varrho \to 0$ not exactly (21) is approximated but a self-snakes equation with a modified pre-smoothing.

However, in practical computation of amoeba filters one always uses a positive amoeba radius ϱ. This means that such a filtering procedure with amoebas derived from $f = u_\sigma/\lambda$ would contain two spatial scale parameters, σ and ϱ, both of which act as some sort of spatial averaging.

It can therefore be conjectured that the amoeba radius itself acts similarly as a pre-smoothing step. While a more exhausting investigation of this issue has to be left for future work, we compare pre-smoothed self-snakes with $g(s^2) = 1/(1+s^2)$ to amoeba median filtering ($f = u$) with positive amoeba radius for a very simple example.

Test Case. We consider the function $u : \mathbb{R}^2 \to \mathbb{R}$ given by

$$u(x, y) = x + \varepsilon \cos(kx) , \qquad \varepsilon \ll 1 . \tag{22}$$

It is composed of a simple linear slope (which would be stationary under each of the filters) and single-frequency perturbations of small amplitude. We will analyse the response of filters to that perturbation, dependent on the frequency parameter k.

Given the nonlinearity of the filters in question, there is no superposition property for these perturbations. Nevertheless, sufficiently small perturbations will interact with each other only in higher order terms $\mathcal{O}(\varepsilon^2)$, such that the technique will still give some intuition of the behaviour of the filters.

The chosen setting is representative of the practically meaningful situation of staircasing arising in a smooth transition.

Self-snakes. In our test case all level lines are parallel, so the 2D self-snakes equation simplifies to $u_t = g\,|\boldsymbol{\nabla} u| \operatorname{div}\big(\boldsymbol{\nabla} u/|\boldsymbol{\nabla} u|\big) + \langle \boldsymbol{\nabla} g, \boldsymbol{\nabla} u \rangle$. The first summand vanishes, while the second one simplifies to $g_x u_x$. From (22) we obtain $u_x = 1 - \varepsilon\, k\, \sin(kx)$, and with g as given by (2) further $g_x = \frac{1}{2}\,\varepsilon\, k^2\, \cos(kx) + \mathcal{O}(\varepsilon^2)$. Thus, we have

$$u_t = g_x u_x = \frac{\varepsilon\, k^2}{2}\, \cos(kx) + \mathcal{O}(\varepsilon^2) \tag{23}$$

indicating an indefinite amplification of higher frequencies. At the same time, the higher-order terms resulting from nonlinearity lead to an instantaneous propagation of the perturbation from a given frequency k to higher frequencies, which means that even for a single-frequency perturbation arbitrarily high frequencies with arbitrarily high amplification ratios will appear within short evolution time, enabling a loss of regularity of the evolving function.

Pre-smoothing. Replacing $g \equiv g(|\boldsymbol{\nabla} u|^2)$ with $g_\sigma \equiv g(|\boldsymbol{\nabla} u_\sigma|^2)$, we have in our test case $u_\sigma = x + \varepsilon\, \mathrm{e}^{-k^2\sigma^2/2} \cos(kx)$, thus $\partial_x g_\sigma = \frac{\varepsilon\, k^2}{2}\, \mathrm{e}^{-k^2\sigma^2/2} \cos(kx) + \mathcal{O}(\varepsilon^2)$ and finally

$$u_t = \partial_x g_\sigma \cdot \partial_x u = \frac{\varepsilon\, k^2}{2}\, \mathrm{e}^{-k^2\sigma^2/2} \cos(kx) + \mathcal{O}(\varepsilon^2)\ . \tag{24}$$

Unlike before, the amplification ratio $k^2 \exp(-k^2\sigma^2/2)$ is bounded and reaches a maximum for $k = \sqrt{2}/\sigma$, such that regularity of the evolving function is kept.

Amoeba Filter with Finite-Size Radius. We calculate the effect of amoeba median filtering with amoeba radius ϱ on our test case in the same way as in the proof of Theorem 1 via the area difference $\Delta := |\mathcal{A}_1| - |\mathcal{A}_2|$. As in our test settings level lines are not curved, only the asymmetry contribution Δ_1 needs to be considered.

The amoeba around $\boldsymbol{x}_0 = (x_0, y_0)$ is symmetric with respect to the line $y = y_0$ (parallel to the x-axis). We parametrise this symmetry line as $(x(s), y_0)$, where s is an arc-length parameter in the amoeba metric, i.e.

$$\int_0^{x(s)} \sqrt{1 + u_x^2(z, y_0)}\, \mathrm{d}z = s\ . \tag{25}$$

From the level line through $(x(s), y_0)$ (parallel to the y-axis), the amoeba cuts out a piece of length $2\sqrt{\varrho^2 - s^2}$. The sought area difference is therefore

$$\begin{aligned}\Delta(\boldsymbol{x}_0) &= \int_0^{\varrho} \frac{2\sqrt{\varrho^2 - s^2}}{\sqrt{1 + u_x^2(x(s), y_0)}}\, \mathrm{d}s - \int_{-\varrho}^{0} \frac{2\sqrt{\varrho^2 - s^2}}{\sqrt{1 + u_x^2(x(s), y_0)}}\, \mathrm{d}s \\ &= 2 \int_0^{\varrho} \sqrt{\varrho^2 - s^2} \left(\frac{1}{\sqrt{1 + u_x^2(x(s), y_0)}} - \frac{1}{\sqrt{1 + u_x^2(x(-s), y_0)}} \right) \mathrm{d}s \end{aligned} \tag{26}$$

with $u_x(x, y) = 1 - k\,\varepsilon\, \sin(k\,x)$.

Analogously to (15), the resulting median is $u(\boldsymbol{x}_0) + \Delta(\boldsymbol{x}_0)/(4\,\varrho)$. Numerical integration of (26) confirms that $\Delta(x, y_0)$ itself is approximately a multiple of the perturbation function $\varepsilon\, \cos(k\,x)$. For easy comparison with (24), we divide the amplification

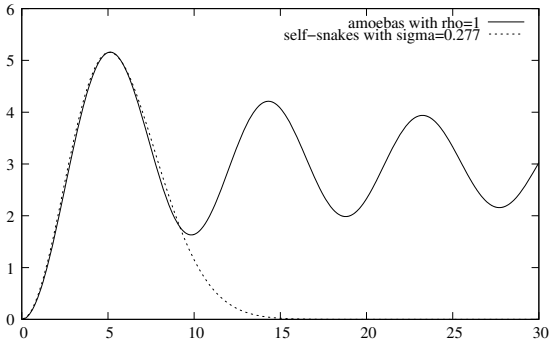


Fig. 3. Comparison of amplification factors depending on the frequency parameter k for pre-smoothed self-snakes and amoeba median filtering with fixed amoeba size. Horizontal axis shows k, vertical axis shows amplification factors.

factor $\Delta(x, y_0)/(4\,\varrho\,\cos(k\,x))$ by $\varrho^2/6$ (the evolution time corresponding to amoeba radius ϱ in the asymptotic approximation results).

Figure 3 shows the numerically computed factor $\Delta(x, y_0)/(4\,\varrho\,\cos(k\,x)) \cdot 6/\varrho^2$ along with the factor $k^2\exp(-k^2\sigma^2/2)/2$ from (24) as functions of the frequency parameter k. Here, ϱ and σ were chosen for an optimal fit of the first maximum. It is evident that the first lobe of the amplification functions is very similar. For higher frequencies the exponential dampening of the pre-smoothed self-snakes is superior to the oscillations of the amoeba amplification factor around a positive value. However, when practically filtering images, higher frequencies are cut off by spatial discretisation anyway. If the amoeba radius is not larger than approx. $10/\pi \approx 3$, the higher lobes of the amplification function in Figure 3 will disappear entirely.

6 Conclusion

We have analysed our amoeba active contour method proposed in [18] and derived a partial differential equation that it approximates asymptotically for vanishing structuring element size. Our result reproduces as special cases two earlier results from literature: the approximation of geodesic active contours in a special case [18] and the approximation of self-snakes by iterated amoeba median filtering [19]. In the general case, the PDE derived here differs from the geodesic active contour equation. The implications of the differences for active contour segmentation have been discussed and found to be consistent with the experimental findings of [18].

Finally, we have discussed from the same view point the approximation of self-snakes with pre-smoothing by amoeba filters. As a first step towards a more comprehensive investigation of the relation between curvature-based PDEs with pre-smoothing, and amoeba filtering with non-vanishing structuring elements, we have compared the effect of both methods in a simple special case with single-frequency perturbations of a constant gradient image. Future work extending this analysis is expected to lead to a deeper understanding of the interplay between adaptive morphology and PDE methods.

References

1. Alvarez, L., Lions, P.L., Morel, J.M.: Image selective smoothing and edge detection by nonlinear diffusion. II. SIAM Journal on Numerical Analysis 29, 845–866 (1992)
2. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics and Image Processing 34, 344–371 (1986)
3. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. Fifth International Conference on Computer Vision, pp. 694–699. IEEE Computer Society Press, Cambridge (1995)
4. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22, 61–79 (1997)
5. Feddern, C., Weickert, J., Burgeth, B., Welk, M.: Curvature-driven PDE methods for matrix-valued images. International Journal of Computer Vision 69(1), 91–103 (2006)
6. Guichard, F., Morel, J.M.: Partial differential equations and image iterative filtering. In: Duff, I.S., Watson, G.A. (eds.) The State of the Art in Numerical Analysis. IMA Conference Series (New Series), vol. 63, pp. 525–562. Clarendon Press, Oxford (1997)
7. Ikonen, L., Toivanen, P.: Shortest routes on varying height surfaces using gray-level distance transforms. Image and Vision Computing 23(2), 133–141 (2005)
8. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Gradient flows and geometric active contour models. In: Proc. Fifth International Conference on Computer Vision, pp. 810–815. IEEE Computer Society Press, Cambridge (1995)
9. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Conformal curvature flows: from phase transitions to active vision. Archives for Rational Mechanics and Analysis 134, 275–301 (1996)
10. Kimmel, R.: Fast edge integration. In: Osher, S., Paragios, N. (eds.) Geometric Level Set Methods in Imaging, Vision and Graphics, pp. 59–77. Springer, New York (2003)
11. Lerallut, R., Decencière, E., Meyer, F.: Image processing using morphological amoebas. In: Ronse, C., Najman, L., Decencière, E. (eds.) Mathematical Morphology: 40 Years On. Computational Imaging and Vision, vol. 30. Springer, Dordrecht (2005)
12. Lerallut, R., Decencière, E., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25(4), 395–404 (2007)
13. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. Journal of Computational Physics 79, 12–49 (1988)
14. Osher, S., Rudin, L.I.: Feature-oriented image enhancement using shock filters. SIAM Journal on Numerical Analysis 27, 919–940 (1990)
15. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 629–639 (1990)
16. Sapiro, G.: Vector (self) snakes: a geometric framework for color, texture and multiscale image segmentation. In: Proc. 1996 IEEE International Conference on Image Processing, Lausanne, Switzerland, vol. 1, pp. 817–820 (September 1996)
17. Tukey, J.W.: Exploratory Data Analysis. Addison–Wesley, Menlo Park (1971)
18. Welk, M.: Amoeba active contours. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 374–385. Springer, Heidelberg (2012)
19. Welk, M., Breuß, M., Vogel, O.: Morphological amoebas are self-snakes. Journal of Mathematical Imaging and Vision 39, 87–99 (2011)
20. You, Y.L., Kaveh, M., Xu, W., Tannenbaum, A.: Analysis and design of anisotropic diffusion for image processing. In: Proc. 1994 IEEE International Conference on Image Processing, Austin, Texas, USA, vol. 2, pp. 497–501 (November 1994)

Scale and Edge Detection with Topological Derivatives

Guozhi Dong[1], Markus Grasmair[1,2], Sung Ha Kang[3], and Otmar Scherzer[1,4]

[1] Computational Science Center,
University of Vienna,
Nordbergstrasse 15,
1090 Wien, Austria
{guozhi.dong,markus.grasmair,otmar.scherzer}@univie.ac.at
[2] Catholic University Eichstätt–Ingolstadt,
Ostenstrasse 26,
85072 Eichstätt, Germany
[3] School of Mathematics,
Georgia Institute of Technology,
686 Cherry Street NW
Atlanta, GA 30332-0160, USA
kang@math.gatech.edu
[4] Johann Radon Institute for Computational and
Applied Mathematics (RICAM),
Austrian Academy of Sciences,
Altenbergerstrasse 69, A-4040 Linz, Austria

Abstract. A typical task of image segmentation is to partition a given image into regions of homogeneous property. In this paper we focus on the problem of further detecting scales of discontinuities of the image. The approach uses a recently developed iterative numerical algorithm for minimizing the Mumford-Shah functional which is based on topological derivatives. For the scale selection we use a squared norm of the gradient at edge points. During the iteration progress, the square norm, as a function varied with iteration numbers, provides information about different scales of the discontinuity sets. For realistic image data, the graph of the norm function is regularized by using total variation minimization to provide stable separation. We present the details of the algorithm and document various numerical experiments.

Keywords: Mumford-Shah Functional, Topological Derivatives, Scale Selection, Total Variational Filtering.

1 Introduction

One of the most well-studied image segmentation model is the Mumford–Shah functional [16], which is to find the image u which minimizes the following:

$$F(u,K) = \frac{1}{2}\int_{\Omega}(u-f)^2dx + \frac{\alpha}{2}\int_{\Omega\setminus K}|\nabla u|^2dx + \beta\mathcal{H}^1(K) \tag{1.1}$$

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 404–415, 2013.

over all sets $K \subset \Omega$ and all smooth functions u defined on $\Omega \backslash K$. The first component provides a piecewise smooth approximation of the given image data $f\colon \Omega \to [0,\infty)$. The second component provides the information on the discontinuity set of the image f. Here $\mathcal{H}^s(K)$ denotes the s-dimensional Hausdorff-measure of the set K. We focus on the two-dimensional case $\Omega \subset \mathbb{R}^2$.

There are number of methods proposed to minimize the Mumford-Shah functional. One of the most important approach is by Ambrosio and Tortorelli [1,4]. The general idea is to approximate the functional by a family of elliptic functionals, where each of them in principle can be minimized with numerical partial differential equation solvers. Related works include the Chan–Vese model [7], where the Mumford–Shah model is simplified to the reconstruction of piecewise constant functions only, and the discontinuity sets are eliminated using an explicit notion of boundary via a level set formulation or using well-potential models (see [13,21]).

Recently, a numerical algorithm for minimizing the Mumford–Shah functional based on topological derivatives has been developed [12]. The implementation of the algorithm is iterative in nature and selects edges successively according to certain rules. In this paper, we further experimentally analyse these criteria. We show that the algorithm based on topological derivatives can distinguish between edges of different scales and therefore can be used for detecting scales of edges. This is different from [1,4] where the global approximation of the Mumford–Shah functional is achieved by partial differential equations, which, however, does not allow a selective selection of edges.

The approach of [12] consists of approximating the Mumford–Shah functional by the family of functionals

$$\begin{aligned} J_{\varepsilon,\kappa}(u,K) &= G_{\varepsilon,\kappa}(u,K) + 2\beta\varepsilon m_\varepsilon(K) \\ &= \frac{1}{2}\int_\Omega (u-f)^2 dx + \frac{\alpha}{2}\int_{\Omega\backslash K} |\nabla u|^2\, dx + \kappa\frac{\alpha}{2}\int_{K\cap\Omega} |\nabla u|^2\, dx + 2\beta\varepsilon m_\varepsilon(K)\,, \end{aligned} \tag{1.2}$$

where

$$m_\varepsilon(K) = \inf\{\mathcal{H}^0(Y) : Y \subset \mathbb{R}^2,\ K = \bigcup_{y\in Y} B_\varepsilon(y)\}.$$

The minimization is performed over all $u \in H^1(\Omega)$ and $K \subset \mathbb{R}^2$. It has been shown in [12] that these functionals Γ-converge to F, if $\kappa = o(\varepsilon)$. For fixed ε and κ the approximate minimization of the functional $J_{\varepsilon,\kappa}$ is performed by using a *topological asymptotic analysis* (see [9,10,22]). In the context of image processing, topological asymptotic analysis has been recently applied by Auroux et al. [2,3] and by Muszkieta [17]. In [12], an implementation for minimizing $J_{\varepsilon,\kappa}$ is proposed, and compared to the Ambrosio–Tortorelli approach [1].

The outline of this paper is as follows: In the following section, we recapitulate the algorithm from [12] for approximate minimization of the Mumford–Shah functional. We present a simple example where we can explain the idea of scales of edges in Section 3. Section 4 considers scale detection for realistic image

data. In the later cases, total variation regularization of the according scale detection functions over the number of iterations has to be performed to be able to calculate the according edges. We use the taut string algorithm for computing the total variation minimizers.

2 A Topological Algorithm for Edge Detection

We shortly review the algorithm from [12] for detecting edges in an image $f\colon \Omega \to \mathbb{R}$. In this iterative algorithm, the edges are approximated by a collection of balls of small diameter 2ε (in implementations, the diameter is chosen as the pixel distance). In each iteration, we smooth the original image using a diffusivity which is small at the (previously found) edge set and large outside this set. Then we add the new balls where the gradient norm is largest to the edge set. A detailed outline is given in Algorithm 1.

Algorithm 1. Topological algorithm for edge detection.

Let $f \in L^\infty(\Omega)$, α, $\beta > 0$, $\varepsilon > 0$ and $0 < \kappa < 1$ be given. Set $k = 0$ and $K_0 := \emptyset$.

Step 1. Define

$$u_k := \arg\min_u G_{\varepsilon,\kappa}(u, K_k).$$

Step 2. For $i = 1, \ldots, m$, find $y_k^{(i)} \in \Omega \setminus K_k$ such that $|\nabla u_k(y)|^2$ is maximal, and replace K_k by $K_k \cup \{B_\epsilon(y_k^{(i)})\}$.

Step 3. If

$$\max_i \frac{\alpha}{2}\pi\frac{1-\kappa}{1+\kappa}|\nabla u_k(y_k^{(i)})|^2 < \frac{\beta}{\varepsilon},$$

stop the iteration; else set $K_{k+1} := K_k$, $u := u_k$, increase k by 1, and go to Step 1.

Result: Approximation of an optimal edge set K and smoothed image u for the Mumford–Shah functional with parameters α and β.

It is shown in [12] that the resulting set K and the smoothed image u can be considered approximations of the minimizer of the Mumford–Shah functional.

Remark 1. The parameters α and β in Algorithm 1 are identical with the parameters in the Mumford–Shah functional. For noisy images, they should be chosen in dependence of the noise level: the larger the parameters are the smoother the filtered results are and the smaller the edge sets are. In the numerical experiments, we used α in the range from 5 to 10 and β between 100 and 200, and the size of the images are 256×256, of which the intensities range from 0 to 255.

In the numerical implementations, the parameter ε is always chosen as half the distance between adjacent pixels. According to [12], the parameter κ should be chosen as $o(\varepsilon)$. In our implementations we have set κ equal to 0.005. In general, the results proved quite robust with respect to the variations of κ, except for the optimization problem $G_{\varepsilon,\kappa}(u, K) \to \min$ in Step 2 of the algorithm, which becomes more difficult to solve as κ decreases.

The original algorithm from [12] uses an update of the function u whenever a ball has been added to K; this corresponds to setting $m = 1$ in Algorithm 1. For larger images, this is obviously not feasible. In this paper, we add multiple balls during each iteration, slightly compromising the accuracy in favour of vastly improved computation times.

3 Detecting Scale of Discontinuities

We present how the iterative construction of the edge set K in Algorithm 1 can be applied to detecting the edges of different *scales*. To highlight the idea, we consider the test image depicted in Figure 1, which consists of different flat regions that appear well separable. For this piecewise constant example the different scales of edges correspond to edges of a certain magnitude.

We first define the following norm to distinguish between different scales of the edges:

$$S(k) := |\nabla u_k(y_k^{(1)})|^2 \,. \tag{3.1}$$

This is the squared norm of the gradient of u_k at the center point of the first ball detected in each iteration, i.e., the largest gradient outside of the edge set at the previous iteration.

Figure 1 shows the results of Algorithm 1 for a fixed set of parameters. The function S shows some discriminative features: there are intervals where the values are approximately constant. This reflects that the edges recovered during these iterations are of a similar magnitude. Moreover, there are clear discontinuities (clear drops in height), which reflects that all the edges of a certain scale have been completely detected.

This change in the edge jump is illustrated in the lower images in Figure 1. They present the current edge indicators K_k at steps k of the iteration where the most significant jumps in S occur. In addition, we have chosen the depicted jumps sufficiently far from each other so that the differences between subsequent edge indicators are not too small. The first jump in S appears after approximately 40 iterations. The corresponding edge indicator K_{40} indicates the upper left edges of the image, which have the largest absolute value of the gradient. The next significant jump of the function S occurs at iteration 74 — here we extract the full shape of the spade; the surface of the clubs is fully recovered at iteration 124 and the diamond comes out last around the 182 iterations with a very slight drop of S as the last obviously detectable scale.

4 Regularization of $S(k)$

The Cards image consists of large piecewise constant regions and the edges betweens these regions are clearly pronounced, and different scales of edges are easily identified. For natural images, the scales of edges are less pronounced, and the function S shows a less regular behaviour. This effect worsens if the data are noisy. We experimented with the *Cameraman* and the *Peppers* image data (see

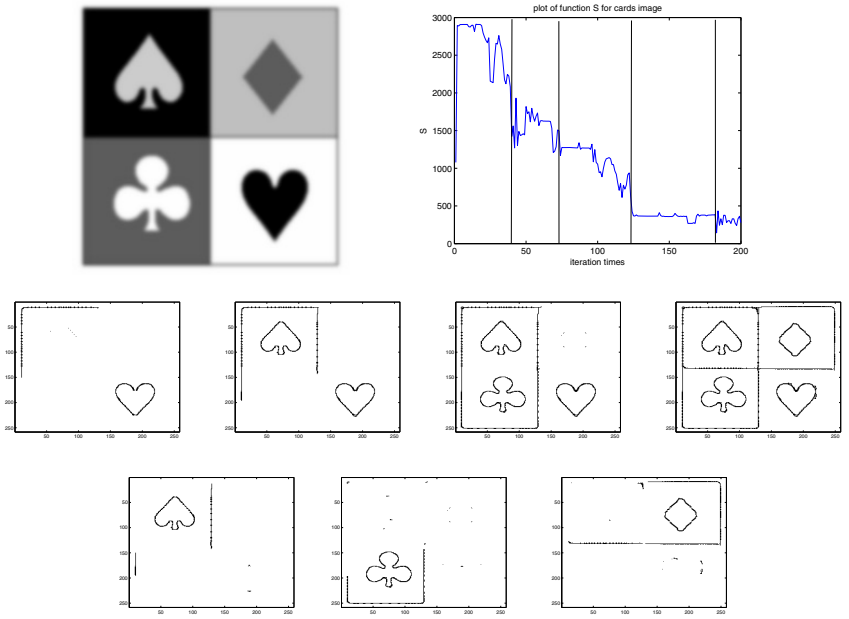


Fig. 1. Cards pictures. We apply Algorithm 1 with parameters $\alpha = 5$, $\beta = 100$, and $m = 20$. *Upper row, left:* The *cards* image. *Upper row, right:* The graph of the function S, one can discern distinct jumps of this function. *Middle row:* The edge set K_k at the iterations k where the most important jumps occur (from left to right: iterations 40, 74, 124, and 182). *Lower row:* The difference between two neighbouring edge sets.

Figure 2), and the functions S do not reveal similar obvious plateaus as for the Cards image.

To better deal with these natural images, we smooth the function S (considered as a function of iterates) by minimizing the discrete total variation, setting

$$\hat{S}_\lambda := \arg\min_R \left(\sum_k (R(k) - S(k))^2 + \lambda |R(k+1) - R(k)| \right).$$

This optimization problem has been studied for a long time in the contexts of one-dimensional signal processing and non-parametric regression (see for instance [8,15,23]). There are numerous methods for solving this minimization problem, most of which deal specifically with total variation regularization for image denoising (see for instance [5,6,19]). In the one-dimensional case, the most efficient method for total variation minimization is the *taut string algorithm* [8,11,18], which can be implemented in the form of a dynamical programming algorithm with linear time and space complexity. A detailed derivation of this method can be found in [11,20]. The dynamical programming algorithm for the solution is described in [8]. For this algorithm recall that a one-dimensional function of bounded variation is continuous outside its jump set, and that a function $U \in W^{1,1}(\Omega)$ is continuous.

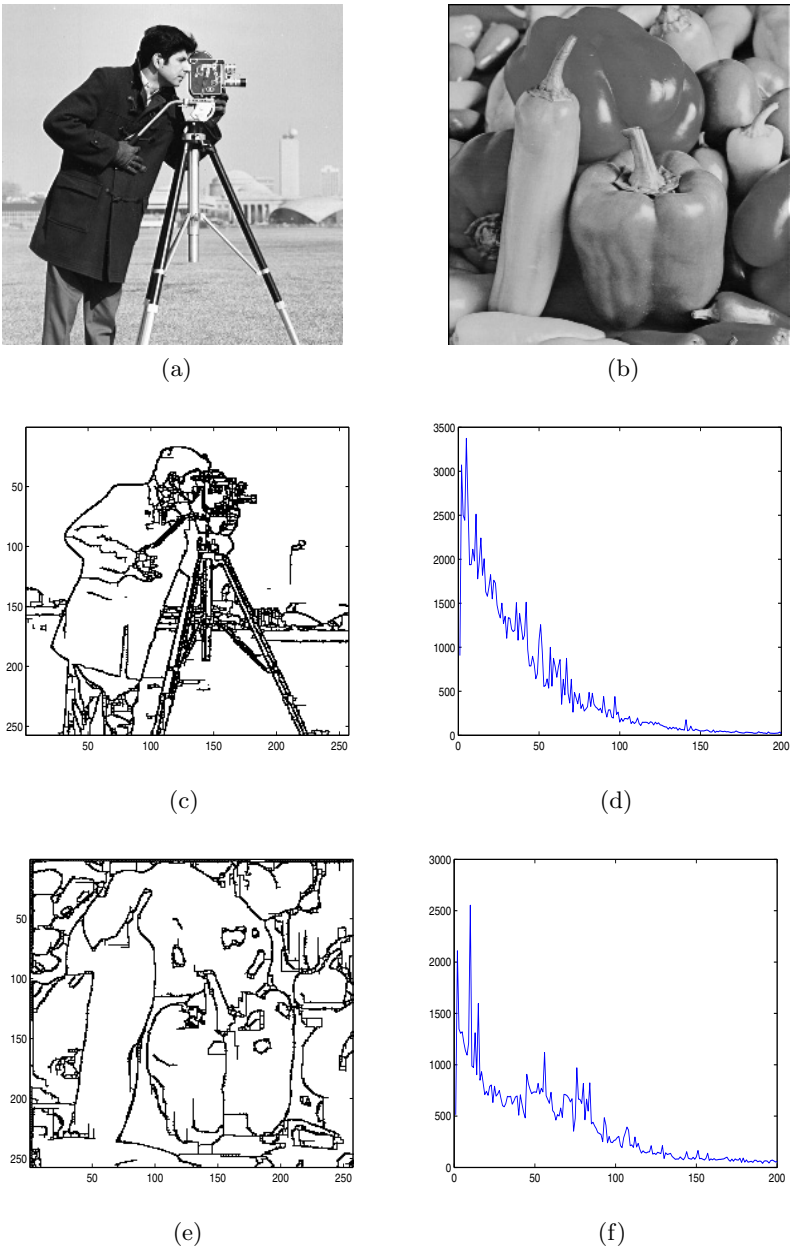


Fig. 2. Edge sets and scale detection function S for the Cameraman and Peppers images. (c) and (e) are the edges of the Cameraman and Peppers images, respectively, detected by our algorithm. In both cases the algorithm has been used with parameters $\alpha = 10$, $\beta = 200$, and $m = 50$. (d) and (f) are the graphs of the functions S for the Cameraman and Peppers images, respectively, where the horizontal axis represents the iteration numbers and the longitudinal axis is the maximum norm of gradients for the center of added balls in that iterations.

Algorithm 2. Taut String Algorithm

Given discrete data $\boldsymbol{u}^\delta = (f_i)$, $i = 1, \ldots, s$, and $\lambda > 0$, the taut string algorithm is defined as follows:
Step 1. Let $U_0^\delta = 0$ and $U_i^\delta = \frac{1}{s}\sum_{j=1}^{i} f_j$, $i = 1, \ldots, s$. We denote by $U^\delta(x)$ the linear spline with nodal points $x_i = i/s$, $i = 0, \ldots, s$, and function values U_i^δ at x_i.
Step 2. Define the λ-*tube*

$$\mathcal{Y}_\lambda := \big\{U \in W^{1,1}(0,1) : U(0) = U^\delta(0)\,,\ U(1) = U^\delta(1)\,, \\ \text{and } |U(t) - U^\delta(t)| \le \lambda \text{ for } t \in (0,1)\big\}\,.$$

Step 3. We calculate the function $U_\lambda \in \mathcal{Y}_\lambda$ which minimizes the graph length, that is,

$$U_\lambda = \operatorname{argmin}_{U \in \mathcal{Y}_\lambda} \int_0^1 \sqrt{1 + (U')^2}\,.$$

Step 4. $u_\lambda := U_\lambda'$ is the outcome of the taut string algorithm.

In this approach, the amount of regularization depends on the parameter $\lambda > 0$. However, due to the properties of one-dimensional total variation regularization, the results are quite stable with respect to λ. We recall that one-dimensional total variation regularization satisfies a semi-group property: Repeated regularization first with a parameter $\lambda_1 > 0$ and then with a parameter $\lambda_2 > 0$ is the same as a single regularization step with a parameter $\lambda_1 + \lambda_2$ (see e.g. [20, Thm. 4.38]). In particular, this implies that for $\mu > \lambda$, the jump set of $\hat{S}_\mu$ is contained in the jump set of $\hat{S}_\lambda$. For this reason, the precise choice of the regularization parameter has, for modest values, no effect on the location of the most prominent jumps of $\hat{S}_\lambda$. In our implementations, we therefore chose the regularization parameter experimentally, using a sufficiently large parameter in order to remove most of the noise, but still retaining all the significant jumps.

Another possibility is to choose the smallest parameter λ for which the function $\hat{S}_\lambda$ is monotonically decreasing, since it is sufficient to find points where the scale of edges suddenly decreases.

Finally, we present some experimental results with Cameraman and Peppers image data, respectively in Figure 3 and Figure 4. To obtain these results, we first applied the *taut string algorithm* to filter the plots of the oscillating function S defined in (3.1). The plateaus between two discontinuities in the filtered function $\hat{S}_\lambda$ mark areas of a specific scale. Actually, there are more jumps detected than which we are separately displaying, as well shown in the filtered graphs with the figures, for the purpose of emphasising the scales detected by Algorithm 1, we just specify the most obvious jumps and figure them out part by part.

See Figure 3 and Figure 4, respectively. In Figure 3, the first discontinuity of S appears at iteration 17. There the most contrasty edges of the image — the boundary of the hair and black coat of the photographer — are almost developed. The next significant jump appears at iteration 23, where the outline

of the photographer has almost been formed. Then, at iteration 43, the structure of the cameraman has been caught. Finally, at iteration 64, also smaller details like the the camera are segmented.

In Figure 4 the situation is slightly different. The data contains many peppers where the contrast of the edges is rather similar. Thus the algorithm does not select complete pepper components but only parts of them on different peppers and associates them to a unique scale.

As a final example we apply the algorithm to the Cameraman, superimposed with Gaussian noise (see Figure 5). The results show the robustness of the method. Because of the noise, we used a larger regularization parameter in the taut string algorithm in order to select the different scales. Note that it was not necessary to change the regularization parameters α and β.

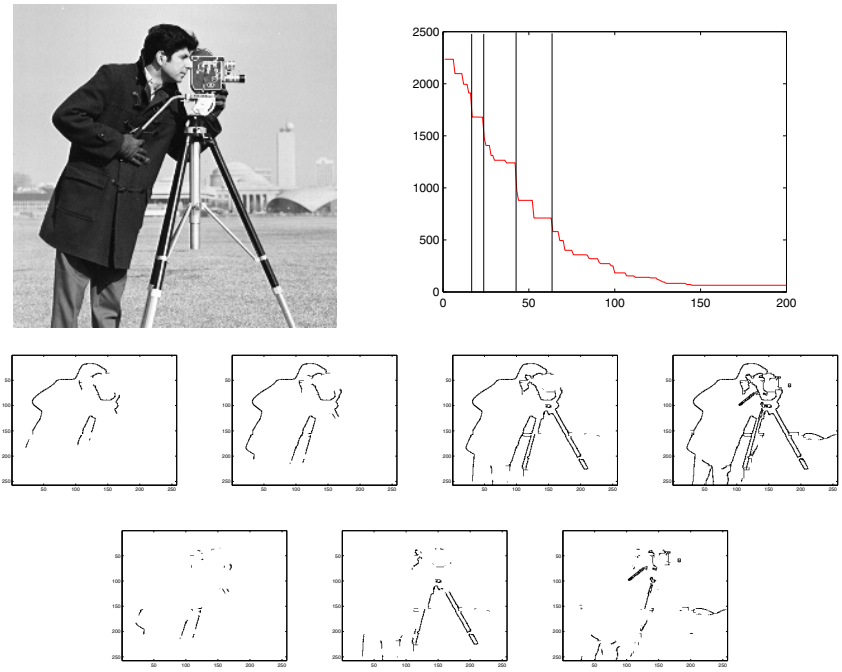


Fig. 3. Edge and scale detection in the Cameraman image. *Upper row, left:* The Cameraman image. *Upper row, right:* The TV filtered graph of function S. *Middle row:* The edge set K_k at the iterations k where the most prominent jumps occur (from left to right: iterations 17, 23, 43, and 64). *Lower row:* The difference between two subsequent edge sets.

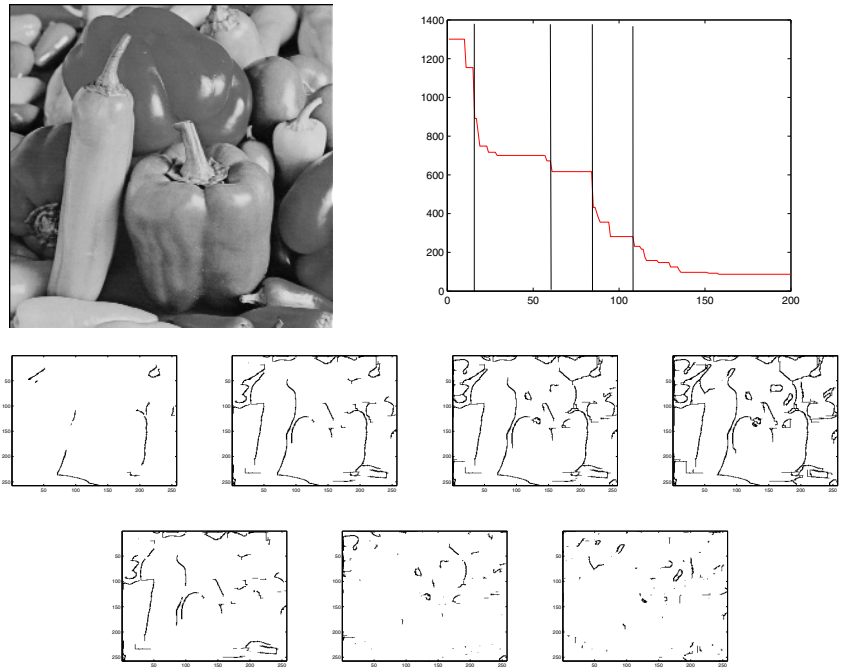


Fig. 4. Edge and scale detection of Peppers image. *Upper row, left:* The Peppers image. *Upper row, right:* The TV filtered graph of function S. *Lower row:* The edge set K_k at the iterations k where the most prominent jumps occur (from left to right: iterations 15, 60, 85, and 108). *Lower row:* The difference between two subsequent edge sets.

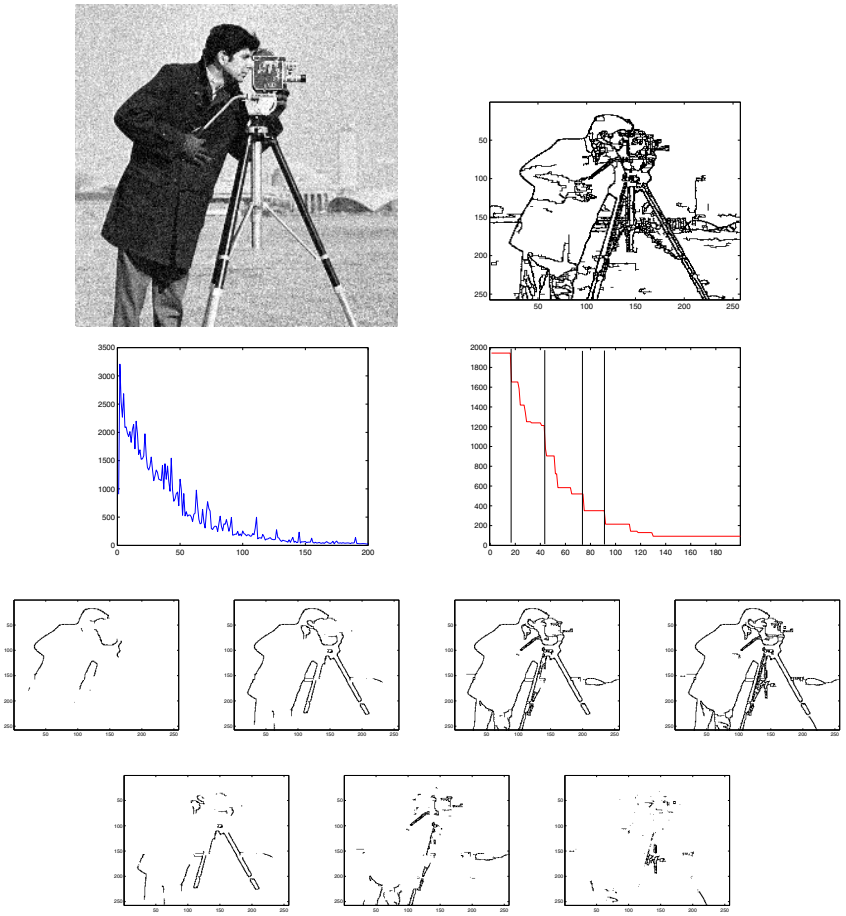


Fig. 5. Edge and scale detection in the presence of noise. *Upper row, left:* The noisy Cameraman image. *Upper row, right:* The edge set after 200 iterations. *Middle upper row, left:* The graph of the function S. *Middle upper row, right:* TV filtered graph of function S. *Middle lower row:* The edge set K_k at the iterations k where the most prominent jumps occur (from left to right: iterations 16, 43, 74, and 91). *Lower row:* The difference between two subsequent edge sets.

5 Conclusion

We presented a method to detect different scales of jumps across the boundary, using a recently developed algorithm [12] for approximating the Mumford–Shah algorithm using topological derivatives. This process is possible due to the locality of the algorithm, contrasting the globality of approaches like the Ambrosio–Tortorelli approximation. By considering the function S which represents the scale change of the edges, one can distinguish different parts of boundaries with different levels of jumps. For realistic images (also noisy), we applied total variation minimization using the taut string algorithm for a more stable scale separation. For future works, different norms can be considered for the function S, and there are possible improvements by adapting particular image features to the function $\hat{S}_\lambda$, or either considering to replace the current discrete count k which is the variable of the function S by a continuous parameter, as well, the impact of the roughness of edges on the scales separation is worth to be discussion.

Acknowledgements. The work of GD and OS has been supported by the Austrian Science Fund (FWF) within the national research networks Photoacoustic Imaging in Biology and Medicine, project S10505 and Geometry and Simulation, project S11704. All the authors want to express their thanks to the anonymous reviewers for their comments to improve the paper.

References

1. Ambrosio, L., Tortorelli, V.M.: Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. Comm. Pure Appl. Math. 43(8), 999–1036 (1990)
2. Auroux, D., Belaid, L.J., Masmoudi, M.: A topological asymptotic analysis for the regularized grey-level image classification problem. Math. Model. Numer. Anal. 41(3) (2007)
3. Auroux, D., Masmoudi, M.: Image processing by topological asymptotic expansion. J. Math. Imaging Vision 33(2) (2009)
4. Chambolle, A.: Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. SIAM J. Appl. Math. 55(3), 827–863 (1995)
5. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vision 20(1-2), 89–97 (2004)
6. Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. Numer. Math. 76(2), 167–188 (1997)
7. Chan, T., Vese, L.: Active Contours without Edges. IEEE Trans. Image Processing 10(2), 266–277 (2001)
8. Davies, P.L., Kovac, A.: Local extremes, runs, strings and multiresolution. Ann. Statist. 29(1), 1–65 (2001)
9. Feijóo, R.A., Novotny, A., Padra, C., Taroco, E.: The topological derivative for the Poisson problem. Math. Mod. Meth. Appl. Sci. 13(12), 1825–1844 (2003)
10. Garreau, S., Guillaume, P., Masmoudi, M.: The topological asymptotic for PDE systems: The elasticity case. SIAM J. Control Optimiz. 39(6), 1756–1778 (2000)

11. Grasmair, M.: The equivalence of the taut string algorithm and BV-regularization. J. Math. Imaging Vision 27(1), 59–66 (2007)
12. Grasmair, M., Muszkieta, M., Scherzer, O.: An approach to the minimization of the Mumford–Shah functional using Γ-convergence and topological asymptotic expansion. Preprint on ArXiv, arXiv:1103.4722v1, University of Vienna, Austria (2011)
13. Jung, Y.M., Kang, S.H., Shen, J.: Multiphase Image Segmentation via Modica-Mortola Phase transitio. SIAM Applied Mathematics 67(5), 1213–1232 (2007)
14. Kimmel, R., Sochen, N.A., Weickert, J. (eds.): Scale Space and PDE Methods in Computer Vision. LNCS, vol. 3459. Springer, Heidelberg (2005)
15. Mammen, E., van de Geer, S.: Locally adaptive regression splines. Ann. Statist. 25(1), 387–413 (1997)
16. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. 42(5), 577–685 (1989)
17. Muszkieta, M.: Optimal edge detection by topological asymptotic analysis. Math. Methods Appl. Sci. 19(11), 2127–2143 (2009)
18. Pöschl, C., Scherzer, O.: Characterization of minimizers of convex regularization functionals. In: Frames and Operator Theory in Analysis and Signal Processing. Contemp. Math., vol. 451, pp. 219–248. Amer. Math. Soc., Providence (2008)
19. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D 60(1-4), 259–268 (1992)
20. Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational methods in imaging. In: Applied Mathematical Sciences, vol. 167. Springer, New York (2009)
21. Shen, J.: A stochastic-variational model for Soft Mumford-Shah segmentation. In: International Journal of Biomedical Imaging, ID92329 (2006)
22. Sokołowski, J., Żochowski, A.: On topological derivative in shape optimization. SIAM J. Control Optimiz 37(4), 1251–1272 (1999)
23. Steidl, G., Didas, S., Neumann, J.: Relations between higher order TV regularization and support vector regression. In: [14], pp. 515–527 (2005)

Active Contours for Multi-region Image Segmentation with a Single Level Set Function

Anastasia Dubrovina, Guy Rosman, and Ron Kimmel

Computer Science Department, Technion - IIT, Haifa, Israel
{nastyad,rosman,ron}@cs.technion.ac.il

Abstract. Segmenting the image into an arbitrary number of parts is at the core of image understanding. Many formulations of the task have been suggested over the years. Among these are axiomatic functionals, which are hard to implement and analyze, while graph-based alternatives impose a non-geometric metric on the problem.

We propose a novel approach to tackle the problem of multiple-region segmentation for an arbitrary number of regions. The proposed framework allows generic region appearance models while avoiding metrication errors. Updating the segmentation in this framework is done by level set evolution. Yet, unlike most existing methods, evolution is executed using a *single non-negative* level set function, through the Voronoi Implicit Interface Method for a multi-phase interface evolution. We apply the proposed framework to synthetic and real images, with various number of regions, and compare it to state-of-the-art image segmentation algorithms.

1 Introduction

Image segmentation plays an important role in object detection and classification, scene understanding, action classification, and other visual information analysis processes. In this paper we consider active contour approaches, which have been proven to be very successful for that goal. These include edge-based methods [15,5,18,6] , region-based techniques [21,8,10,13] , and combined approaches [37,24,28], to mention just a few.

Several approaches have been suggested for numerical computation of region boundaries. These include explicit spline evolution [15], level set evolution [23,5], graph-cuts [3,27,12], and continuous convex optimization [25,7]. Among these, the level set framework provides a significant amount of flexibility in the design of the segmentation criterion. While being naturally suitable for variable topology of the regions, this framework has been extended to accommodate different assumptions on the image and its structure. These include various appearance models [13,20,22,1], and different shape priors [16,11,26].

However, the level set framework is geared towards two-region image segmentation. To alleviate this limitation, various methods were developed; most of them require managing *multiple* level set functions. Some associate a level set function with each image region, and evolve these functions in a coupled manner

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 416–427, 2013.

[36,35,29]. Others perform hierarchical segmentation, by iteratively splitting previously obtained regions using the conventional level set framework [33,4]. These methods too require coupled level set evolution, so that the resulting regions do not develop gaps or overlaps. It is also possible to use a smaller number of level set functions, say n, and segment an image into 2^n regions [34]. Another approach was recently suggested in [17]. It uses a single level set function, similar to the proposed approach. However, when evolving the contour, it requires managing multiple auxiliary level set functions, so that no gaps/overlaps are created.

Other approaches to multi-region image segmentation either use a discrete labeling problem formulation and solve it using graph-cuts [27,12], or perform convex relaxation [25,7]. These methods are less easy to adapt for arbitrary segmentation functionals, in terms of both data and geometry priors. In addition, such approaches usually require knowing the number of regions a priori. Yet another method for image segmentation is by mean-shift clustering [10]. This approach does not, however, allow flexible choice of shape priors or arbitrary probability models.

We propose a new level set method for multiple region image segmentation. It overcomes previous challenges and allows segmenting images with arbitrary number of regions using various image appearance models. For this purpose we utilize a novel level set framework for multi-phase, or multi-region, interface evolution, named the Voronoi Implicit Interface Method (VIIM), which was introduced by Saye and Sethian in [30]. According to it, evolution is performed using a *single* non-negative level set function, while implicitly dealing with regions merging and splitting, and naturally handling arbitrary topological structures such as triple junctions.

Our main contributions can be summarized as follows: first, we review the axiomatic formulation of the multi-region image segmentation problem as an energy functional minimization. Specifically, we consider energy terms used in image segmentation based on region statistics, and extend them to the context of multiple regions. We then derive the active contour evolution equation minimizing the above energy functional, formulate it as a level set evolution problem, and solve it by utilizing the VIIM level set framework. The proposed approach does not require knowing the number of the regions in the image or their statistics a priori, and produces good segmentation results for various initial contours.

The structure of the paper is as follows: we begin by reviewing the Voronoi Implicit Interface Method, which is the numerical basis for our approach, in Section 2. In Section 3 we describe the main ideas that underlie the proposed method. We shortly review the multi-region segmentation model, for which we derive the corresponding level set evolution equation in terms of the VIIM framework, and describe prominent segmentation priors that fit within the suggested framework. In Section 4 we present segmentation results of the proposed approach, and compare it to state-of-the-art methods. Section 5 concludes the paper and describes potential extensions of the proposed framework.

2 Review of the Voronoi Implicit Interface Method

The VIIM was recently suggested for the solution of interface propagation problems with arbitrary number of phases, or regions, in m-dimensional Euclidean space. In 2D, the interface separating between different phases is a curve, possibly with multiple junctions. In 3D, the interface consists of two-dimensional surfaces. Illustrations of 2D and 3D interfaces can be found in [30].

The interface propagation is performed using a *single* non-negative level set function $\phi(\mathbf{x}), \mathbf{x} \in \mathbb{R}^m$, given by the unsigned distance from the interface Γ, and defined on a fixed regular grid. The propagation is governed by the equation

$$\phi_t = F_{ext} \left| \nabla \phi \right|, \tag{1}$$

where F_{ext} is the extension of the interface propagation speed F to the whole m-dimensional region. The examples in [30] include curvature and mean curvature flows, as well as physical simulations of the dynamics of dry foams.

The central idea of the VIIM is as follows: assume we are given a zero level set of a function ϕ, and a velocity F defined along it. We can extend this velocity to the neighboring level sets in a smooth manner, to obtain the extension velocity F_{ext} and apply Eq. (1). Then, two evolving ϵ-level sets will always encapsulate the evolving zero level set they are adjacent to. Moreover, the ϵ-level sets of ϕ are simple curves, without multiple-junction points, and their evolution is well defined. Thus, the evolved ϵ-level sets of the level set function can be used to reconstruct the evolving interface, which is assumed to lie at an equal distance from the two ϵ-level sets adjacent to it. It is calculated using the Voronoi regions of the ϵ-level sets.

In order to evolve the interface as described above, Saye and Sethian suggested the following three step-algorithm.

1. Evolve the level set function ϕ by solving Eq. (1).
2. Find the ϵ-level sets of the new function. Reconstruct the interface Γ to be the intersections of the Voronoi regions of the ϵ-level sets, where $\phi(\mathbf{x}) < \epsilon$. Update the level set function ϕ using the reconstructed interface Γ.
3. Update the propagation speed function F; return to 1.

The VIIM is formulated in terms of a general interface velocity F, and thus it is applicable to various interface evolution problems utilizing the level set approach. Below, we show how it can be employed for multiple regions image segmentation, where the active contour acts as an interface, and the regions it defines are the phases in the VIIM notation.

3 Multi-region Image Segmentation

A general energy functional describing an active contour model is given by

$$E(C) = E_{data}(C) + \mu E_{reg}(C). \tag{2}$$

The data term $E_{data}(C)$ is determined by the region-based image intensity model, for instance [21,8,13,20], etc. In this paper we demonstrate region-based terms that rely on two specific image models - the piecewise-constant model of [21,8] and a more general *Gaussian mixture model* (GMM). The regularization term $E_{reg}(C)$ is determined by the properties of the segmenting contour, and may depend on the contour alone [15,21], or incorporate image information as well [5,6]. The minimizing flow is derived from (2) using methods from calculus of variations, namely the active contour evolution is proportional to the first variation of the above energy functional.

3.1 Region-Competition Model with Geodesic Active Contours Regularization

Here, we consider a modified version of the region competition model of Zhu and Yuille [37], with added *geodesic active contour* (GAC) regularization term

$$E(C, \{\alpha_i\}) = \sum_i \iint_{\Omega_i} -\log P\left(I(x,y)|\alpha_i\right) dxdy + \mu \oint_C g(C(s))ds, \qquad (3)$$

where $I(x,y)$ is the image to be segmented, defined on a 2D domain Ω. The contour C divides the image domain into non-overlapping regions $\{\Omega_i\}_i$, such that $\Omega = \{\bigcup_i \Omega_i\} \bigcup C$. In the data term, $P\left(z|\alpha_i\right)$ is the probability distribution function of the image intensity values in region Ω_i, with corresponding parameters α_i. In the GAC term, $g(x,y)$ is the edge indicator function. Following [6], in this work we used $g(x,y) = \left(1 + \left|\nabla \hat{I}\right|^2\right)^{-1}$, where $\hat{I}$ is a smooth version of I. For color images we used $g(x,y)$ suggested in [28]: we treat the image as a 5-dimensional manifold $(x, y, \mathrm{R}(x,y), \mathrm{G}(x,y), \mathrm{B}(x,y))$ with metric $g_{\mu\nu}(x,y)$, so that the edge indicator function becomes $g(x,y) = \det\left(g_{\mu\nu}(x,y)\right)^{-1}$.

We perform alternating minimization: for a fixed contour C, for each region Ω_i we calculate the optimal parameters maximizing the image probability in that region

$$\alpha_i^* = \arg\max_{\alpha_i} \prod_{(x,y)\in\Omega_i} P\left(\alpha_i|I(x,y)\right), \quad \forall i. \qquad (4)$$

Then, for fixed region probability distribution parameters, the active contour evolution minimizing the energy $E(C, \{\alpha_i\})$ is given by

$$C_t = -\frac{\delta E}{\delta C} = \sum_{i\in N(x,y)} \log P\left(I|\alpha_i\right) \mathbf{n}_i + \mu\left(\kappa g - \langle \nabla g, \mathbf{n}\rangle\right)\mathbf{n}. \qquad (5)$$

For some $(x,y) \in C$, $N_i(x,y)$ denotes the set of indices of the regions Ω_j adjacent to C at (x,y). In each region, the normal $\mathbf{n}_i$ is defined such that it points outwards of the region Ω_i. The first term of the minimizing flow is obtained by differentiating the functional $E_{data}(C)$, as shown in [37]. The second term is the well known explicit geodesic active contour flow, obtained by differentiating the

regularization term in Eq. 3. The above evolution rule is well-defined for (x, y) lying on a contour segment defining a boundary between two regions Ω_i and Ω_j, for which $|N(x, y)| = 2$. We will denote such contour segments by C_{ij}.

The traditional methods, described in the introduction, require using multiple level set functions to perform the above evolution implicitly. In this work we suggest to exploit the advantages of the Voronoi implicit interface method for this purpose. The next section describes how to adapt the evolution rule Eq. (5) to be applicable within the VIIM framework. We would also like to note that the above formulation is general and may be applied for various models of image intensity probability distribution. In order to demonstrate this we apply the proposed method to two such models – Gaussian probability distribution with constant variance, leading to piecewise constant image segmentation functional [21,8], and a more elaborated Gaussian mixture model (GMM). Both models will be described in details in Section 3.3.

3.2 Contour Evolution Using the VIIM

In terms of the VIIM framework, the contour (interface) velocity $F(x, y)$ is well defined for points lying along a boundary between two regions, and is given by

$$F(x,y) = \left[\log P\left(I(x,y)|\alpha_i\right) - \log P\left(I(x,y)|\alpha_j\right)\right] + \mu\left(\kappa g - \langle \nabla g, \mathbf{n}_i\rangle\right), \quad (6)$$

in the direction $\mathbf{n}_i$, for $(x, y) \in C_{ij}$. According to the VIIM formulation, the contour velocity F needs to be extended to the neighboring level sets of the level set function $\phi(x, y)$, to create $F_{ext}(x, y)$. We observe that a straight forward extension of (6) produces a velocity profile with discontinuities at the boundaries of the Voronoi regions of different contour segments. This is also related to the fact that the interface velocity F is not well defined at the junction points.

Alternatively, we suggest to evolve the level sets of $\phi(x, y)$ in each region according to the local information of that region alone. Thus, the extension velocity, used to evolve the level set function according to Eq. (1), is defined by

$$F_{ext}(x,y) = \log P\left(I(x,y)|\alpha_i\right) + \mu \mathrm{div}\left(g(x,y)\frac{\nabla\phi}{|\nabla\phi|}\right), \quad (x,y) \in \Omega_i. \quad (7)$$

Proposition 1. *Assume that the level set function is given by an unsigned distance function from the evolving contour, and the parameters $\{\alpha_i\}$ are fixed. For $\epsilon \ll 1$, the VIIM framework with the extension velocity $F_{ext}(x, y)$ defined in Eq. (6) will move every regular point (x, y) on the contour in the direction of the velocity $F(C(x, y))\mathbf{n}_i$ (6) minimizing the energy functional $E(C)$ in Eq. (3).*

The suggested extension velocity F_{ext} (7) evolves the contour points along the same direction as $F(C(x, y))$ (if not by the same amount). Our experiments show that the suggested extension velocity produces valid segmentation results. Particularly, for the two-region piecewise constant problem, the results obtained with the proposed method are similar to those obtained using the original formulation of Chan-Vese [8]. Proof of Prop. 1 is given in the accompanying supplementary material.

The proposed approach can be summarized as follows: assume we are given an initial contour C_0 and the corresponding unsigned distance level set function $\phi(x, y)$.

1. Calculate extension velocity in each region using Eq. (7). Evolve the function $\phi(x, y)$ using the obtained velocity according to the evolution equation (1).
2. Extract the ϵ-level sets of the evolved level set function. Calculate the Voronoi regions of these ϵ-level sets in the narrow band $\{(x, y) : \phi(x, y) < \epsilon\}$, and reconstruct the evolved contour C as the collection of the boundaries between these Voronoi regions, as suggested by [30]. Perform re-distancing: re-calculate the unsigned level set function $\phi(x, y)$ using the new contour C.
3. Stop the evolution if a pre-defined stopping criterion was met; otherwise, return to Step 1.

3.3 Image Segmentation Models

Piecewise constant model: In this case we assume Gaussian probability distribution, given by $I \sim \mathcal{N}(c_i, \sigma_i^2)$ in region Ω_i. Further simplified by an assumption $\sigma_i = \sigma_j, \forall i, j$, the energy functional becomes

$$E_{MR}(C, \{c_i\}) = \sum_i \iint_{\Omega_i} (I(x, y) - c_i)^2 \, dxdy + \mu \oint_C g(C(s))ds. \tag{8}$$

The above is a modified version of the piecewise constant Mumford-Shah energy functional [21], in the sense that the regularization term is given by the geodesic active contours model (GAC). The contour C now separates multiple regions, denoted by Ω_i, and may have multiple-junction points. For $N = 2$ and $g = 1$, (8) is the well known Chan-Vese functional [8].

According to Equation (7), the extension velocity F_{ext} in the region Ω_i is given by

$$F_{ext}(x, y) = -(I(x, y) - c_i) + \mu \text{div}\left(g(x, y)\frac{\nabla\phi}{|\nabla\phi|}\right), \quad (x, y) \in \Omega_i. \tag{9}$$

For a given contour C, the optimal mean intensity values in each region, c_i^*, are given by

$$c_i^* = \frac{\iint_{\Omega_i} I(x, y)dxdy}{\iint_{\Omega_i} dxdy}. \tag{10}$$

Gaussian Mixture Model: Here, we model image intensity values in each region using the Gaussian mixture model [19], which have been successfully applied to various signal analysis tasks; specifically, in computer vision it was used for tracking [32], MR image segmentation [14], background substraction [38], etc. In GMM, the intensity probability distribution in region Ω_i is modeled by a weighted sum of m Gaussians, each with mean $c_i^{(j)}$ and covariance matrix $\sigma_i^{(j)} I$,

$$P(z|\alpha_i) = \sum_{j=1}^{m} \lambda_i^{(j)} \mathcal{N}\left(z \mid c_i^{(j)}, \sigma_i^{(j)} I\right), \tag{11}$$

where $\mathcal{N}\left(z \mid c_i^{(j)}, \sigma_i^{(j)} I\right)$ is the j^{th} component of the Gaussian mixture in the region Ω_i. The results shown in the next section were obtained with $m = 6$.

The extension velocity (7) becomes

$$F_{ext}(x,y) = \log\left[\sum_{j=1}^{m} \lambda_i^{(j)} \mathcal{N}\left(I(x,y)|c_i^{(j)}, \sigma_i^{(j)} I\right)\right] + \mu \text{div}\left(g(x,y)\frac{\nabla\phi}{|\nabla\phi|}\right), \quad (x,y) \in \Omega_i. \qquad (12)$$

The optimal model parameters α_i^*, where $\alpha_i = \left\{\lambda_i^{(j)}, c_i^{(j)}, \sigma_i^{(j)}\right\}_{j=1}^{m}$, are then calculated as suggested in Eq. (4), using an Expectation Maximization (EM) algorithm [19].

Finally, note that though the above problem formulation is given in terms of the image intensity values, other image representations can be easily utilized in the suggested framework, depending on a specific segmentation problem.

4 Experimental Results

In this section we present segmentation results obtained with the proposed method for different types of images, and compare them to the results obtained using the convex relaxation method of Chambolle and Pock [7]. In all our experiments, the image intensity values were normalized to the range $[0, 1]$. The algorithm parameters were $\mu \in [0.02, 0.1]$, the time step $dt = [25, 50]$, and $\epsilon = 0.1$. In order to prevent over-segmentation, we united separate regions with similar region statistics, as a part of Step 2 of the algorithm. For the piecewise constant model, we united regions with mean intensity value difference smaller than some threshold (if not stated explicitly, $T = 0.1$ was used).For color images, we used the maximal difference among the three color channels. For the GMM, we used the L_2-distance between sampled three-dimensional (for color images) probability distributions. The level set function evolution (1) was performed using the forward Euler scheme. To perform re-distancing and Voronoi region calculation we used the fast marching method [31], efficiently initialized as suggested in [9]. Both the ϵ-level set and the evolved contour extraction were performed with sub-pixel precision. It should be also noted that the width of the ϵ-level sets influences the size of the smallest feature that the algorithm is able to segment. To capture small features one may up-sample the image before the segmentation, similar to the technique used in [2].

It is important to note the computational efficiency of the proposed method. Typically, significant parts of the evolution can be performed in a narrow-band fashion. Specifically, the update of the piecewise-constant model, as well as the M-step of the EM estimation for the Gaussian mixture model can be performed incrementally, keeping the same complexity of the 2-region active contours scheme. The expectation step of the EM algorithm, however, requires computation over the entire image domain. Exploring efficient implementation aspects such as incremental update of the expectation is left for future work.

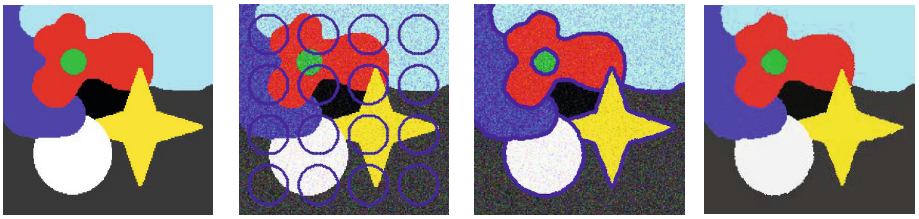


Fig. 1. Segmentation of noisy synthetic color image with overlapping objects. Left to right: the original image, noisy image with the initial contour, region boundaries obtained using the piecewise constant model, piecewise constant segmentation.

In our first experiment, we applied the algorithm with the piecewise constant model to a noisy synthetic image with several overlapping regions, with triple-junction boundary intersections. The segmentation result is shown in Fig. 1.

Fig. 2 presents a comparison of the proposed method, and the convex relaxation method of Chambolle and Pock [7], minimizing the piecewise constant Mumford-Shah functional [21], closely related to the piecewise constant model described above. To evaluate [7] we used the code published by the authors, with the algorithm parameters chosen to obtain visually optimal results: isotropic TV, simple relaxation, initialization with k-means clustering, $K = 8$, and $\lambda = 5.0$. We further compared the proposed method with the graph-cut based approach of [12], which we applied to the piecewise constant model. We iterated segmentation and model-estimation, as described in [12], with initial model parameters obtained with k-means clustering, and the algorithm parameters chosen to obtain optimal results with the same number of regions as the two previous algorithms: 8-connected neighborhood, $\lambda = 1/16$, with label cost set to be zero. From examining the images in Fig. 2, (d),(e) and (f), we observe that in this case the three methods produce comparable results.

Fig. 3 presents segmentation results obtained with the piecewise constant variant of the proposed method, and different values of the threshold T. Specifically, increasing T results in more regions being deemed similar and merged during the evolution process, thus producing less detailed segmentation. The above results were compared to segmentation obtained with [7] and [12], with algorithms' parameters chosen to produce similar number of regions as the proposed method. [12] was used with 8-connected neighborhood, with the initial parameters obtained using k-means clustering for both methods. The results are shown in Fig. 4. We observe that in this case both latter approaches fail to segment one of the objects, namely, the orange candy, and associate part of it with the background.

Fig. 5 presents the segmentation result obtained with the proposed method for an image from the Berkeley Segmentation Dataset[1], along with the ground-truth segmentation. Our method captures the main objects in the image, though it does not detect small image features, such as thin lines and tiny structures.

[1] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`

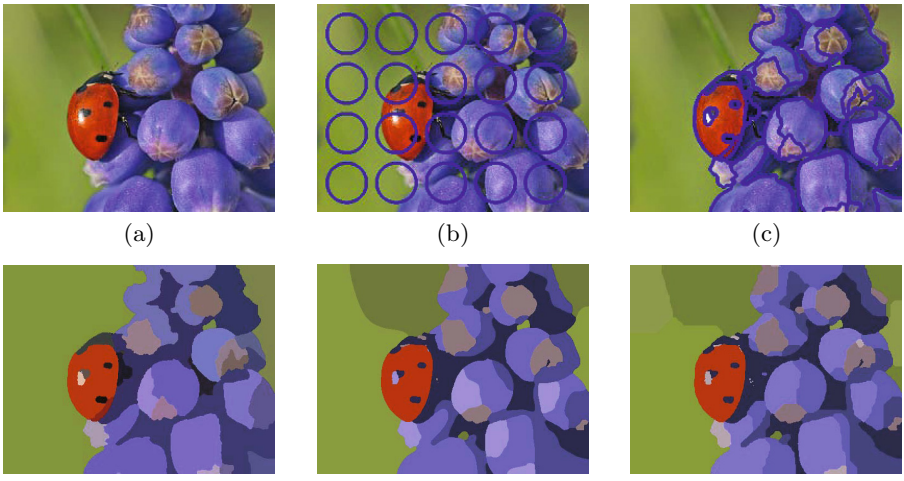


Fig. 2. Comparison of the proposed method using the multi-region piecewise constant model, the convex relaxation approach of [7], and the graph-cut based method of [12]. (a) The original image. (b) Initial contour. (c) Region boundaries detected by our method. (d) Regions detected by our method, colored according to their mean intensity values. (e), (f) The results of [7] and [12], accordingly.

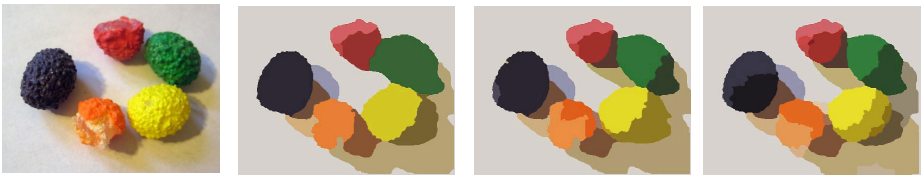


Fig. 3. Segmentation results obtained with the proposed method using different values of the absolute intensity difference T

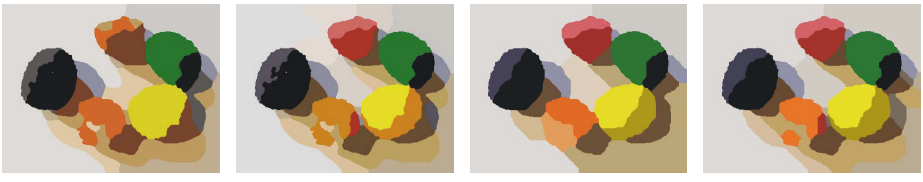


Fig. 4. Segmentation results obtained with (a), (b) [7] and (c), (d) [12], with different algorithm parameters

Fig. 5. Segmentation of an image from the Berkeley Segmentation Dataset. Left to right: the original image, region boundaries obtained using our method with piecewise constant model, ground-truth segmentation.

Fig. 6. Tracking in a thermal camera video sequence from a surveillance camera. Yellow contours show the region boundaries in four frames from the sequence. The leftmost subfigure demonstrates the initial contour.

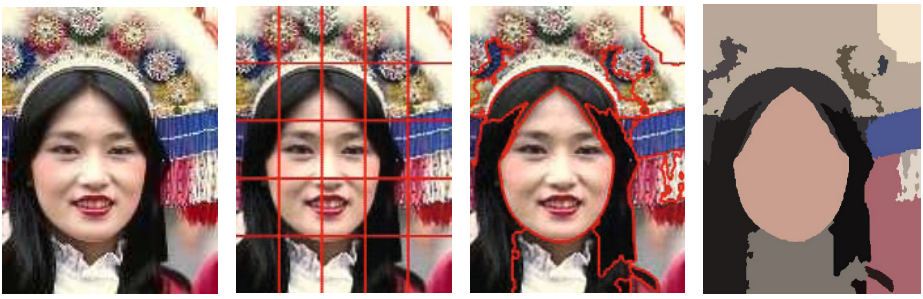


Fig. 7. Segmentation obtained using the Gaussian mixture to model image intensity probability density. Left to right: the original image, the initial contour, region boundaries obtained by our method, regions colored according to their mean intensity values.

This can be overcome by up-sampling the image prior to the segmentation [2]. It also should be noted that some of the object boundaries provided in the ground-truth segmentation and not detected by the proposed method, may be found only using a prior knowledge of the object structure.

In Fig. 6 we demonstrate the application of multi-region piecewise constant model (8) for tracking in a thermal camera video sequence, where the segmentation obtained for k-th frame is used to initialize the algorithm in frame $k+1$. The proposed approach seemlessly allows multiple target tracking in the video sequence. In Fig. 7 we demonstrate the segmentation obtained using the proposed method with Gaussian mixture model. The introduction of more expressive region appearance models naturally allows us to segment more complex images.

5 Conclusions and Future Work

In this paper we addressed the problem of segmenting an image into an arbitrary number of regions using a novel active contours formulation. The proposed framework allows utilizing various region appearance priors and employs the new Voronoi implicit interface method in order to treat multiple regions in a uniform manner, while avoiding metrication errors. Finally, we demonstrated that the proposed method works well on challenging images from various data sets and applications.

Acknowledgements. The authors thank J.A. Sethian of UC Berkeley for intriguing discussions and introducing the VIIM to our group. This research was supported by European Community's FP7-ERC program, grant agreement no. 267414.

References

1. Adam, A., Kimmel, R., Rivlin, E.: On scene segmentation and histograms-based curve evolution. IEEE-TPAMI 31(9), 1708–1714 (2009)
2. Amiaz, T., Lubetzky, E., Kiryati, N.: Coarse to over-fine optical flow estimation. Pattern Recognition 40(9), 2496–2503 (2007)
3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE-TPAMI 23(11), 1222–1239 (2001)
4. Brox, T., Weickert, J.: Level set segmentation with multiple regions. IEEE-TIP 15(10), 3213–3218 (2006)
5. Caselles, V., Catté, F., Coll, T., Dibos, F.: A geometric model for active contours in image processing. Numerische Mathematik 66(1), 1–31 (1993)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. IJCV 22(1), 61–79 (1997)
7. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. JMIV 40(1), 120–145 (2011)
8. Chan, T., Vese, L.: Active contours without edges. IEEE-TIP 10(2), 266–277 (2001)
9. Chopp, D.L.: Some improvements of the fast marching method. SIAM Journal on Scientific Computing 23(1), 230–244 (2001)
10. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE-TPAMI 24(5), 603–619 (2002)
11. Cremers, D., Kohlberger, T., Schnörr, C.: Shape statistics in kernel space for variational image segmentation. Pattern Recognition 36(9), 1929–1943 (2003)
12. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. IJCV 96(1), 1–27 (2012)
13. Freedman, D., Zhang, T.: Active contours for tracking distributions. IEEE-TIP 13(4), 518–526 (2004)
14. Greenspan, H., Ruf, A., Goldberger, J.: Constrained Gaussian mixture model framework for automatic segmentation of MR brain images. IEEE Transactions on Medical Imaging 25(9), 1233–1245 (2006)
15. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. IJCV 1(4), 321–331 (1988)
16. Leventon, M.E., Grimson, W.E.L., Faugeras, O.D.: Statistical shape influence in geodesic active contours. In: CVPR, pp. 1316–1323 (2000)

17. Lucas, B.C., Kazhdan, M., Taylor, R.H.: Multi-object spring level sets (MUSCLE). In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part I. LNCS, vol. 7510, pp. 495–503. Springer, Heidelberg (2012)
18. Malladi, R., Sethian, J., Vemuri, B.: Shape modeling with front propagation: A level set approach. IEEE-TPAMI 17(2), 158–175 (1995)
19. McLachlan, G., Peel, D.: Finite mixture models. Wiley-Interscience (2000)
20. Michailovich, O., Rathi, Y., Tannenbaum, A.: Image segmentation using active contours driven by the Bhattacharyya gradient flow. IEEE-TIP 16(11), 2787–2801 (2007)
21. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics 42(5), 577–685 (1989)
22. Ni, K., Bresson, X., Chan, T., Esedoglu, S.: Local histogram based segmentation using the Wasserstein distance. IJCV 84(1), 97–111 (2009)
23. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79(1), 12–49 (1988)
24. Paragios, N., Deriche, R.: Geodesic active regions: A new framework to deal with frame partition problems in computer vision. Journal of Visual Communication and Image Representation 13(1-2), 249–268 (2002)
25. 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)
26. Riklin-Raviv, T., Kiryati, N., Sochen, N.: Prior-based segmentation by projective registration and level sets. In: ICCV, vol. 1, pp. 204–211. IEEE (2005)
27. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)
28. Sagiv, C., Sochen, N., Zeevi, Y.: Integrated active contours for texture segmentation. IEEE-TIP 15(6), 1633–1646 (2006)
29. Samson, C., Blanc-Féraud, L., Aubert, G., Zerubia, J.: A level set model for image classification. IJCV 40(3), 187–197 (2000)
30. Saye, R., Sethian, J.: The Voronoi Implicit Interface Method for computing multiphase physics. Proceedings of the National Academy of Science 108(49), 19498–19503 (2011)
31. Sethian, J.: A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences 93(4), 1591 (1996)
32. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: CVPR, vol. 2. IEEE (1999)
33. Tsai, A., Yezzi Jr., A., Willsky, A.: Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. IEEE-TIP 10(8), 1169–1186 (2001)
34. Vese, L., Chan, T.: A multiphase level set framework for image segmentation using the Mumford and Shah model. IJCV 50(3), 271–293 (2002)
35. Yezzi Jr, A., Tsai, A., Willsky, A.: A statistical approach to snakes for bimodal and trimodal imagery. In: ICCV, vol. 2, pp. 898–903. IEEE (1999)
36. Zhao, H., Chan, T., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. J. of Computational Physics 127(1), 179–195 (1996)
37. Zhu, S., Yuille, A.: Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE-TPAMI 18(9), 884–900 (1996)
38. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: ICPR, vol. 2, pp. 28–31. IEEE (2004)

Regularized Discrete Optimal Transport

Sira Ferradans[1,⋆], Nicolas Papadakis[2,⋆⋆], Julien Rabin[3], Gabriel Peyré[4], and Jean-François Aujol[5]

[1] Ceremade, Univ. Paris-Dauphine
sira.ferradans@ceremade.dauphine.fr
[2] IMB, Université Bordeaux 1
Nicolas.Papadakis@imag.fr
[3] ENSICAEN, Université de Caen
julien.rabin@unicaen.fr
[4] Ceremade, Univ. Paris-Dauphine
grabriel.peyre@ceremade.dauphine.fr
[5] IMB, Université Bordeaux 1
Jean-Francois.Aujol@math.u-bordeaux1.fr

Abstract. This article introduces a generalization of discrete Optimal Transport that includes a regularity penalty and a relaxation of the bijectivity constraint. The corresponding transport plan is solved by minimizing an energy which is a convexification of an integer optimization problem. We propose to use a proximal splitting scheme to perform the minimization on large scale imaging problems. For un-regularized relaxed transport, we show that the relaxation is tight and that the transport plan is an assignment. In the general case, the regularization prevents the solution from being an assignment, but we show that the corresponding map can be used to solve imaging problems. We show an illustrative application of this discrete regularized transport to color transfer between images. This imaging problem cannot be solved in a satisfying manner without relaxing the bijective assignment constraint because of mass variation across image color palettes. Furthermore, the regularization of the transport plan helps remove colorization artifacts due to noise amplification.

Keywords: Optimal Transport, color transfer, variational regularization, convex optimization, proximal splitting, manifold learning.

1 Introduction

A large class of Image Processing problems involves probability densities estimated from local or global image features. In contrast to most distances from information theory (e.g. the Kullback-Leibler divergence), Optimal Transport

⋆ This work has been supported by the European Research Council (ERC project SIGMA-Vision).

⋆⋆ Nicolas Papadakis acknowledges the support of the French Agence Nationale de la Recherche (ANR) under reference ANR-11-BS01-014-01.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 428–439, 2013.

takes into account the spacial localization of the modes of the densities [1]. Furthermore, it also provides as a by-product a warping (the so-called transport plan) between the densities. This plan can be used to perform image modifications such as color transfer. However, an important flaw of this Optimal Transport plan is that it is usually highly irregular, thus introducing unwanted artifacts in modified images. In this article, we propose a variational formalism to relax and regularize the transport. This novel regularized Optimal Transport improves visually the result for color image modification.

1.1 Optimal Transport and Imaging

Discrete Optimal Transport. The discrete Optimal Transport (OT) is the solution of a convex linear program originally introduced by Kantorovitch. It corresponds to the convex relaxation of a combinatorial problem when the densities are sums of the same number of Diracs. This relaxation is tight (i.e. the solution of the linear program is an assignment) and extends the notion of Optimal Transport to arbitrary sum of weighted Diracs, see for instance [1]. Although there exists dedicated linear solvers (transportation simplex) and combinatorial algorithms (such as the Hungarian and auction algorithms), computing Optimal Transport is still a challenging task for densities composed of thousands of Dirac masses.

Optimal Transport Distance. The OT distance (also known as the Wasserstein distance or the Earth Mover distance) has been shown to produce state of the art results for the comparison of statistical descriptors, see for instance [2].

Optimal Transport Map. Another line of applications of OT makes use of the transport plan to warp an input density on another one. Optimal transport is strongly connected to fluid dynamic partial differential equations [3]. These connexions have been used to perform Image Registration [4]. Color transfer between images is a challenging problem, and has been tackled by computing non-linear mappings between color spaces, see for instance [5,6,7]. For grayscale images, the usual histogram equalization algorithm corresponds to the application of the 1-D Optimal Transport plan to an image, see for instance [8]. It thus makes sense to consider the 3-D Optimal Transport as a mathematically-sound way to perform color palette transfer, see for instance [9] for an approximate transport method.

1.2 Regularized and Relaxed Transport

Removing Transport Artifact. The Optimal Transport map between complicated densities is usually irregular. Using directly this transport plan to perform color transfer creates artifacts and amplifies the noise in flat areas of the image. Since the transfer is computed over the 3-D color space, it does not take into account the pixel-domain regularity of the image. The visual quality of the transfer is thus improved by denoising the resulting transport using a pixel-domain regularization either as a post-processing [10] or by solving a variational problem [10,11].

Transport Regularization. A more theoretically grounded way to tackle the problem of colorization artifacts should use directly a regularized Optimal Transport. This corresponds to adding a regularization penalty to the Optimal Transport energy. This however leads to difficult non-convex variational problems, that have not yet been solved in a satisfying manner either theoretically or numerically. The only theoretical contribution we are aware of is the recent work of Louet and Santambrogio [12]. They show that in 1-D the (un-regularized) Optimal Transport is also the solution of the Sobolev regularized transport problem.

Quadratic Assignment Problems. Regularized transport shares similarities with regularized graph matching, which is a quadratic assignment problem, known to be NP-hard to solve. This class of problems have been convexified using an SDP relaxation of the quadratic assignment problem [13]. Such a relaxation is only tractable for small size problems, and cannot be used for imaging applications. We propose here to use a simpler convexification that works well in practice for imaging problems.

Graph Regularization. For imaging applications, we use regularizations built on top of a graph structure connecting neighboring points in the input density. This follows ideas introduced in manifold learning [14], that have been applied to various Image Processing problems [15]. Using these regularizations enables us to design regularizations that are adapted to the geometry of the input density, that often has a manifold-like structure.

Transport Relaxation. The result of Louet and Santambrogio [12] is deceiving from the applications point of view, since it shows that, in 1-D, no regularization is possible if one maintains a 1:1 assignment between the two densities. This is our first motivation for introducing a relaxed transport which is not a bijection between the densities. Another (more practical) motivation is that relaxation is crucial to solve imaging problems such as color transfer. Indeed, the color distributions of natural images are multi-modals. An ideal color transfer should match the modes together. This cannot be achieved by classical Optimal Transport because these modes often do not have the same mass. A typical example is for two images with strong foreground and background dominant colors (thus having bi-modal densities) but where the proportion of pixels in foreground and background are not the same. Such simple examples cannot be handle properly with Optimal Transport. Allowing a controlled variation of the matched densities thus requires an appropriate relaxation of the bijective matching constraint.

1.3 Contributions

In this paper, we generalize the discrete formulation of Optimal Transportation to tackle the two major flaws that we just mentioned: i) the lack of regularity of the transport and ii) the need for a relaxed matching between densities. Our main contribution is the integration of these two properties in a unified convex variational problem. This problem can be solved using standard convex

optimization procedures such as proximal splitting methods. Our second contribution is the application of this framework to the problem of color transfer. Numerical results show the relevance of this approach to this particular imaging problem.

2 Discrete Optimal Transport

Optimal Assignment. We consider discrete measures in $\mathbb{R}^d$ with a fixed number N of points, that we write as $\mu_X = \frac{1}{N}\sum_{i=1}^N \delta_{X_i}$ where δ_a is the Dirac at position $a \in \mathbb{R}^d$, and $X = (X_i)_{i=1}^N \in \mathbb{R}^{N\times d}$, is the position of the N points supporting the distribution.

The Optimal Transport between two such distributions solves the optimal assignment

$$W(\mu_X, \mu_Y)^2 = \sum_i C_{i,\sigma^\star(i)} \quad \text{where} \quad \sigma^\star \in \operatorname*{argmin}_{\sigma\in\mathcal{S}_1} \sum_i C_{i,\sigma(i)} \tag{1}$$

where $\mathcal{S}_1$ is the set of permutation of N indexes.

A usual choice is to consider the L^α Wasserstein distance for some $\alpha > 0$, so that the cost is $C_{i,j} = \|X_i - Y_j\|^\alpha$ where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^d$.

Convex Relaxation. The Kantorovitch Optimal Transport formulation uses the embedding of permutation σ as permutation matrix $\mathcal{M}(\sigma) \in \mathbb{R}^{N\times N}$

$$\mathcal{M}(\sigma)_{i,j} = \begin{cases} 1 & \text{if} \quad j = \sigma(i), \\ 0 & \text{otherwise.} \end{cases}$$

The convex hull of permutation matrices $\mathcal{M}(\mathcal{S}_1)$ is the set of bi-stochastic matrices

$$\bar{\mathcal{S}}_1 = \left\{ \Sigma \in \mathbb{R}^{N\times N} \setminus \Sigma\mathbb{I} = \mathbb{I}, \Sigma^*\mathbb{I} = \mathbb{I}, \Sigma \geqslant 0 \right\}$$

where $\mathbb{I} = (1, \ldots, 1)^* \in \mathbb{R}^N$, where A^* is the adjoint of the matrix A, which for real matrices amounts to the transpose. One can show that the relaxation is tight, i.e. there exists a solution $\Sigma^\star$ of

$$\min_{\Sigma\in\bar{\mathcal{S}}_1} \langle C, \Sigma\rangle \quad \text{where} \quad \langle C, \Sigma\rangle = \sum_{i,j=1}^N C_{i,j}\Sigma_{i,j}$$

such that $\Sigma^\star = \mathcal{M}(\sigma^\star)$ where $\sigma^\star$ is a solution of (1).

3 Relaxed Transport

For many applications in Imaging Science, it is not desirable to impose that the mapping σ between X and Y is 1:1. This is for instance the case for the colorization problem we consider in Section 5 where the ratio of similar colors across the image is not constant.

We relax this constraint by only imposing a maximum number of elements of X linked to a single element of Y

$$\mathcal{S}_\kappa = \left\{\sigma : \{1,\dots,N\} \to \{1,\dots,N\} \setminus \forall\, j = 1,\dots,N,\ |\sigma^{-1}(j)| \leqslant \kappa\right\}$$

where $\kappa \in \mathbb{N}^*$ is a maximum capacity parameter. Note that for $\kappa = 1$ one recovers the set $\mathcal{S}_1$ of permutations. The natural convex relaxation is

$$\bar{\mathcal{S}}_\kappa = \left\{\Sigma \in \mathbb{R}^{N\times N} \setminus \Sigma^*\mathbb{I} \leqslant \kappa\mathbb{I}, \Sigma\mathbb{I} = \mathbb{I}, \Sigma \geqslant 0\right\}.$$

The following proposition shows that this relaxation is tight when κ is an integer.

Proposition 1. *For $\kappa \in \mathbb{N}^*$, there exists a solution $\Sigma^\star$ of*

$$\min_{\Sigma\in\bar{\mathcal{S}}_\kappa} \langle C,\, \Sigma\rangle \tag{2}$$

such that $\Sigma^\star = \mathcal{M}(\sigma^\star)$ where $\sigma^\star$ is solution of

$$\min_{\sigma\in\mathcal{S}_\kappa} \sum_i C_{i,\sigma(i)}. \tag{3}$$

Proof. One can write $\bar{\mathcal{S}}_\kappa = \left\{\Sigma \in \mathbb{R}^{N\times N} \setminus \mathcal{A}(\Sigma) \leqslant b_\kappa\right\}$ where $\mathcal{A}$ is the linear mapping $\mathcal{A}(\Sigma) = (-\Sigma, \Sigma\mathbb{I}, \Sigma^*\mathbb{I}) \in \mathbb{R}^{N\times N} \times \mathbb{R}^N \times \mathbb{R}^N$ and $b_\kappa = (0_{N\times N}, \mathbb{I}, \kappa\mathbb{I})$. A standard result shows that $\mathcal{A}$ is a totally unimodular matrix [16]. For any $\kappa \in \mathbb{N}$, the vector b_κ has integer coefficients, and thus the polytope $\bar{\mathcal{S}}_\kappa$ has integer vertices. Since there is always a solution of the linear program (2) which is a vertex of $\bar{\mathcal{S}}_\kappa$, it has coefficients in $\{0,1\}$.

Note however that this relaxation is in general not tight when κ is an arbitrary real number. When $\kappa \geqslant N$, there is no restriction on the map $\sigma \in \mathcal{S}_\kappa$. The solution of (3) is then the nearest neighbor assignment,

$$\forall\, i = 1,\dots,N, \quad \sigma^\star(i) = \operatorname*{argmin}_{0\leqslant j\leqslant N} C_{i,j}. \tag{4}$$

3.1 Numerical Illustrations

In Fig. 1, we show a simple example to illustrate the properties of the method proposed so far. Given a set of points X (in blue) we compute the mapping with the set of points Y (in red), that is a solution of (2). For all the mappings between X_i and Y_j with a value $\Sigma_{i,j} > 0$, we draw a line, solid if $\Sigma_{i,j} = 1$, and dashed otherwise.

As we pointed out in last section, for non integer values of κ, the mappings $\Sigma_{i,j}$ are in $[0,1]$ while for integer values of κ, $\Sigma_{i,j} \in \{0,1\}$. Note that as we increase the values of κ (Fig. 1, right), the points in X tend to be mapped to the closer points in Y, as defined in (4).

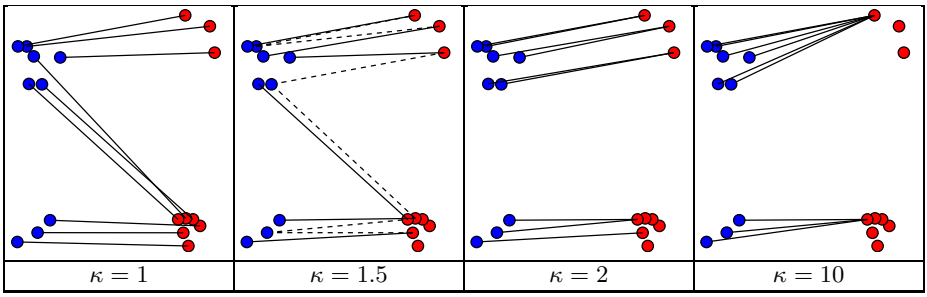


Fig. 1. Relaxed transport computed between X (blue dots) and Y (red dots) for different values of κ. Note that $\kappa = 1$ corresponds to classical OT. A dashed line between X_i and Y_j indicates that $\Sigma_{i,j}$ is not an integer.

4 Discrete Regularized Transport

4.1 Gradient on Graphs

One can view a relaxed assignment $\sigma \in \mathcal{S}_\kappa$ as a vector field $X_i \mapsto V_i = Y_{\sigma(i)} - X_i$ defined on the point cloud X. A usual way to impose regularity of such a map V is by measuring the amplitude of its derivatives GV where $G : \mathbb{R}^{N\times d} \to \mathbb{R}^{P\times d}$ is a discrete differential operator. A natural way to define a gradient is by imposing a graph structure defined by $\mathcal{G}_X \subset \{1,\dots,N\}^2$, and where $P = |\mathcal{G}_X|$. This graph structure is application dependent, and one can think of it as some sort of nearest neighbor graph. Section 5 gives an example of such a construction for the color transfer problem. The gradient measures the (weighted) difference along the edges of the graph

$$GV = (w_{i,j}(V_i - V_j))_{(i,j)\in\mathcal{G}_X} \in \mathbb{R}^{P\times d}.$$

where $w_{i,j} > 0$ is some weight. A classical choice, to ensure consistency with the directional derivative, is to choose $w_{i,j} = \|X_i - X_j\|^{-1}$.

4.2 Convex Formulation

The regularity of a transport map $V \in \mathbb{R}^{N\times d}$ is then measured according to some norm of GV, that we choose here for simplicity to be the vectorial ℓ^p norm $J_p(GV)$

$$J_p(GV) = \sum_{(i,j)\in\mathcal{G}_x} \|w_{i,j}(V_i - V_j)\|_2^p.$$

The case $p = 2$ corresponds to a graph-based Sobolev H^1 norm, whereas the case $p = 1$ corresponds to a graph-based total variation norm, see for instance [15] for applications of these functional to imaging problem regularization.

The relaxed and regularized optimal assignment problem thus reads

$$\min_{\sigma \in \mathcal{S}_\kappa} \sum_i C_{i,\sigma(i)} + \lambda J_p(G(X - Y \circ \sigma)) \tag{5}$$

where $Y \circ \sigma = (Y_{\sigma(i)})_{i=1}^N \in \mathbb{R}^{N \times d}$. To introduce a convexified regularized energy, we replace the relaxed assignment $\sigma \in \mathcal{S}_\kappa$ by $\Sigma \in \bar{\mathcal{S}}_\kappa$, and consider $X - \Sigma Y$ in place of the mapping $X - Y \circ \sigma$. We consider the following relaxed and regularized convex formulation

$$\Sigma^\star \in \min_{\Sigma \in \bar{\mathcal{S}}_\kappa} \langle C,\, \Sigma \rangle + \lambda J_p(G(X - \Sigma Y)) \tag{6}$$

The case $(\kappa, \lambda) = (1, 0)$ corresponds to the usual Optimal Transport, and $\lambda = 0$ corresponds to the un-regularized formulation (3).

4.3 Minimization Algorithm

Problem (6) is a convex minimization. In the case $p = 2$, it corresponds to a quadratic minimization, whereas in the case $p = 1$ it can be cast as a conic optimization problem. They can be solved for medium-scale problem using standard interior point methods. An alternative solution is to use first order proximal scheme (see for instance [17]), that are well tailored for such highly structured problems.

Proximal Splitting. Problem (6) can be reformulated as

$$\min_{\Sigma} \langle C,\, \Sigma \rangle + \lambda J_p(G(X - \Sigma Y)) + \iota_{\mathcal{D}_1}(\Sigma) + \iota_{\mathcal{C}_\kappa}(\Sigma^*) \tag{7}$$

where ι_C is the indicator function of a convex set C and we introduced the constraint sets

$$\mathcal{C}_\kappa = \left\{ \Sigma \setminus \Sigma \geqslant 0,\ \Sigma \mathbb{I} \leqslant \kappa \mathbb{I}, \Sigma \in \mathbb{R}^{N \times N} \right\}.$$

and $\iota_{\mathcal{D}_1}$ is the indicator function of the convex set $\mathcal{D}_1$ defined as:

$$\mathcal{D}_1 = \left\{ \Sigma \setminus \Sigma \geqslant 0,\ \Sigma \mathbb{I} = \mathbb{I}, \Sigma \in \mathbb{R}^{N \times N} \right\},$$

where every line of $\Sigma \in \mathcal{D}_1$ belongs to the N-dimensional simplex. Problem (7) can be re-casted as a minimization of the form

$$\min_{\Sigma \in \mathbb{R}^{N \times N}} F(K(\Sigma)) + H(\Sigma)$$

$$\text{where} \quad \begin{cases} K(\Sigma) = (\Sigma, G\Sigma Y), \quad K^*(\Sigma, U) = \Sigma + G^* U Y^*, \\ F(\Sigma, U) = \lambda J_p(U - GX) + \iota_{\mathcal{D}_1}(\Sigma), \\ H(\Sigma) = \langle C,\, \Sigma \rangle + \iota_{\mathcal{C}_\kappa}(\Sigma^*). \end{cases}$$

Orthogonal Projection on Constraint Sets. The proximal operators read

$$\begin{aligned}\mathrm{Prox}_{\gamma F}(\Sigma, U) &= (\mathrm{Proj}_{\mathcal{D}_1}(\Sigma), \lambda GX + \mathrm{Prox}_{\gamma J_p(\cdot)}(U - \lambda GX)) \\ \mathrm{Prox}_{\gamma H}(\Sigma) &= \left(\mathrm{Proj}_{\mathcal{C}_\kappa}((\Sigma - \gamma C)^*)\right)^*\end{aligned}$$

where the orthogonal projection on $\mathcal{C}_\kappa$ is computed for each line Σ_ℓ of a matrix Σ as

$$\tilde{\Sigma} = \mathrm{Proj}_{\mathcal{C}_\kappa}(\Sigma) \quad \text{where} \quad \tilde{\Sigma}_\ell = \left\{ \begin{array}{ll} \max(0, \Sigma_l) & \text{if } \max(0, \Sigma_l)\mathbb{I} \leqslant \kappa \\ \mathrm{Proj}_{\mathcal{D}_\kappa}(\Sigma_l) & \text{otherwise,} \end{array} \right\}$$

where $\mathrm{Proj}_{\mathcal{D}_\kappa}(V)$ is the projection of the line vector V on the convex set $\mathcal{D}_\kappa$

$$\mathcal{D}_\kappa = \{V \setminus V \geqslant 0,\ V\mathbb{I} = \kappa\}.$$

This last projection, as well as the projection on $\mathcal{D}_1$, can be efficiently computed as detailed for instance in [18].

Proximal Operators. Let us now recall that the proximal operator of a function F is defined as

$$\mathrm{Prox}_{\gamma F}(\Sigma) = \operatorname*{argmin}_{\tilde{\Sigma}} \frac{1}{2}\|\Sigma - \tilde{\Sigma}\|^2 + \gamma F(\tilde{\Sigma}).$$

One can check that for $p = 1$ and $p = 2$, the proximal operator of J_p evaluated at $U \in \mathbb{R}^{N \times d}$ can be computed in closed form as:

$$\mathrm{Prox}_{\lambda\gamma J_1(\cdot)}(U)_i = \max\left(0, 1 - \frac{\lambda\gamma}{\|U_i\|}\right) U_i \quad \text{and} \quad \mathrm{Prox}_{\lambda\gamma J_2(\cdot)}(U)_i \quad = \frac{U_i}{1 + 2\gamma\lambda}$$

Note that being able to compute the proximal mapping of F is equivalent to being able to compute the proximal mapping of F^*, thanks to Moreau's identity

$$\Sigma = \mathrm{Prox}_{\gamma F^*}(\Sigma) + \gamma\, \mathrm{Prox}_{F/\gamma}(\Sigma/\gamma).$$

Primal-Dual Splitting Scheme. The primal-dual algorithm of [17] applied to our problem finally reads

$$\begin{aligned}\Gamma^{k+1} &= \mathrm{Prox}_{\mu F^*}(\Gamma^k + \mu K(\tilde{\Sigma}^k), \\ \Sigma^{k+1} &= \mathrm{Prox}_{\tau H}(\Sigma^k - \tau K^*(\Gamma^{k+1})), \\ \tilde{\Sigma}^{k+1} &= \Sigma^{k+1} + \theta(\Sigma^{k+1} - \Sigma^k),\end{aligned}$$

where $\theta \in [0; 1]$ and the two other parameters should satisfy $\mu\tau\|K\|^2 < 1$ where $\|K\|$ is the spectral norm of the operator K. Under these conditions, it is shown in [17] that Σ^k converges to a solution $\Sigma^\star$ of (6).

4.4 Numerical Illustrations

In Fig. 2, we can observe, on a synthetic example, the influence of the parameters κ and λ (see equation (6)). Given two sets X (in blue) and Y (in red), we compute the mapping an optimal regularized transport $\Sigma^\star$ solving eq. 6 with the minimization algorithm proposed in Section 4.3, and plot in green the set $\Sigma^\star Y$. Points X_i and Y_j are connected by a line if $\Sigma^\star_{i,j} > 0$, which is dashed if $\Sigma^\star_{i,j} \in]0,1[$ or solid if $\Sigma^\star_{i,j} = 1$. For $\lambda = 0$ and $\kappa = 1$, one obtains the classical OT solution. The influence of regularization can be observed as we increase λ: the influence of the two outliers in Y (top of the figures) in the mappings is reduced.

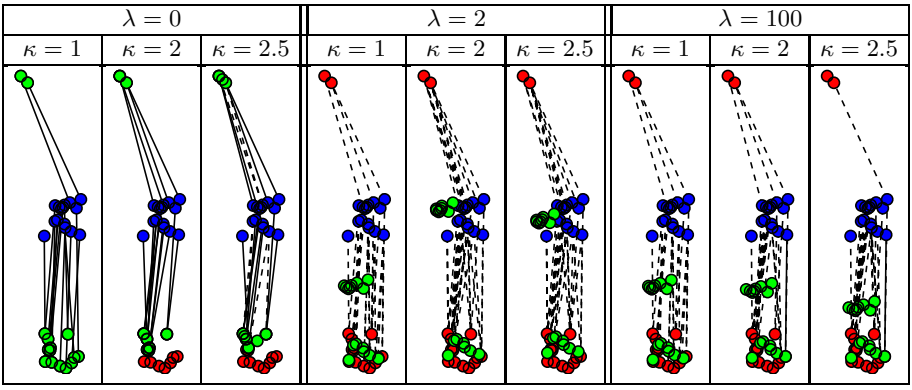


Fig. 2. Given two sets of points X (in blue) and Y (in red), we show in ΣY (in green), and the mappings $\Sigma_{i,j}$ as lines connecting X_i and Y_j, which are dashed if $\Sigma_{i,j} \in]0,1[$ and solid if it is an integer

5 Application to Color Transfer

The color transfer problem consists in modifying an input image $X^0 \in \mathbb{R}^{N_0 \times d}$ (here $d = 3$ for RGB color image) to obtain $\tilde{X}^0 = T(X^0)$ whose color palette (its pixel empirical distribution) $\mu_{\tilde{X}^0}$ is equal or close to a target color distribution μ_{Y^0}. This target distribution is here chosen as the empirical distribution of a second image Y^0.

Nearest-Neighbors Transport Interpolation. We compute T as an interpolation of a regularized Optimal Transport map between sub-sampled point clouds $X, Y \in \mathbb{R}^{N \times d}$. These two points clouds X, Y are computed by applying the K-means algorithm to the input clouds X^0, Y^0.

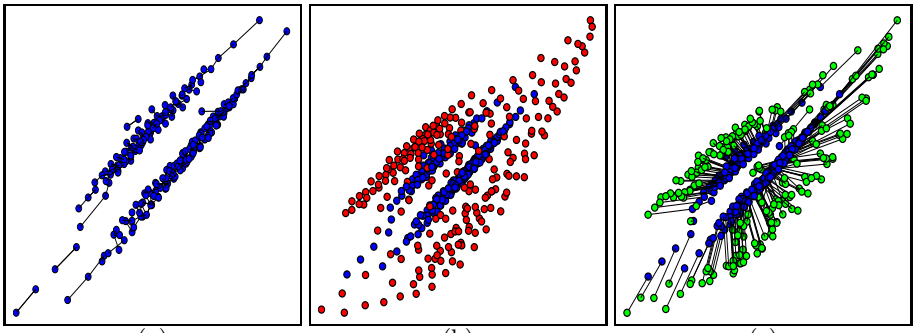


Fig. 3. In blue, the empirical distribution of the "Wheat" image pair (Fig. 4 second column), projected on the Red-Green plane. **(a)** Nearest neighbor graph $\mathcal{G}_X$ with $K = 1$, **(b)** Y in red, and **(c)** $U = \Sigma^\star Y$ in green with the lines connecting X_i to U_i.

The regularized Optimal Transport matrix $\Sigma^\star \in \mathbb{R}^{N\times N}$ is obtained by solving (6). The quantized regularized transport then maps X to $U = \Sigma Y$. It is then extended to the whole space by a nearest neighbor interpolation

$$\forall\, x \in \mathbb{R}^d, \quad T(x) = U_{i(x)} \quad \text{where} \quad i(x) = \operatorname*{argmin}_{1\leqslant i\leqslant N} \|x - X_i\|.$$

This transport is then applied to the input image to obtain the new pixel values $(\tilde{X}^0)_i = T(X_i^0)$.

Graph and G Operator. As exposed in Section 4.1, computing a regularized transfer requires the user to design a graph structure $\mathcal{G}_X$ and weights $w_{i,j}$ that reflects the geometry of the input cloud X that supports the distribution μ_X. Inspired by several recent works on manifold learning (see Section 1.2), we use here a K-nearest neighbor graph, where K is the number of edges adjacent to each vertex, i.e. $|\{j \setminus (i,j) \in \mathcal{G}_x\}| = K$.

Comparison with the State of the Art. In Fig. 4, we show some results and compare them with the methods of Pitie et al. [9] and Papadakis et al. [10]. The goal of the experiment is to transfer the color palette of the images in the second row to the image on the first row. Note that the methods in the state of the art introduce color aberration (in the first column there is violet outside the flower, and in the second column the wheat is blueish), which can be avoided with the proposed method by an appropriate choice of λ and κ.

Implementation Details. The results shown in this paper were obtained setting $w_{i,j} = \|X_i - X_j\|^{-1}$ and $N = 400$. The set of parameters (λ, κ, K, p) used in Fig. 4 are, by column from left to right (1400,1.05,2,1), (1000,1.2,1,1), and (1000,1,1,1).

Fig. 4. Comparison between the results obtained with our method and with the methods of [9] and [10]

6 Conclusion

In this paper, we have proposed a generalization of discrete Optimal Transport that enables to regularize the transport map and to relax the bijectivity constraints. We show how this novel class of transports can be applied to color transfer. Regularization is crucial to reduce noise amplification artifacts, while relaxation enables to cope with mass variation of the modes of the color palettes.

References

1. Villani, C.: Topics in Optimal Transportation. Graduate Studies in Mathematics Series. American Mathematical Society (2003)
2. Rubner, Y., Tomasi, C., Guibas, L.: A metric for distributions with applications to image databases. In: International Conference on Computer Vision (ICCV 1998), pp. 59–66 (1998)
3. Benamou, J.D., Brenier, Y.: A computational fluid mechanics solution of the monge-katorovich mass transfer problem. Numerische Mathematik 84, 375–393 (2000)
4. Haker, S., Zhu, L., Tannenbaum, A., Angenent, S.: Optimal mass transport for registration and warping. International Journal of Computer Vision 60, 225–240 (2004)
5. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Transactions on Computer Graphics and Applications 21, 34–41 (2001)
6. Morovic, J., Sun, P.L.: Accurate 3d image colour histogram transformation. Pattern Recognition Letters 24, 1725–1735 (2003)
7. McCollum, A.J., Clocksin, W.F.: Multidimensional histogram equalization and modification. In: International Conference on Image Analysis and Processing (ICIAP 2007), pp. 659–664 (2007)
8. Delon, J.: Midway image equalization. Journal of Mathematical Imaging and Vision 21, 119–134 (2004)
9. Pitié, F., Kokaram, A.C., Dahyot, R.: Automated colour grading using colour distribution transfer. Computer Vision and Image Understanding 107, 123–137 (2007)
10. Papadakis, N., Provenzi, E., Caselles, V.: A variational model for histogram transfer of color images. IEEE Transactions on Image Processing 20, 1682–1695 (2011)
11. Rabin, J., Peyré, G.: Wasserstein regularization of imaging problem. In: IEEE International Conderence on Image Processing (ICIP 2011), pp. 1541–1544 (2011)
12. Louet, J., Santambrogio, F.: A sharp inequality for transport maps in via approximation. Applied Mathematics Letters 25, 648–653 (2012)
13. Schellewald, C., Roth, S., Schnörr, C.: Evaluation of a convex relaxation to a quadratic assignment matching approach for relational object views. Image Vision Comput 25, 1301–1314 (2007)
14. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290, 2319–2323 (2000)
15. Elmoataz, A., Lezoray, O., Bougleux, S.: Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. Trans. Img. Proc. 17, 1047–1060 (2008)
16. Schrijver, A.: Theory of linear and integer programming. John Wiley & Sons, Inc., New York (1986)
17. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40, 120–145 (2011)
18. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ^1-ball for learning in high dimensions. In: Proc. ICML 2008, pp. 272–279. ACM, New York (2008)

Variational Method for Computing Average Images of Biological Organs*

Shun Inagaki[1], Atsushi Imiya[2], Hidekata Hontani[3], Shouhei Hanaoka[4], and Yoshitaka Masutani[4]

[1] Graduate School of Advanced Integration Science, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba, 263-8522, Japan
[2] Institute of Media and Information Technology, Chiba University 1-33 Yayoi-cho, Inage-ku, Chiba, 263-8522, Japan
[3] Department of Computer Science, Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya, Aichi, 466-8555, Japan
[4] Department of Radiology, The University of Tokyo Hospital Division of Radiology and Biomedical Engineering, Graduate School of Medicine The University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8655, Japan

Abstract. In this paper, we develop a variational method for the computation of average images of biological organs in three-dimensional Euclidean space. The average of three-dimensional biological organs is an essential feature to discriminate abnormal organs from normal organs. We combine the diffusion registration technique and optical flow computation for the computation of spatial deformation field between the averages and each input organ. We define the average as the shape which minimises the total deformation.

1 Introduction

In this paper, we introduce an algorithm for the computation of the average shape of a collection of biological organs in three-dimensional Euclidean space. We first define the warp between a pair of shapes. Using the warp to represent the deformation between shapes, we define the average shape in three-dimensional space as the shape which minimises the total deformation to the given collection of shapes. By computing the difference between the average shapes of organs, it is possible to discriminate normal organs from abnormal organs as a prescreening step in medical diagnosis.

In medical image registration, the establishment of relations between different images is the main issue in research. This registration process between images, clarifies the difference between images which is used for medical diagnosis. This

* This research was supported by "Computational anatomy for computer-aided diagnosis and therapy: Frontiers of medical image sciences" funded by the Grant-in-Aid for Scientific Research on Innovative Areas, MEXT, Japan, the Grants-in-Aid for Scientific Research funded by Japan Society of the Promotion of Sciences and the Grant-in-Aid for Young Scientists (A), JSPS, Japan.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 440–451, 2013.

registration process is mainly achieved by a matching process, which is an established fundamental methodology in pattern recognition.

In medical image diagnosis and retrieval [2,3], the average image and shape of individual organs provide essential properties for the general expression of organs. Shape retrieval categorises and classifies shapes, and finds shapes from portions of shapes. In shape retrieval, the matching of shapes based on the deformorpholism of shapes [8,9] and the descriptor of shape boundary contours [11] is used. In the matching process for discrete shapes, the string edit distance [5,7] computed by dynamic programming is a fundamental tool. Moreover, in the matching process of images, the variational registration strategy [2,3] is a typical tool. In computational anatomy, the statistical average shape, which is computed using principal component analysis of the shape descriptor, is well defined [13,14]. In both structure pattern recognition [5,6] and variation registration [2,4], the average shape of a collection of given shapes is of interest.

There are some methods for the calculation of the average shape that are based on the mathematical definition that shapes are the boundary contours of physical objects with shapes defined in the shape space $\mathbb{S}$ [17,18,19]. This definition is suitable for dealing with highly nonlinear geometric variations [17]. Setting $d(S_i, S_j)$ to be a distance measure [1] between shapes S_i and S_j over the shape space $\mathbb{S}$, the averages and median in a shape subset $\{S_i\}_{i=1}^n = \mathbb{S}_n \subset \mathbb{S}$ are

$$\text{average}(\mathbb{S}_n) = \arg\left(\min_{S\in\mathbb{S}} \sum_{i=1}^{n} d(S_i, S)\right), \tag{1}$$

$$\text{median }(\mathbb{S}_n) = \arg\left(\min_{S\in\mathbb{S}_n} \sum_{i=1}^{n} d(S_i, S)\right). \tag{2}$$

We adopt eq. (1) for the construction of the average shape for the numerical atlas in computational anatomy. Equation (1) is an ill-posed problem. Therefore, setting $\{\phi_i\}_{i=1}^n$ to be a collection of shape-deformation operation

$$\text{average}(\mathbb{S}_n) = \arg\left(\min_{S\in\mathbb{S}} \sum_{i=1}^{n} d(\mathbb{S}_i, \phi_i(\mathbb{S})) + \lambda P(\mathbb{S}) + \mu Q(\phi_i)\right), \tag{3}$$

where P and Q are regularizes for $\mathbb{S}$ and ϕ_i. We establish a variational method to compute the average of simple planar curves, which are derived as the boundary curves of biological organs.

We give the preliminaries for the calculation of average images by the variational method in Section 2. In Section 3 and Section 4, we derive our model using the variational registration. Moreover, we test our method by computing the deformation fields between average images and input images in Section 5 and conclude in Section 6.

2 Preliminaries

Diffusion Registration[16]
The nonparametric registration of images $g(\boldsymbol{x})$ to $f(\boldsymbol{x})$ is achieved by minimising the criterion

$$J[\boldsymbol{u}] = D[g, f; \boldsymbol{u}] + \alpha S[\boldsymbol{u}], \tag{4}$$

where $\boldsymbol{u}$ is the displacement vector between $g(\boldsymbol{x})$ and $f(\boldsymbol{x})$, D is the flow warp between f and g and $S(\boldsymbol{u})$ is the smoothness constraint on the deformation field $\boldsymbol{u}$. In diffusion registration [16],

$$D[g, f; \boldsymbol{u}] = \frac{1}{2}\int_{\Omega} |f_{\boldsymbol{u}} - g|^2 d\boldsymbol{x}, \qquad S[\boldsymbol{u}] = \frac{1}{2}\int_{\Omega} \|\nabla \boldsymbol{u}\|^2 d\boldsymbol{x}, \tag{5}$$

where $f_{\boldsymbol{u}}$ is an image warped by $\boldsymbol{u}$ such that $f_{\boldsymbol{u}} = f(\boldsymbol{x} - \boldsymbol{u})$. This regulariser causes images to be smooth [16] and it is introduced in context of optical flow [15].

Local Linear Space and Geometric Perturbation
If an image $f(\boldsymbol{x})$ defined in the n-dimensional Euclidean space $\mathbf{R}^n$ is geometrically perturbed, it satisfies the relation

$$f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \delta^{\top} \nabla f(\boldsymbol{x}). \tag{6}$$

This is because for $i = 1, 2, \cdots, n$, $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\top}$,

$$\int_{\mathbf{R}^n} f(\boldsymbol{x}) \partial_{x_i} f(\boldsymbol{x}) d\boldsymbol{x} = 0, \qquad \partial_{x_i} f(\boldsymbol{x}) = \frac{\partial f}{\partial x_i} \tag{7}$$

and $\{\partial_{x_i} f(\boldsymbol{x})\}_{i=1}^n$ are independent [1].

For an image f

$$g(\boldsymbol{x}) = f(\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}) \tag{8}$$

for a small rotation $\boldsymbol{R}$ and small translation vector $\boldsymbol{t}$, we can assume the relation

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{k=1}^{n} a_k \partial_{x_k} f(\boldsymbol{x}). \tag{9}$$

Equation (9) implies that the number of independent images in the collection of images

$$L(f) = \{f_{ij} | f_{ij}(\boldsymbol{x}) = \lambda f(\boldsymbol{R}_i \boldsymbol{x} + \boldsymbol{t}_j)\}_{i,j=1}^{p,q} \tag{10}$$

is $(n + 1)$.

Therefore, Setting $f \otimes g$ to be a linear operation such that $(f \otimes g)h = (h, g)f$, where $(\cdot, \cdot)$ is the inner product of image space, the covariance of $L(f)$ is defined as $\boldsymbol{L}_f = E_{ij} f_{ij} \otimes f_{ij}$ where $E_{i=1}^k(f_i)$ is the expectation of $\{f_i\}_{i=1}^n$. We can use the first $(n + 1)$ principal vectors of $\boldsymbol{L}_f$ as the local bases for image expression.

[1] For a calculation of volumetric image $f(\boldsymbol{x})$ such the $\int_{\mathbb{R}^2} |f|^2 \mathrm{d}\boldsymbol{x} < \infty$, $\int_{\mathbb{R}^2} |\partial_x f|^2 \mathrm{d}\boldsymbol{x} < \infty$, $\int_{\mathbb{R}^2} |\partial_y f|^2 \mathrm{d}\boldsymbol{x} < \infty$, $\int_{\mathbb{R}^2} |\partial_y f|^2 \mathrm{d}\boldsymbol{x} < \infty$, we have the relation $\int f \partial_x f \mathrm{d}\boldsymbol{x} = \int f \partial_y f \mathrm{d}\boldsymbol{x} = \int f \partial_z f \mathrm{d}\boldsymbol{x} = 0$, since $\partial_x f \neq \lambda \partial_y f$, $\partial_y f \neq \mu \partial_z f$, $\partial_z f \neq \kappa \partial_x f$. $f, \partial_x f, \partial_y f, \partial_z f$ are independent.

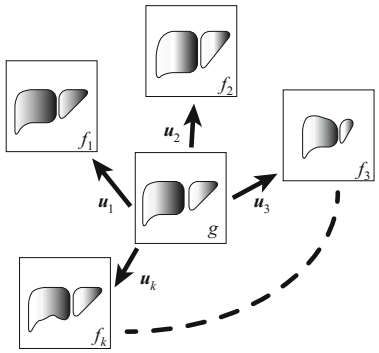


Fig. 1. Concept of variational mean image calculation. A variational mean image represents the mean image among m images f_k $(1 \leq k \leq m)$. The mean image warps each image f_k with deformation fields $\boldsymbol{u}_k$ $(1 \leq k \leq m)$ defined on the variational mean image g. The deformation fields approach to each image f_k from the variational mean image g.

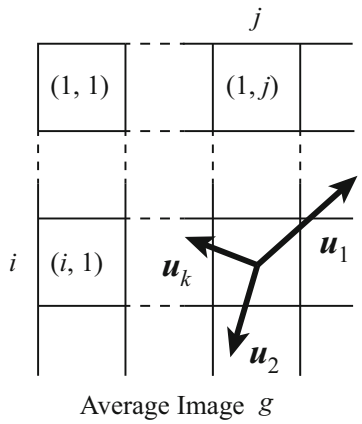


Fig. 2. Regularisation term of deformation fields. This figure suggests that the mean image is the median point among the input images when the sums of vectors on each pixel are minimised.

3 Modeling of Variational Average

First we define the variational average image g. The variational average image g should be a "average" image when the sum of the variations $\boldsymbol{u}_k$ $(1 \leq k \leq m)$ defined on g applied to each image f_k $(1 \leq k \leq m)$ is minimised. In other words, we define a variational average image g as an image with minimised deformation energies. Figure 1 depicts the concept of the variational average image computation.

We define an energy function for calculating a variational average image as

$$J(g, u_1, u_2, \ldots, u_k) = J_d + J_g + J_s + J_c. \tag{11}$$

Here, J_d is a data term that models the image registration. J_g and J_s are regularisation terms which cause flow fields and the variational image to be smooth, respectively. J_c denotes a constraint on the flow fields.

The data term models image registration problem and is given by,

$$J_d = \sum_{k=1}^{m} \int_{\Omega} (g(\boldsymbol{x}) - f_k(\boldsymbol{x} - \boldsymbol{u}_k))^2 d\boldsymbol{x}. \tag{12}$$

We generate the average image at the median point among the input images to minimise the distance metric

$$J_c = \gamma \int_{\Omega} (\sum_{k=1}^{m} \boldsymbol{u}_k)^2 d\boldsymbol{x}. \tag{13}$$

Figure 2 depicts the geometric relation of this term.

We deal with this ill-posed problem by introducing the regularisation terms J_g and J_s. They cause the average image and deformation fields to be smooth in their frames, respectively,

$$J_g = \alpha \int_{\Omega} (\nabla g)^2 d\boldsymbol{x}, \tag{14}$$

$$J_s = \beta \sum_{k=1}^{m} \int_{\Omega} (\nabla \boldsymbol{u}_k)^2 d\boldsymbol{x}. \tag{15}$$

Since J_d, J_g, J_s, and J_c are quadric functions, the functional defined by eq. 11 is convex. Therefore, the minimizer of eq. 11 is unique.

Minimising the energy function (11), we can obtain the variational average image g and deformation fields $\boldsymbol{u}_k$. We derive Euler-Lagrange equations,

$$\alpha \Delta g(\boldsymbol{x}) - G = 0, \quad \beta \Delta \boldsymbol{u}_k(\boldsymbol{x}) - U_k = 0, \tag{16}$$

where

$$G = \sum_{k=1}^{m} (g(\boldsymbol{x}) - f_k(\boldsymbol{x} - \boldsymbol{u}_k)), \tag{17}$$

$$U_k = (\gamma \sum_{k=1}^{m} \boldsymbol{u}_k + (g(\boldsymbol{x}) - f_k(\boldsymbol{x} - \boldsymbol{u}_k)) \nabla (g(\boldsymbol{x}) - f_k(\boldsymbol{x} - \boldsymbol{u}_k))). \tag{18}$$

We convert eqs.(16) to the diffusion equations

$$\frac{\partial g}{\partial t} = \Delta g(\boldsymbol{x}) - \frac{1}{\alpha} G, \quad \frac{\partial \boldsymbol{u}_k}{\partial t} = \Delta \boldsymbol{u}_k(\boldsymbol{x}) - \frac{1}{\beta} U_k, \tag{19}$$

and discretise them as follows

$$\frac{g^{(n+1)} - g^{(n)}}{\tau} = \boldsymbol{L} g^{(n+1)} - \frac{1}{\alpha} G^{(n)}, \quad \frac{\boldsymbol{u}_k^{(n+1)} - \boldsymbol{u}_k^{(n)}}{\tau} = \boldsymbol{L} \boldsymbol{u}_k^{(n+1)} - \frac{1}{\beta} U_k^{(n)}. \tag{20}$$

Therefore, we obtain the iteration forms

$$(\boldsymbol{I} - \tau \boldsymbol{L}) g^{(n+1)} = g^{(n)} - \frac{\tau}{\alpha} G^{(n)}, \quad (\boldsymbol{I} - \tau \boldsymbol{L}) \boldsymbol{u}_k^{(n+1)} = \boldsymbol{u}_k^{(n)} - \frac{\tau}{\beta} U_k^{(n)}. \tag{21}$$

4 Numerical Method

A 3D discrete Laplace matrix consists of three 1D discrete Laplace matrices $\boldsymbol{D}$ according to the Kronecker product [23,24,25]

$$\boldsymbol{L} = \boldsymbol{D} \otimes \boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{D} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{D}. \tag{22}$$

For fast computation we employ eigenvalue decomposition and the discrete cosine transform [23,20,24]. We can decompose $\boldsymbol{D}$ as $\boldsymbol{D} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top$, where $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, \cdots, \lambda_n)$, with the eigenvalues of $\boldsymbol{D}$,

$$\lambda_i = \frac{4}{h^2}\sin^2\left(\frac{\pi(i-1)}{2(n+1)}\right), \tag{23}$$

for $i = 1, \cdots, n$ and the eigenmatrix is the discrete cosine transform matrix $\boldsymbol{\Phi}$. h means the resolution of spatial derivation, in this paper $h = 1$. Moreover, matrix $\boldsymbol{L}$ is decomposed as

$$\boldsymbol{D} \otimes \boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{D} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top, \tag{24}$$

where $\boldsymbol{U} = (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi} \otimes \boldsymbol{\Phi})$, $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} \otimes \boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{\Lambda} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{\Lambda})$ and $\boldsymbol{I}$ is identity matrix. Therefore, we obtain the equation

$$g^{(n+1)} = \boldsymbol{U}(\boldsymbol{I} - \tau\Sigma)^{-1}\boldsymbol{U}^\top\left(g^{(n)} - \frac{\tau}{\alpha}G\right), \tag{25}$$

$$\boldsymbol{u}^{(n+1)} = \boldsymbol{U}(\boldsymbol{I} - \tau\Sigma)^{-1}\boldsymbol{U}^\top\left(\boldsymbol{u}^{(n)} - \frac{\tau}{\alpha}U_k\right). \tag{26}$$

Equations (25) and (26) are numerically solved using DC(S)T-II [20] for Neumann boundary conditions [24,25].

We need to analyse the convergence of the iterative algorithm by setting up the spectral radius of the iterative matrix. First, we deform eq.(25) and obtain

$$g^{(n+1)} = \boldsymbol{M}\left(1 - \frac{m\tau}{\alpha}\right)g^{(n)} + \boldsymbol{c}, \tag{27}$$

where $\boldsymbol{M} = (\boldsymbol{I} - \tau\boldsymbol{L})^{-1}$, $\boldsymbol{c} = \sum_{k=1}^{m}\frac{\tau}{\alpha}f_k(\boldsymbol{x} - \boldsymbol{u}_k)$. If

$$\max\left(\left|\left(\frac{1}{1 - \tau(\lambda_i + \lambda_j + \lambda_k)}\right)\left(1 - \frac{m\tau}{\alpha}\right)\right|\right) < 1 \tag{28}$$

is satisfied, the iterative calculation converges. We assume the Lagrange multiplier and the time marching parameter $0 < \alpha, \quad 0 < \tau, \quad \tau \ll 1, \quad 0 < 1 - \frac{m\tau}{\alpha}$. Equation.(23) derives $0 \leq (\lambda_i + \lambda_j + \lambda_k) \leq 4\sum_{n=h,w,d}\sin^2(\frac{\pi(n-1)}{2(n+1)})$, where h, w, d equals imags's height, width, depth. When $\frac{1}{1-\tau(\lambda_i+\lambda_j+\lambda_k)}$ is maximised, $\lambda_i + \lambda_j + \lambda_k = 4\sum_{n=h,w,d}\sin^2(\frac{\pi(n-1)}{2(n+1)})$ and $\frac{1}{1-\tau(\lambda_i+\lambda_j+\lambda_k)} > 0$. Therefore, we obtain the relation $\alpha < \frac{m}{\max(\lambda_i+\lambda_j+\lambda_k)}, \quad 0 < \tau < \frac{\alpha}{m}$ as the convergence condition.

Next, we deform eq.(26) and obtain

$$\boldsymbol{u}_k^{(n+1)} = \boldsymbol{M}\left(1 - \frac{(1+\epsilon)\gamma\tau}{\beta}\right)\boldsymbol{u}_k^{(n)} + \boldsymbol{c}, \tag{29}$$

where $\boldsymbol{M} = (\boldsymbol{I} - \tau\boldsymbol{L})^{-1}$, $\boldsymbol{c} = \frac{\tau}{\beta}(\gamma\sum_{j=1,j\neq k}^{m}\boldsymbol{u}_j)$, and ϵ represents the warping term. Similar to eq.(28), we obtain the relation

$$\max\left(\left|\left(\frac{1}{1 - \tau(\lambda_i + \lambda_j + \lambda_k)}\right)\left(1 - \frac{\tau(1+\epsilon)\gamma}{\beta}\right)\right|\right) < 1 \tag{30}$$

Hence of, $\beta < \frac{(1+\epsilon)\gamma}{\max(\lambda_i+\lambda_j+\lambda_k)}, \quad 0 < \tau < \frac{\beta}{(1+\epsilon)\gamma}$ is satisfied, the iteration forms of eq.(21) converges.

Fig. 3. Examples of used data. These figures are the 20th slices of different 3D liver images and are aligned according to their centre of gravity.

Table 1. Used data and their sizes

Region	Size(pixel)
Liver	$84 \times 95 \times 68$
Pancreas	$39 \times 72 \times 27$
Lung(R)	$88 \times 79 \times 91$
Lung(L)	$89 \times 63 \times 94$
Kidney(R)	$31 \times 31 \times 34$
Kidney(L)	$35 \times 32 \times 37$
Heart	$59 \times 70 \times 50$

5 Numerical Examples

To evaluate average images, we define some criteria : warped image error(WIE), flow norm(FN) and flow sum norm (FSN). WIE is defined on each pixel $\boldsymbol{x}$ as

$$\mathrm{WIE}(\boldsymbol{x}) = \|g(\boldsymbol{x}) - f(\boldsymbol{x} - \boldsymbol{u})\|_2. \tag{31}$$

WIE indicates the closeness of the correspondence between the average image g and the input image f. FN and FSN are also defined on each pixel $\boldsymbol{x}$ as

$$\mathrm{FN}(\boldsymbol{x}) = \|\boldsymbol{u}(\boldsymbol{x})\|_2, \quad \mathrm{FSN}(\boldsymbol{x}) = \|\sum_{k=1} \boldsymbol{u}_k(\boldsymbol{x})\|_2. \tag{32}$$

The average image g is located near the median point of images f_k when FSN is small. Moreover, FN shows how far one of the input images lies from average image g.

We tested our algorithm with 3D medical images. First, we need to preprocess the images. We resize the images so that they fit in the surrounding box and align the images using their centres of gravity. Figure 3 shows examples of liver images and Table 1 shows the data used for testing and their image sizes.

In addition to our algorithm, we used two more methods: linear average g_{linear} and eigen average g_{eigen}, given as

$$g_{\mathrm{linear}} = \frac{1}{n} \sum_{k}^{n} f_k, \quad g_{\mathrm{eigen}} = \sum_{i} \lambda_i \boldsymbol{v}_i, \tag{33}$$

Table 2. Parameters used for numerical computation

α	β	γ	τ
10^{-1}	10^{2}	10^{3}	10^{-2}

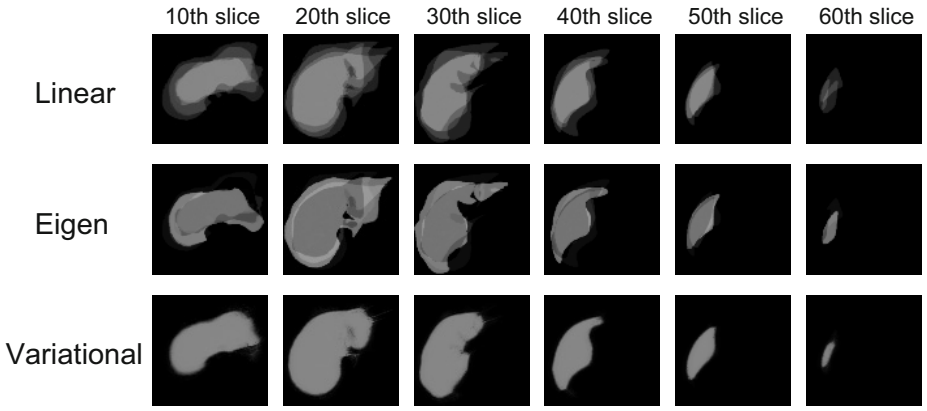


Fig. 4. Mean images of livers. Linear images and eigenimages are blurred. However, the contour of the variational mean image is clear.

where λ_i is the ith eigenvalue and $\boldsymbol{v}_i$ is the correspondent eigenvector [2].

Table 2 shows parameters used for numerical computation. These parameters are selected to counterbalance each term.

Figure 4 shows average liver images. In this figure, the contours of linear images are blurred. Eigenimages are fabricable and have clearer contours than linear images. However, colours of eigenimages are affected by the background colour. On the other hand, the variational average images have clear contours and their original colours. Therefore in this figure, we can see the variational average images have good correspondences to the input livers. Table 3 to Table 9 show the values of WIE. To calculate this criterion, we set zero vectors as the deformation fields for linear averages and eigenaverages. In these tables, the variational average images have smaller WIE values than other methods. Therefore, we can conclude that there are good correspondences between the the variational average images and the input images.

Next, we tested whether the average images lie at the median points of the input images. Table 10 shows FSN for each organ. These values are smaller than the values of FN are in Table 11; hence, we can conclude that the variational average images are located at the centre of the input images.

[2] For a small displacement, we have the relation $f(\boldsymbol{x}+\delta) \cong f(\boldsymbol{x})+a+\nabla f(\boldsymbol{x})$. therefore, $f(\boldsymbol{x})$ is expressed by four principal eigen functions of covariance of f_0 and $E = E(f \otimes f)$, $(f \otimes f)g = (g, f)f$ for inner product $(f, g) = \int_{\mathbb{R}^3} fg\mathrm{d}\boldsymbol{x}$.

Table 3. WIE among liver images. The error of the variational method is smallest.

Method	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Variational	2.33	6.83	2.63	8.88	2.35	7.08	2.19	6.60	2.31	7.07
Linear	8.34	20.50	9.04	22.16	9.50	23.35	8.11	19.91	11.43	27.30
Eigen(1st)	6.83	17.07	7.85	20.34	13.75	33.26	8.44	21.74	13.20	31.69
Eigen(1st-4th)	12.29	32.19	10.44	27.25	15.10	39.22	4.85	11.94	14.25	39.38

Table 4. WIE among pancreas images

Method	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Variational	1.19	4.91	1.43	6.29	1.34	5.69	1.30	5.45	1.26	5.30
Linear	6.50	17.66	7.51	20.41	7.61	20.70	7.37	20.06	6.63	18.01
Eigen(1st)	4.11	11.39	6.23	18.86	9.20	26.05	9.81	27.72	9.58	28.50
Eigen(1st-4th)	5.64	15.19	8.77	24.22	13.37	36.97	13.07	36.46	10.08	28.50

Table 5. WIE among right lung images

Method	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Variational	1.62	4.33	1.34	3.84	1.70	4.20	2.01	4.39	1.93	4.90
Linear	8.23	15.56	6.80	12.64	8.32	15.34	7.12	12.97	8.85	16.19
Eigen(1st)	7.85	13.64	7.78	12.49	12.32	19.74	6.30	11.41	14.12	22.45
Eigen(1st-4th)	8.75	13.76	12.77	21.53	10.91	17.26	10.28	17.34	16.27	26.50

Table 6. WIE among left lung images

Method	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Variational	2.01	5.52	1.67	4.71	2.02	4.88	1.96	4.66	2.18	6.01
Linear	8.99	16.42	7.92	14.42	8.44	14.91	7.34	13.22	9.56	17.34
Eigen(1st)	7.94	14.30	7.92	13.26	11.89	19.74	6.53	12.16	14.20	23.57
Eigen(1st-4th)	11.91	20.83	7.64	12.66	15.98	27.78	10.89	19.62	13.10	22.20

Table 7. WIE among right kidney images

Method	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Variational	1.79	6.37	1.57	5.80	1.78	6.06	1.68	6.27	1.86	6.50
Linear	11.91	23.02	10.41	19.69	14.20	27.19	10.21	19.47	14.00	26.71
Eigen(1st)	8.78	17.00	12.13	24.25	13.22	26.34	12.22	24.60	21.37	40.41
Eigen(1st-4th)	11.94	23.17	8.95	16.41	21.50	41.48	14.70	30.61	22.66	46.91

Table 8. WIE among left kidney images

Method	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Variational	2.10	6.97	1.82	6.27	1.86	6.60	1.84	6.45	2.09	7.00
Linear	12.73	25.61	10.39	21.07	10.13	20.41	11.05	22.39	12.86	25.86
Eigen(1st)	6.80	15.58	14.69	31.77	7.73	17.96	12.79	28.45	19.65	39.37
Eigen(1st-4th)	9.15	17.83	21.60	42.51	13.56	27.27	16.31	31.93	19.34	37.04

Table 9. WIE among heart images

Method	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Variational	1.11	4.47	1.01	4.47	1.03	4.31	0.97	4.17	1.00	4.07
Linear	12.01	26.18	9.85	21.63	9.79	21.40	9.42	20.62	10.62	23.26
Eigen(1st)	13.35	22.95	11.82	19.34	12.80	21.80	14.46	25.09	18.06	31.85
Eigen(1st-4th)	17.02	30.65	14.02	25.02	18.02	34.36	11.16	18.40	19.64	39.34

Table 10. FSN for each organ images

	Mean	SD	Max
Liver	0.02	0.07	2.78
Pancreas	0.01	0.05	1.23
Lung(R)	0.01	0.03	1.35
Lung(L)	0.01	0.03	0.82
Kidney(R)	0.02	0.07	1.08
Kidney(L)	0.02	0.07	1.52
Heart	0.01	0.04	1.67

Table 11. FN values for each organ image. For example, for the case of the liver Data 2 has the largest mean and largest SD. Therefore, Data 2 is furthest from the average liver.

Region	Data 1		Data 2		Data 3		Data 4		Data 5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Liver	0.39	0.62	0.42	0.85	0.40	0.72	0.33	0.58	0.38	0.69
Pancreas	0.12	0.38	0.15	0.45	0.14	0.41	0.14	0.43	0.13	0.41
Lung(R)	0.43	0.78	0.28	0.56	0.44	0.81	0.37	0.67	0.55	1.05
Lung(L)	0.40	0.82	0.32	0.64	0.38	0.77	0.31	0.63	0.44	0.88
Kidney(R)	0.30	0.44	0.22	0.32	0.37	0.57	0.21	0.33	0.43	0.66
Kidney(L)	0.27	0.54	0.19	0.38	0.17	0.35	0.18	0.35	0.27	0.55
Heart	0.37	0.75	0.26	0.55	0.27	0.52	0.26	0.55	0.35	0.65

Since the heart images are not temporally synchronised, the average image is not considered phase.

Our method found correspondences between the variational average image and input images. That is, image registration between each of input images and the average image is achieved. Moreover, we can conclude that the variational average image is located at the median point of input images from the fact that the sum of the vectors of the flow fields on the average image is smaller than each vector. Therefore, we could calculate the average image that we modeled.

6 Conclusion

We defined the variational average image based on variational registration and modeled the energy function to calculate it. In our experiments, we confirmed that our method can generate the variational average image and the deformation fields. Moreover, the variational average images are characterised by the median point of the input images. This information was obtained from the deformation fields.

References

1. Grigorescu, C., Petkov, N.: Distance Sets for Shape Filters and Shape Recognition. IEEE Trans. IP 12, 1274–1286 (2003)
2. Hajnal, J.V., Hill, D.L.G., Hawkes, D.J.: Medical Image Registration. Phys. Med. Biol. 46, R1–R45 (2001)
3. Fischer, B., Modersitzki, J.: Ill-posed Medicine - An Introduction to Image Registration. Inverse Prob. 24, 1–17 (2008)
4. Rumpf, M., Wirth, B.: A Nonlinear Elastic Shape Averaging Approach. SIAM J. Imaging Sci. 2, 800–833 (2009)
5. Sebastian, T.B., Klein, P.N., Kimia, B.B.: On Aligning Curves. IEEE Trans. PAMI 25, 116–125 (2003)
6. Baeza-Yates, R., Valiente, G.: An Image Similarity Measure Based on Graph Matching. In: Proc. SPIRE 2000, pp. 8–38 (2000)
7. Riesen, K., Bunke, H.: Approximate Graph Edit Distance Computation by Means of Bipartite Graph Matching. Image and Vision Comput. 27, 950–959 (2009)
8. Arrate, F., Ratnanather, J.T., Younes, L.: Diffeomorphic Active Contours. SIAM J. Imaging Sci. 3, 176–198 (2010)
9. Sharon, E., Mumford, D.: 2D-Shape Analysis using Conformal Mapping. IJCV 70, 55–75 (2006)
10. Najman, L.: The 4D heart database, `http://www.laurentnajman.org/heart/`
11. Tănase, M., Veltkamp, R.C., Haverkort, H.J.: Multiple Polyline to Polygon Matching. In: Deng, X., Du, D.-Z. (eds.) ISAAC 2005. LNCS, vol. 3827, pp. 60–70. Springer, Heidelberg (2005)
12. Arkin, E.M., Chew, L.P., Huttenlocher, D.P., Kedem, K., Mitchell, J.S.B.: An Efficiently Computable Metric for Comparing Polygonal Shapes. IEEE Trans. PAMI 13, 209–216 (1991)
13. Stegmann, M.B., Gomez, D.D.: A Brief Introduction to Statistical Shape Analysis. Informatics and Mathematical Modelling. Technical University of Denmark, p. 15 (2002)
14. Srivastava, A., Joshi, S.H., Mio, W., Liu, X.: Statistical Shape Analysis: Clustering, Learning, and Testing. IEEE Trans. PAMI 27, 590–602 (2005)

15. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow. Artif. Intell. 17, 185–204 (1981)
16. Modersitzki, J.: Numerical Methods for Image Registration. OUP (2004)
17. Rumpf, M., Wirth, B.: An Elasticity-Based Covariance Analysis of Shapes. IJCV 92, 281–295 (2011)
18. Wirth, B., Bar, L., Rumpf, M., Sapiro, G.: A Continuum Mechanical Approach to Geodesics in Shape Space. IJCV 93, 293–318 (2011)
19. Berkels, B., Linkmann, G., Rumpf, M.: An $SL(2)$ Invariant Shape Median. JMIV 37, 85–97 (2010)
20. Strang, G., Nguyen, T.: Wavelets and Filter Banks. Wellesley-Cambridge Press (1996)
21. Beg, M.F., Miller, M.I., Trouvé, A., Younes, L.: Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms. IJCV 61(2), 139–157 (2005)
22. Garcin, L., Rangarajan, A., Younes, L.: Non Rigid Registration of Shapes via Diffeomorphic Point Matching and Clustering. In: Proc. ICIP 2004, pp. 3299–3302. IEEE (2004)
23. Strang, G.: Computational Science and Engineering. Wellesley-Cambridge Press (2007)
24. Demmel, J.W.: Applied Numerical Linear Algebra. SIAM (1997)
25. Varga, R.S.: Matrix Iteration Analysis, 2nd edn. Springer (2000)
26. Vialard, F., Risser, L., Rueckert, D., Cotter, C.J.: Diffeomorphic 3D Image Registration via Geodesic Shooting using an Efficient Adjoint Calculation. IJCV 97, 229–241 (2012)

A Hierarchical Approach to Optimal Transport

Bernhard Schmitzer and Christoph Schnörr

Heidelberg University

Abstract. A significant class of variational models in connection with matching general data structures and comparison of metric measure spaces, lead to computationally intensive dense linear assignment and mass transportation problems. To accelerate the computation we present an extension of the auction algorithm that exploits the regularity of the otherwise arbitrary cost function. The algorithm only takes into account a sparse subset of possible assignment pairs while still guaranteeing global optimality of the solution. These subsets are determined by a multiscale approach together with a hierarchical consistency check in order to solve problems at successively finer scales. While the theoretical worst-case complexity is limited, the average-case complexity observed for a variety of realistic experimental scenarios yields a significant gain in computation time that increases with the problem size.

1 Overview and Contribution

Overview. The linear assignment problem (LAP) and, more general, optimal transport (OT) can be considered fundamental tools in computer vision and mathematical image processing and their properties have been thoroughly examined [10,12]. For optimal transport between smooth distributions on $\mathbb{R}^n$ with convex cost functions, in particular the squared Euclidean distance, specialized solution methods are available [5,6]. However, this is a rather restricted class of scenarios and the proposed ODE/PDE solutions are very involved numerically. For the LAP there are two classical algorithms: the Hungarian method [7] and the auction algorithm [1], which is apt for parallelization [2] and can be generalized to OT [4]. The evolution of the auction algorithms has also sparked investigation of more general min-cost flow problems [3].

Despite all its merits as a metric on measures [8], optimal transport has the disadvantage of being computationally considerably more expensive than simple comparisons like the L_1 distance. Thus, equivalent, yet more easily computable metrics [11], thresholded cost functions [9] or tangent space approximations [13] have been proposed.

The mentioned classical algorithms do not take into account any particular structure of the cost function, whereas for virtually all practical problems, the cost functions are far from arbitrary, but usually obey some regularity criterion. Secondly, said algorithms become very slow for large, dense problems. However many natural problems are a priori dense, i.e. any conceivable mass assignment is theoretically possible (e.g. linear shape matching relaxations discussed in [8]).

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 452–464, 2013.

The regularity of the cost function can sometimes be exploited to devise heuristics that aim at ruling out very unlikely (mass) assignments, to reduce the problem size beforehand. Yet, in general it is very hard to come up with a *simple* in-/exclusion rule, that can both rule out a substantial fraction of possible assignments, so as to significantly reduce the problem size, and, at the same time guarantee, that the global optimum of the full problem will not be lost.

Contribution. In this paper we present a modification of the auction algorithm that **(a)** can exploit any available heuristic for estimating a relevant sparse subset of assignments. However, it will at the same time be **(b)** guaranteed to find a globally optimal solution of the underlying dense problem by hierarchically checking for violated constraints of the dual problem, which relies on regularity of the cost function. In fact the hierarchical structure will lend itself to **(c)** provide a reasonable sparsity estimate for the problem at hand by a multiscale approach. Although some additional steps are required as compared to the standard auction algorithms, we show that **(d)** the worst case complexity overhead of our proposed method is limited. At the same time **(e)** we demonstrate with realistic examples, that the 'typical' problem complexity for practical setups is significantly reduced (see Fig. 1). In fact, the gain in computation time grows with problem size. This will enable application of the auction algorithm to problem sizes that were unfeasible so far and which due to their more general structure cannot be solved by PDE methods.

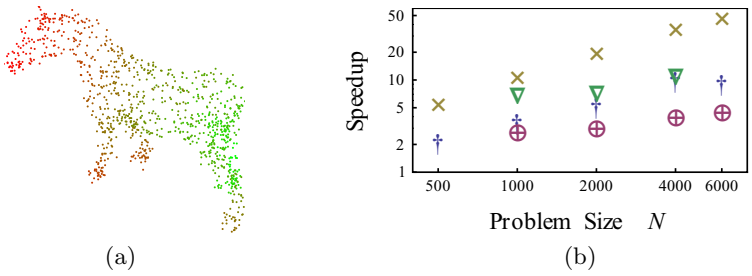


Fig. 1. (a) Illustration of experimental scenario "mesh": mass distributions on point clouds sampled from manifolds, cost function given by point distance in underlying geodesic metric. (b) Ratio of runtimes of standard auction algorithm and our proposed extension for various scenarios (see Sect. 6) and problem sizes N. †: P2H, $\oplus$: P3H, $\times$: grid, ∇: mesh. P2H-P1, P2I and P2H-LB perform essentially like P2H. N gives the number of points per point cloud or vertices per grid. For $N = 6000$ (i.e. $N^2 = 3.6 \cdot 10^7$ potential assignment pairs) the observed speedup ranges between 4.6 and 48, consistently increasing with problem size.

In Section 2 we will recall the definitions of LAP and OT. Section 3 reviews the auction algorithm for the LAP and discusses the extension to OT. In Section 4 we present our proposed method. A comparative worst case complexity analysis is given in Sect. 5, before demonstrating with realistic experiments in Sect. 6 the significant benefit of the proposed extensions. The paper concludes in Sect. 7.

2 Linear Assignment Problem and Optimal Transport

The Linear Assignment Problem. For two finite sets X, Y and a cost function $c : X \times Y \to \mathbb{R}_+ \cup \{\infty\}$ let $\mathcal{N} = \{(x, y) \in X \times Y : c(x, y) < \infty\}$. We call $\mathcal{N}$ the set of neighbours and write $\mathcal{N}(x) = \{y \in Y : (x, y) \in \mathcal{N}\}$ and similarly $\mathcal{N}(y)$. We will refer to a subset $S \subset X \times Y$ as *assignment* [4] if it satisfies

(a) $S \subseteq \mathcal{N}$,
(b) $|\{(x', y') \in S : y' = y\}| \leq 1 \,\forall\, y \in Y$,
(c) $|\{(x', y') \in S : x' = x\}| \leq 1 \,\forall\, x \in X$.

An assignment is called *complete* if for any $x \in X$ there is a $y \in Y$ such that $(x, y) \in S$ and vice versa.

The LAP [10] is then readily stated as

$$\min \left\{ \sum_{(x,y)\in S} c(x, y) : S \text{ is a complete assignment between } X \text{ and } Y \right\} . \quad (1a)$$

The corresponding dual problem is

$$\max \left\{ \sum_x \alpha(x) + \sum_y \beta(y) : \alpha(x) + \beta(y) \leq c(x, y) \right\} . \quad (1b)$$

Note that for any fixed β the corresponding best choice of α is given by

$$\alpha(x) = \min_y c(x, y) - \beta(y) . \quad (2)$$

It is a well known result that for any optimal assignment S of the primal problem (1a) and optimal (α, β) of the dual problem (1b) one finds

$$(x, y) \in S \Rightarrow \alpha(x) + \beta(y) = c(x, y) . \quad (3)$$

Optimal Transport. For two finite sets X, Y let $\mu_X \in \mathbb{R}^{|X|}, \mu_Y \in \mathbb{R}^{|Y|}$ be two vectors with non-negative entries and equal sum of entries $\sum_x \mu_X(x) = \sum_y \mu_Y(y)$, indicating mass distributions on X, Y. Here, $c : X \times Y \to \mathbb{R} \cup \{\infty\}$ is a cost function, giving the cost to transport one unit of mass between elements of the sets.

The optimal transport problem [12] can then be written as

$$\inf \left\{ \sum_{x,y} c(x, y)\, \mu(x, y) : \mu \geq 0, \sum_y \mu(x, y) = \mu_X(x), \sum_x \mu(x, y) = \mu_Y(y) \right\} \quad (4a)$$

where a μ is dubbed a *coupling*. The respective dual is given by

$$\sup \left\{ \sum_x \alpha(x)\, \mu_X(x) + \sum_y \beta(y)\, \mu_Y(y) : \alpha(x) + \beta(y) \leq c(x, y) \right\} . \quad (4b)$$

Analogous to the primal-dual relation of the LAP (3) one finds for optimal transport: for any optimal μ of primal (4a) and (α, β) of dual (4b) have

$$\mu(x, y) > 0 \Rightarrow \alpha(x) + \beta(y) = c(x, y) . \quad (5)$$

3 The Auction Algorithm

The Auction Algorithm for the Assignment Problem. We now recall the description of the auction algorithm for the LAP from [4, Sect. 2]. Note that we flipped the signs relative to the original presentation. Thus in the following the comparison to an auction is no longer very intuitive (the lowest bid gets accepted). However this makes the algorithm compatible with the usual notion of optimal transport as presented in Sect. 2.

The main loop of the algorithm is divided into two phases: *bidding* and *assignment*. During the bidding phase elements of X locally determine their most suitable assignment partner in Y and propose a corresponding dual variable change. After that, during the assignment phase, for each $y \in Y$ the best proposed dual variable change is implemented. Different x do not interact during the bidding phase and neither do different y during the assignment phase. Thus both stages can be easily parallelized.

The state of the algorithm is represented by an assignment S and dual variable β. The corresponding α is held implicitly via (2). The algorithm is initialized with the empty assignment $S = \emptyset$ and some arbitrary β. A key property of the auction algorithm is, that condition (3) does not hold strictly throughout the iterations. Instead at any stage during the algorithm, for any $(x, y) \in S$ the weaker condition $\alpha(x) + \beta(y) \geq c(x, y) + \varepsilon$ is satisfied, where ε is some positive parameter. Positivity of ϵ is essential for convergence of the algorithm. However, as long as $\varepsilon < \Delta c/|X|$ the resulting complete S is guaranteed to solve (1a), where Δc is the smallest difference between two non-equal values of c.

Bidding Phase. For every $x \in X$ that is unassigned under S:

Compute the corresponding value of $\alpha(x)$ as given by (2):

$$\alpha(x) = \min_{y \in \mathcal{N}(x)} c(x, y) - \beta(y) \tag{6}$$

and find a minimizer y^*. Determine also the slack of the second 'nearest' constraint:

$$\alpha'(x) = \min_{y \in \mathcal{N}(x) \setminus \{y^*\}} c(x, y) - \beta(y) \tag{7}$$

Then element $x \in X$ bids for element $y^* \in Y$ with value

$$b_{xy^*} = c(x, y^*) - \alpha'(x) - \varepsilon\,. \tag{8}$$

Assignment Phase. For each $y \in Y$ let $P(y)$ be the set of $x \in X$ from which y received a bid in the bidding phase of the iteration. If $P(y)$ is nonempty, decrease $\beta(y)$ to the lowest bid

$$\beta(y) := \min_{x \in P(y)} b_{xy}\ , \tag{9}$$

remove from the assignment S any pair (x, y) (if one exists), and add to S the pair (x^*, y) where x^* is some element in $P(y)$ attaining the minimum in (9). If $P(y)$ is empty, $\beta(y)$ is left unchanged.

Repeat the two stages until S is complete.

The Auction Algorithm for Optimal Transport. In principle any optimal transport problem with integer mass distributions can be translated into an LAP by introducing a 'mass-atom' and splitting up each node $x \in X$, $y \in Y$ into multiple copies, depending on how many atoms fit into $\mu_X(x), \mu_Y(y)$. By applying suitable data structures this splitting can be made implicit and the auction algorithm does not actually need to handle each mass atom separately. For example, *assignments* S will be replaced by *couplings* μ. Also, some modifications in the bidding process are advisable to prevent inefficient competition between atoms originating from the same elements of X.

Such a reformulation is given in [4, Sect. 4], which we cannot repeat here, due to space limitations. Instead we will briefly comment on the modifications which are relevant for our proposed extensions to be discussed in the next section.

In the generalized algorithm, due to the splitting, the dual variable β need not be constant 'within' every y. Thus, there is a dual variable $\tilde{\beta}$ for every pair (x, y) and one variable $\tilde{\beta}(\lozenge, y)$ for mass atoms in y which have not yet received a bid. A dual variable β can be obtained by

$$\beta(y) = \begin{cases} \max_{x' \in X \colon \mu(x',y)>0} \beta(x', y) & \text{if } \sum_{x'} \mu(x', y) = \mu_Y(y) \\ \beta(\lozenge, y) & \text{else} \end{cases} .$$

In the bidding phase, any x with $\sum_y \mu(x, y) < \mu_X(x)$ can submit bids to multiple y simultaneously. To determine the bid recipients, consider the set

$$\begin{aligned} \Pi(x) = \; & \{c(x, y) - \beta(x', y) \,|\, y \in \mathcal{N}(x),\, x' \neq x \text{ and } x' \in \mathcal{N}(y),\, \mu(x', y) > 0\} \\ & \cup \{c(x, y) - \beta(\lozenge, y) \,|\, y \in \mathcal{N}(x),\, \textstyle\sum_{x'} \mu(x', y) < \mu_Y(y)\} \end{aligned} \tag{10}$$

and assume that the entries are arranged in ascending order, i.e. we have

$$\Pi(x) = \left\{ c(x, y_1) - \beta(x'_1, y_1), \ldots, c(x, y_{|\Pi(x)|}) - \beta(x'_{|\Pi(x)|}, y_{|\Pi(x)|}) \right\} \tag{11}$$

with $c(x, y_i) - \beta(x'_i, y_i) \leq c(x, y_{i+1}) - \beta(x'_{i+1}, y_{i+1})$, for all $i = 1, \ldots, |\Pi(x)| - 1$, where by abuse of notation we allow $x'_i = \lozenge$ for some i.

Values (6) and (7) are the first two entries of this list in the LAP case, for determining the bids in a general OT problem, more than two entries might be relevant. Depending on the mass distributions μ_X, μ_Y, one will determine an integer $m > 1$ such that the equivalent of (7) is given by

$$\alpha'(x) = c(x, y_m) - \beta(x'_m, y_m) \,. \tag{12}$$

For a complete description of the algorithm we refer the reader to [2].

4 A Hierarchical Multiscale Approach to Optimal Transport

Motivation. Obviously both algorithms will perform faster on sparse problems, where the set of neighbours $\mathcal{N}$ is small. For example, the creation of the list

(10) will require much fewer queries. In practice however, many problems are dense and a priori any assignment (x, y) could be possible. For some applications one might be able to devise good heuristics to exclude certain pairs, which are unlikely part of an optimal solution. But due to the combinatorial structure of the underlying LAP it is in general hard to rule out a significant amount of potential assignments and yet guarantee that the global optimum of the full problem will be attained.

In most practical problems the sets X and Y are equipped with some additional structure and notion of *closeness or similarity* which is also represented in the cost function. If x and y are close to x' and y' respectively, then we expect $|c(x, y) - c(x', y')|$ to be somehow bounded. The details of this boundedness condition (e.g. Lipschitz continuity) may depend on the problem at hand and are not crucial for the applicability of the scheme to be discussed.

We will now present a sparse/dense hybrid variant of the auction algorithm, that can be initialized with a good heuristic guess for the subset of relevant assignment pairs and will benefit from the sparsity of this set and the additional available structure of X, Y and c. Yet it will be guaranteed to find a globally optimal assignment or coupling measure (Proposition 1). This hybrid variant can then be used in a multiscale scheme, that successively generates optimal couplings at finer and finer scales of the problem, using the results from the coarser scales for efficiently solving the finer scales. A central concept of this algorithm are hierarchical partitions, to be introduced next.

Hierarchical Partitions. Let $\mathcal{A}_1 \subset 2^X$ be a partition of X, such that any two elements x, x' of one partition cell are considered to be 'close' in the aforementioned sense. Then let $\mathcal{A}_2$ be another (coarser) partition that is compatible with $\mathcal{A}_1$ in the sense that any element $a \in \mathcal{A}_2$ can be written as the union of some cells of $\mathcal{A}_1$. This coarsening can be repeated multiple times, each time ensuring that elements in the same cell satisfy some (scale-adjusted) closeness criterion. The resulting structure implies a directed tree graph with vertex set $\mathcal{A} = \bigcup_{i=0}^{g-1} \mathcal{A}_i$ where $\mathcal{A}_0 = \{\{x\} : x \in X\}$ is the set of singletons of X and g is the depth of the hierarchy. For $0 \leq i < g$ we say $a' \in \mathcal{A}_i$ is a child of $a \in \mathcal{A}_{i+1}$ (and a is parent of a') and write $a' \in \mathrm{ch}(a), a = \mathrm{pa}(a')$ if $a' \subset a$. We call this a *hierarchical partition* of X.

Analogous we let $\mathcal{B}$ be a hierarchical partition of Y and w.l.o.g. assume that $\mathcal{A}$ and $\mathcal{B}$ have the same depth.

Now for a given dual variable α define the extension $\hat{\alpha}$ onto the whole hierarchical partition by

$$\hat{\alpha}(a) = \max_{x \in a} \alpha(x) = \begin{cases} \alpha(x) & \text{if } a = \{x\} \in \mathcal{A}_0 \text{ for some } x \\ \max_{a' \in \mathrm{ch}(a)} \hat{\alpha}(a') & \text{if } a \in \mathcal{A}_i \text{ for some } i > 0 \end{cases} \tag{13}$$

and analogous for β and $\hat{\beta}$.

Similarly define an extension $\hat{c}$ of c onto $\mathcal{A} \times \mathcal{B}$ via

$$\hat{c}(a, b) = \min_{x \in a, y \in b} c(x, y)\,. \tag{14}$$

We now define an extension of the dual constraints of (1b,4b) to coarser scales: we will refer to the following set of inequalities as *dual constraints of generation* n:

$$\hat{\alpha}(a) + \hat{\beta}(b) \leq \hat{c}(a,b) \,\forall\, (a,b) \in \mathcal{A}_n \times \mathcal{B}_n \tag{15}$$

Obviously if the dual constraints of generation n hold for some extended $\hat{\alpha}, \hat{\beta}$ and $\hat{c}$, then so will the constraints at all generations $n' < n$. For $n = 0$ these constraints are those of the original optimal transport problem. The requirement that elements within the same partition cell of any generation should be close, will ensure, that the dual constraints of generation n will not be a lot tighter than those of generation $n-1$.

A Sparse/Dense Hybrid Variant of the Auction Algorithm. Consider a feasible optimal transport problem between (X, μ_X) and (Y, μ_Y) with cost function c. Let $\hat{\mathcal{N}} \subset X \times Y$ such that $(x,y) \in \hat{\mathcal{N}} \Rightarrow c(x,y) < \infty$. However not necessarily $c(x,y) < \infty \Rightarrow (x,y) \in \hat{\mathcal{N}}$, i.e. we might start with a set of neighbours which is smaller than the maximally possible one. We now give an algorithm that will run on a given submaximal neighbour set $\hat{\mathcal{N}}$, but detect if some $(x,y) \in \mathcal{N}$ might have to be considered as part of an assignment and extend $\hat{\mathcal{N}}$ accordingly if necessary. The *bidding* and *assignment* phases will work just as in the standard auction algorithms, Sect. 3, with $\hat{\mathcal{N}}$ in place of $\mathcal{N}$. But there will be an additional *consistency check* step in between:

Consistency Check Phase. Let $\hat{\alpha}'$ be the hierarchical extension of α' as defined in (7,12) and $\hat{\beta}$ the hierarchical extension of $\beta(\cdot)$. Then start with checking whether $\hat{c}(a,b) - \hat{\beta}(b) \geq \hat{\alpha}'(a)$ for all $a \in \mathcal{A}_n, b \in \mathcal{B}_n$ at some generation $n > 0$.

If a checked inequality holds, then certainly $c(x,y) - \beta(x',y) \geq \alpha'(x)$ for all $x \in a, y \in b, x' \in X$ and thus no $y \in b$ could lead to a different bid for $x \in a$ if $(x,y) \in \hat{\mathcal{N}}$ during the *bidding phase*, since these potential candidates would appear further behind in the ordered list $\mathit{\Pi}(x)$, (11).

If a checked inequality $\hat{c}(a,b) - \hat{\beta}(b) \geq \hat{\alpha}'(a)$ is found to be violated, check on a finer level: $\hat{c}(a',b') - \hat{\beta}(b') \geq \hat{\alpha}'(a')$ for $a' \in \mathrm{ch}(a), b' \in \mathrm{ch}(b)$. Recursively continue this process until either all inequalities hold, or at generation 0 a candidate $c(x,y) - \beta(y) < \alpha'(x)$ is found. If for such a candidate $(x,y) \notin \hat{\mathcal{N}}$, then update $\hat{\mathcal{N}} := \hat{\mathcal{N}} \cup \{(x,y)\}$ and list x for rebidding.

After the consistency check, reevaluate the bidding phase for all listed x.

Proposition 1. *The sparse/dense hybrid auction algorithm, initialized with some non-maximal neighbourhood set $\hat{\mathcal{N}}$, such that the problem constrained to $\hat{\mathcal{N}}$ is still feasible, will converge to a globally optimal coupling μ under the same conditions as the dense algorithm variant.*

The proof is rather simple and thus for lack of space will be postponed to a more thorough article on the subject. It hinges on the fact, that elements in the list $\mathit{\Pi}(x)$, Eq. (11), that appear beyond position m (which determines the value of α', see (12)), do not influence the process of the algorithm.

It should be noted, that this modification preserves the parallel structure of the algorithm. Bidding and assignment work as before and the tree structure of the successive hierarchical consistency checks allows for distribution of the consistency evaluation onto multiple processors.

A Hierarchical Multiscale Approach to Optimal Transport. The hybrid variant will give a globally optimal coupling μ for valid initializations of $\hat{\mathcal{N}}$ and usually require far less queries than a naïve dense algorithm, if the initial $\hat{\mathcal{N}}$ is chosen well and c is 'sufficiently regular' within the partition cells. For specific problems one may devise good heuristics for such an initial guess. Now we want to propose a *generic scheme, that works in principle for any problem.* Its practicality will be evaluated in Sect. 6. Again, to save space, we can only so much as give a sketch and must omit proofs for now.

For an optimal transport problem the coarsened problem at generation n is defined by

$$\begin{aligned} &\inf \sum_{(a,b)\in\mathcal{A}_n\times\mathcal{B}_n} \hat{c}(a,b)\,\hat{\mu}(a,b) \text{ subject to} \\ &\hat{\mu}\geq 0,\ \sum_b \hat{\mu}(a,b)=\sum_{x\in a}\mu_X(x),\ \sum_a \hat{\mu}(a,b)=\sum_{y\in b}\mu_Y(y)\,. \end{aligned} \tag{16}$$

Denote by D_n its optimal value.

Let Δc_n be an upper bound on the variation of c within one partition cell of $\mathcal{A}_n\times\mathcal{B}_n$, i.e. $\hat{c}(a,b)\leq c(x,y)\leq \hat{c}(a,b)+\Delta c_n$ for $(a,b)\in\mathcal{A}_n\times\mathcal{B}_n$, $(x,y)\in a\times b$. In addition, any feasible $\hat{\mu}$ of the coarsened problem at some generation n does induce feasible couplings on lower generations. Let $\hat{\mu}'$ be some feasible coupling of generation $n-1$ induced by an optimizer $\hat{\mu}$ of generation n, then one can easily proof that

$$D_n\leq D_{n-1}\leq \sum_{(a,b)\in\mathcal{A}_{n-1}\times\mathcal{B}_{n-1}} \hat{c}(a,b)\,\hat{\mu}'(a,b)\leq D_n+\Delta c_n\cdot M\,,$$

where $M=\sum_x \mu_X(x)$. Thus, solving the problem of generation n not only provides a bounded interval for D_{n-1} but also gives a feasible candidate for the problem of generation $n-1$ which is at most suboptimal by a margin $\Delta c_n\cdot M$.

Since c is supposed to be regular in some sense and partitions are to be chosen according to the closeness structure on X and Y, we can assume, that Δc_n is usually small compared to the fluctuations of c throughout the whole coupling space and that, thus, this bound is of actual practical value.

Also, it seems natural, to pick the support of $\hat{\mu}'$ as initial guess for $\hat{\mathcal{N}}$, when solving the refined problem with the hybrid algorithm. Obviously the restriction to $\hat{\mathcal{N}}$ keeps the problem feasible, since it allows $\hat{\mu}'$.

Thus, in short, instead of directly solving the problem at generation 0, we start at some coarser scale n, where the problem is small enough for direct dense solution. Then we use the obtained minimizers to recursively solve the problem at finer scales, each time producing an initial guess for the sparse support subset.

5 Complexity Analysis

We will first give the *worst case complexity analysis of the auction algorithm* for the dense LAP with $\mathcal{N} = X \times Y$, $|X| = |Y|$. It can be considered a special case of a class of min-cost flow algorithms presented in [3]. From [3, Lemma 5] we can see that the number of bids submitted per source is $\mathcal{O}(|X| \cdot C)$ where

$$C = \max_{x,y} c(x,y) - \min_{x,y} c(x,y)\,.$$

From the description in Sect. 3 we can see that cost of one bid for a given source is of order $\mathcal{O}(|\mathcal{X}|)$, i.e. scanning every possible assignment partner once. This already incorporates the costs of bid acceptance at one sink, since at most one bid is accepted per submitted bid. Hence the total worst case complexity of the algorithm is $\mathcal{O}(|X|^3 \cdot C)$.

The *extension to the sparse/dense hybrid variant* requires several additional steps, of which we must estimate the worst case costs. In a worst case scenario any possible link will be added to $\hat{\mathcal{N}}$, i.e. $\hat{\mathcal{N}} = X \times Y$, as in the full problem. Let p be an upper bound on the number of elements in one partition cell at any generation of the hierarchical partitions and let g be the number of generations. Then per bid submission at most $\mathcal{O}(g)$ steps are required to compute the extension $\hat{\alpha}'$ and at most $\mathcal{O}(p \cdot g)$ per reception to update $\hat{\beta}$. There will be of the order $\mathcal{O}(|\mathcal{B}|)$ hierarchical constraints to be tested per bid. Thus for one bid we get costs of the order $\mathcal{O}(|X| + g \cdot (p+1) + |\mathcal{B}|)$, resulting in a total worst case complexity of $\mathcal{O}\big(|X|^2 \cdot C \cdot (|X| + g \cdot (p+1) + g \cdot |\mathcal{B}|)\big)$. In the worst case, after the consistency phase, the bidding phase needs to be rerun completely. However, this only amounts to a constant factor 2 in the number of steps.

If the *hierarchical partitions* satisfy a relation like $|\mathcal{B}_{n+1}| \leq |\mathcal{B}_n| \cdot q$ for some $q \in]0,1[$ then $|\mathcal{B}| \leq \sum_{k=0}^{g-1} |X| q^k < |X|/(1-q)$. For octrees one has for example $q = 1/8$. Also, usually $g, p \ll |X|$, for example $p \approx |\mathcal{A}_{n+1}|/|\mathcal{A}_n| \approx 1/q$ ($= 8$ for octrees) and $g = \mathcal{O}\big(\log(|X|/|\mathcal{A}_{g-1}|)/\log(1/q)\big)$ where $\mathcal{A}_{g-1}$ would be the coarsest generation of the hierarchical partition. Thus, the complexity of the hybrid variant is usually dominated by the last term, which yields $\mathcal{O}\big(|X|^3 \cdot C \cdot g/(1-q)\big)$. Hence, the overhead scales with a constant factor $(1-q)^{-1}$, depending on the hierarchy structure, and a term logarithmic in $|X|$ which accounts for the hierarchy resolution.

In principle the algorithm presented in [3] can also be used to solve the general optimal transport problem, resulting in a similar complexity bound. The variant referred to in Sect. 3 has a much higher worst case complexity but tends to perform faster in practice due to increased resistance to a phenomenon dubbed *price haggling* [3]. This means that the additional steps required by our hybrid variant are of little significance in the worst case, yet are very useful in the 'typical' case, as demonstrated in the next section.

In practice runtime of the auction algorithms does exhibit a strong sensitivity to C. This can be remedied by a method called *ε-scaling* [3] which can be shown to replace the factor C by $\log(|X| \cdot C)$ in the complexity estimates. Also, this method is compatible with our presented additions.

6 Experiments

In the previous section we have considered the *theoretical worst case complexity* of the auction algorithm and its hybrid extension. It is however very hard to obtain a theoretical estimate for the *'typical' complexity*. Thus, for demonstrating the benefit of the augmented algorithm we need to rely on numerical experiments.

Implementation Details. For evaluation we implemented the auction algorithm in `c++` with sparse data structures. The hybrid variant is based on the same implementation, extended by the consistency phase, to obtain a meaningful performance comparison. All mass distributions were picked to be integer and the cost functions were truncated to a fine discrete grid of equidistant values. To get practically relevant solving times, we used a very rudimentary form of ε-scaling, in which the problem is repeatedly solved for decreasing values of ε until global optimality can be guaranteed.

Performance Measures. Computation time is naturally the measure of performance that matters most in the end. To gain additional insight we also consider the number of queries reguired to construct the list $\mathit{\Pi}(x)$, (11), the additional number of queries in the hierarchical consistency phase and the degree of sparsity of $\hat{\mathcal{N}}$ in the hybrid method.

Experimental Scenarios. We consider a variety of problem scenarios for evaluation: **(a)** P2H: point clouds, each uniformly sampled from the 2D unit square, squared Euclidean distance as cost, **(b)** P3H: same as P2H, but points sampled from 3D unit cube, **(c)** P2H-P1: same as P2H but with non-squared Euclidean distance as cost, **(d)** P2I: same as P2H but with inhomogeneous sampling densities and **(e)** grid: smooth 2D mass distribution, approximated by a discrete grid, cost given by squared Euclidean distance, **(f)** mesh: mass distributions on points sampled from the surface of a 3D mesh, geodesic distance (within mesh surface) as cost function. In all experiments quadtrees (resp. octrees in 3D) were used as hierarchical structures.

Last, we test an additional scenario, **(g)** P2H-LB: same as P2H, but instead of computing $\hat{c}$ by explicit minimization as in (14), we use lower bounds directly obtained from the quadtree structure. This demonstrates that the method can also be applied to avoid explicit computation of all pairwise costs, which for more complicated problems might be a costly task in itself.

Results. A summary of the numerical results is given in Table 1. The hybrid variant is *significantly faster* than the regular algorithm *for all* presented scenarios. This is due to a drastic decrease in the number of necessary constraint violation queries. In particular one can see (Fig. 1) that *the gain increases with growing problem size*. For $N = 6000$ (i.e. for $3.6 \cdot 10^7$ possible assignment pairs) the ratio of runtimes ranges from 4.6 to 48. In the hybrid variant, for most scenarios at the finest scale less than one percent of potential assignments was added to $\hat{\mathcal{N}}$. Only for mesh it was slightly more ($\approx 4\%$), owed to the more complicated

Table 1. Summary of numerical experiments for the scenarios introduced in Sect. 6 and various problem sizes. N gives the (in all experiments equal) cardinality of X and Y. For each scenario the **first row** gives the results of the dense algorithm, where 'queries' gives the number of pairs checked for creating the lists $\Pi(x)$, (11), throughout the algorithm. The **second row** gives the results of the hybrid algorithm, only at the finest scale. Here 'queries' gives the number of checks for creating all $\Pi(x)$ plus the number of hierarchical consistency checks. The **third row** gives the results for the hybrid algorithm summed over all scales, i.e. for solving the whole problem from scratch. All results are averaged over multiple instances.

In all scenarios the number of queries is reduced significantly by the hybrid variant, resulting in a corresponding runtime decrease. For some scenarios the runtime ratio full/hybrid slightly decreased from $N = 4000$ to $N = 6000$. We attribute this to the changing relation of problem size to hierarchy depth, the effects of which have yet to be more carefully examined. We expect the ratio to increase again for $N > 6000$.

N	500	1000	2000	4000	6000	500	1000	2000	4000	6000
	time					queries				
	/s	/10s	/10^2s	/10^3s	/10^3s	/10^7	/10^7	/10^8	/10^9	/10^{10}
	2.81	2.00	1.93	1.98	6.54	2.31	12.00	6.63	4.06	1.10
P2H	1.06	0.53	0.31	0.20	0.73	0.41	1.51	0.40	0.15	0.04
	1.38	0.59	0.38	0.21	0.74	0.57	1.79	0.57	0.18	0.04
		1.16	1.07	0.97	6.54		7.52	3.88	1.93	0.52
P3H		0.31	0.20	0.17	0.99		1.93	0.63	0.20	0.04
		0.42	0.34	0.24	1.41		2.90	1.30	0.34	0.07
	3.04	1.98	1.58	1.00	3.58	2.49	12.30	6.09	2.63	0.67
P2H-P1	1.05	0.49	0.24	0.10	0.31	0.53	2.30	0.61	0.17	0.05
	1.34	0.54	0.29	0.11	0.32	0.69	2.54	0.77	0.19	0.05
	3.64	2.92	2.14	2.35	7.68	2.55	13.20	7.02	4.51	1.21
P2I	1.32	0.67	0.38	0.34	1.14	0.46	2.22	0.43	0.21	0.03
	1.69	0.73	0.45	0.35	1.33	0.62	2.49	0.60	0.24	0.04
	54.70	19.60	19.30	5.92	13.20	56.00	185.00	162.00	46.90	9.91
grid	9.46	1.67	0.94	0.15	0.26	22.90	22.90	26.70	2.00	0.46
	9.53	1.76	0.95	0.16	0.27	23.00	23.70	26.80	2.15	0.47
		21.30	13.10	9.97	N/A		150.00	86.90	49.60	N/A
mesh		2.95	1.55	0.88	10.9		45.40	17.60	9.78	8.52
		3.18	1.84	0.93	11.1		48.40	22.20	10.60	8.70
	2.75	1.82	2.00	1.93	9.04	2.26	11.60	6.74	4.07	1.15
P2H-LB	1.10	0.52	0.35	0.24	1.35	0.41	1.63	0.46	0.20	0.03
	1.57	0.57	0.38	0.25	1.41	0.65	1.97	0.62	0.23	0.04

cost function. Also in the scenario P2D-LB the hybrid variant clearly outperforms the regular algorithm, while at the same time potentially saving explicit assignment cost computation. Thus, for the presented scenarios the multiscale scheme obviously works as intended.

7 Conclusion

As demonstrated in the last Section, the presented extension of the auction algorithm clearly outperforms the regular variant on all presented test scenarios. The observed gain in computation time grows with problem size. Compared to PDE approaches for OT problems our method is much more flexible: X and Y need not be regular grids on $\mathbb{R}^n$ and the cost can be chosen freely, as long as a certain regularity is retained. Due to the very limited space we could only give a very brief sketch on the theoretical properties of the algorithm, i.e. its worst case complexity, the claim that it reliably finds the global optimum and the relation between the different scales of the problem. Proofs for these claims will be presented in a more detailed future publication. It also remains to be examined more carefully how the hierarchical structure we proposed interacts with the ε-scaling scheme or whether under further assumptions on the cost function better theoretical complexity bounds can be obtained. Yet, already at this stage of research the potential of the extension is evident in all tested scenarios.

Acknowledgement. This work was supported by the DFG, grant GRK 1653.

References

1. Bertsekas, D.P.: A distributed algorithm for the assignment problem. Tech. rep., Lab. for Information and Decision Systems Report, MIT (May 1979)
2. Bertsekas, D.P.: The auction algorithm: A distributed relaxation method for the assignment problem. Annals of Operations Research 14, 105–123 (1988)
3. Bertsekas, D.P., Eckstein, J.: Dual coordinate step methods for linear network flow problems. Mathematical Programming, Series B 42, 203–243 (1988)
4. Bertsekas, D., Castanon, D.: The auction algorithm for the transportation problem. Annals of Operations Research 20, 67–96 (1989)
5. Carlier, G., Galichon, A., Santambrogio, F.: From Knothe's transport to Brenier's map and a continuation method for optimal transport. SIAM J. Math. Anal. 41, 2554–2576 (2010)
6. Haker, S., Zhu, L., Tannenbaum, A., Angenent, S.: Optimal mass transport for registration and warping. Int. J. Comput. Vision 60, 225–240 (2004)
7. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics 2, 83–97 (1955)
8. Mémoli, F.: Gromov-Wasserstein distances and the metric approach to object matching. Found. Comp. Math. 11, 417–487 (2011)
9. Pele, O., Werman, W.: Fast and Robust Earth Mover's Distances. In: Proc. Int. Conf. Comp. Vision, ICCV (2009)

10. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics, vol. 24. Springer (2003)
11. Shirdhonkar, S., Jacobs, D.W.: Approximate earth mover's distance in linear time. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008 (2008)
12. Villani, C.: Optimal Transport: Old and New. Springer (2009)
13. Wang, W., Slepčev, D., Basu, S., Ozolek, J.A., Rohde, G.K.: A linear optimal transportation framework for quantifying and visualizing variations in sets of images. International Journal of Computer Vision pp. 1–16 (2012)

Layered Mean Shift Methods

Milan Šurkala, Karel Mozdřeň, Radovan Fusek, and Eduard Sojka

Technical University of Ostrava,
Faculty of Electrical Engieneering and Informatics,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
milan.surkala.st@vsb.cz

Abstract. Segmentation is one of the most discussed problems in image processing. Many various methods for image segmentation exist. The mean-shift method is one of them and it was widely developed in recent years and it is still being developed. In this paper, we propose a new method called Layered Mean Shift that uses multiple mean-shift segmentations with different bandwidths stacked for elimination of the over-segmentation problem and finding the most appropriate segment boundaries. This method effectively reduces the need for the use of large kernels in the mean-shift method. Therefore, it also significantly reduces the computational complexity.

Keywords: layer, segmentation, image, mean shift, over-segmentation.

1 Introduction

Segmentation can be solved by many various algorithms. They differ in speed and accuracy. Both goals are often contradictory, very fast methods are often not very accurate and vice versa. Mean shift (MS) is one of the most popular methods in recent years, although it was firstly presented in 1975 [1]. Nowadays, this method is known as Blurring MS (BMS) and it was deeply discussed in 2006 [2]. The mean-shift methods belong to the more precise methods giving very nice filtration results. Many of them give nice segmentation results too. The problem of MS is in a high computational complexity, although many faster variants were presented in few recent years. The high computational complexity is most obvious in general mean-shift method usually denoted as MS. It was presented in 1995 [3] and deeply studied in [4], [5], and [6].

Mean shift is an iterative method that seeks for a position with the locally highest density of data points. During computation, a kernel density estimate is computed for every data point. Because we are segmenting images, the pixels in images are used as these data points in our case. Each pixel is shifted according to the density estimate and computation is carried out until the convergence when the shift is very small or zero. Two datasets are used in general MS. We distinguish between an original and a shifted data. In the first iteration, both are the same. In the following iterations, we compute mean shift for the already shifted pixel, but the neighboring pixels in the kernel placed on the computed

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 465–476, 2013.

pixel are taken from the original dataset that is never changed. BMS uses a slightly different approach because only one dataset is used. After each iteration, the output from the previous iteration is used as an input dataset for the next one. Therefore, the dataset is slightly blurred after each iteration (the computed pixel is not taken from the original dataset like in MS but it is taken from the slightly blurred dataset) and convergence is faster. All the pixels that converged to the same position, create one segment in the processed image.

The speed of all mean-shift methods is highly dependent on the size of kernel (the number of pixels that are needed to compute the kernel density estimate) and the number of iterations needed to achieve the convergence. It was proved that MS has a higher number of iterations per pixel than BMS [2]. In 2009, Evolving MS (EMS) was presented in [7] and [8]. It promises even lower number of iterations per pixel but each iteration requires a lot of overheads. Minimizing of the kernel sizes is mostly utilized in the hierarchical approaches [9], [10] and [11], where a small kernel is used in the first stage of these algorithms and then larger kernels are used in the following stages where the input is the computed segmentation from the previous stage. In this paper, we present a new Layered Mean Shift method family that is based on minimization of kernel sizes in order to achieve a faster segmentation. Our method also improves the detection of significant boundaries of objects and minimizes the over-segmentation problem.

In the next section, the basics of Mean Shift are going to be described. Section 3 is devoted to our new method called Layered Mean Shift. We use layered versions for several mean-shift methods, but for explanation, LxMS abbreviation for an unspecified layered mean-shift method will be used generally.

2 Mean Shift

Let $X = \{x_i\}_{i=1}^{n} \subset R^d$ be a dataset of n points in the d-dimensional space. The *kernel density estimator* is given by the equation

$$p(x) = \frac{1}{n\sigma^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{\sigma}\right), \tag{1}$$

where σ is a bandwidth parameter limiting the size of kernel function $K(x)$. In some literature, denomination the bandwidth parameter as h is also used. We can distinguish between two types of bandwidths in images. The spatial bandwidth σ_s is the first one and limits the neighbourhood of the processed pixel in x and y axis. The range bandwidth is the second one and it is denoted by σ_r. It indicates the maximal luminance difference of the sample that can fall inside the kernel. We can have more bandwidths in colour images, for example, three bandwidths for each colour channel. In our work, we are focused on the greyscale images and only one σ_r is needed. The fraction before the sum in Eq. (1) is a normalization constant. The processed pixel is denoted as x and all pixels in the neighbourhood (kernel) are labeled as x_i.

Many types of kernel functions exist. The *Gaussian* is the most popular and often gives the nicest results, but it has also few drawbacks. It is not trun-

cated kernel and covers the whole dataset. The bandwidth is not limiting the size of kernel but only the contribution of the samples. For computation of one mean-shift vector, we need to compute the kernel function with all image pixels. Therefore, the Gaussian kernel is very slow and inherently not appropriate for using it in our method that limits the size of kernel. Of course, there is a possibility of truncation of the Gaussian kernel in a defined distance.

In our approach, we use another very famous kernel, the *Epanechnikov* kernel. It is truncated and the σ parameters limits the contribution of the pixels in the distance of σ. All pixels that are out of the hypersphere given by these parameters, are not involved in computation and the algorithm can be much faster, especially with smaller values of σ. The Epanechnikov kernel is given by the equation

$$K(x) = \begin{cases} 1 - x^2, & \text{if } \|x\| \leq 1 \\ 0, & \text{otherwise} \end{cases} . \tag{2}$$

All inferences of kernel estimates and their relationship to mean-shift methods are deeply described in [4] and [6]. Therefore, we do not deal with them deeply in this paper. On the other hand, the mean-shift vector should be mentioned at least. Its equation is given by

$$m_{\sigma,k}(x) = \frac{\sum_{i-1}^{n} x_i k\left(\left\|\frac{x-x_i}{\sigma}\right\|^2\right)}{\sum_{i-1}^{n} k\left(\left\|\frac{x-x_i}{\sigma}\right\|^2\right)} - x\,, \tag{3}$$

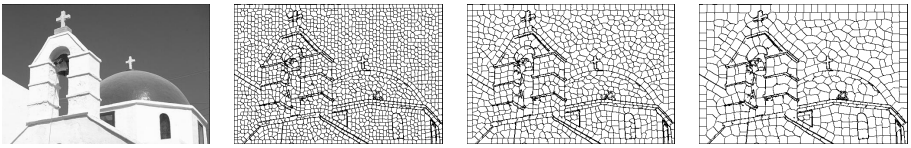
where the function $k(x)$ is a derivative of the kernel $K(x)$. The first term on the right-hand side is a new position of the processed pixel x (estimate of the position with the highest density of data points), the second term is the former position. The difference $m_{\sigma,k}(x)$ between them is called the *mean-shift vector*. In this case, we present the equation for Blurring MS that is faster than general MS. It uses the modified dataset in each iteration and its results are more regular. General experiments with our LxMS method will be carried out with the LBMS version of it.

3 Layered Mean Shift Methods

We present a new *Layered Mean Shift* (LxMS) that is aimed to the reduction of computational time as well as to reduction of the over-segmentation problem. It is well known that the size of segments is highly dependent on the value of σ parameter. The larger the σ parameter is, the larger are the segments. If we expect large segments, we are forced to use larger σ_s. This leads to slower computation because of $O({\sigma_s}^2)$ complexity for evaluation of one mean-shift vector for one pixel. Our LxMS method solves this speed and over-segmentation problem. LxMS does not suffer from over-segmentation even if it is used with small bandwidths.

The main idea of LxMS is in execution of multiple mean-shift segmentations with different kernel sizes that are not very large in any of executed segmentations. General MS, Blurring MS, and Evolving MS can be used as a basic method that will be executed repeatedly. Layered mean-shift method using BMS segmentation as its base can be called LBMS (Layered Blurring Mean Shift), the same applies to MS in HMS method and EMS in HEMS method. As we already said, presented results that explain the layered approach use BMS as its base in all cases (LBMS method).

We use m segmentations, each with a different spatial bandwidth. Then these segmentations are overlaid. Important edges in the image are highlighted in all segmentations, whereas over-segmentation artifacts are positioned in various locations in each segmentation. If we stack these segmentations, these artifacts are almost invisible (they are placed only once in some area) and only important edges remain (each segmentation produces the same border in the same place). For example, we can execute three different mean-shift segmentations, the first one with $\sigma_s = 3$, second one with $\sigma_s = 4$, and third one with $\sigma_s = 5$. We use the same σ_r for all segmentations but it is not necessary.



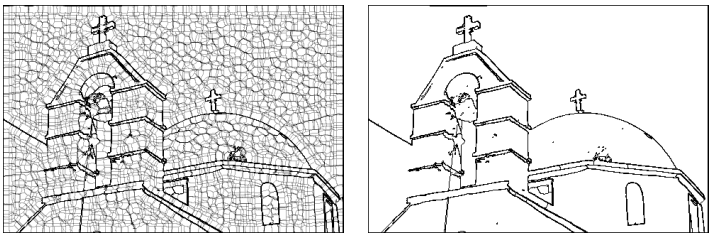
(a) original image (b) $\sigma_s = 5$ (c) $\sigma_s = 7$ (d) $\sigma_s = 9.8$

Fig. 1. Phases of the LBMS method. The original image is in the first column. The images with different bandwidths are shown in next three figures.

Three different segmentations are visible in Fig. 1. Each was computed with a different spatial bandwidth and, therefore, created a different segmentation. In all three cases, the boundaries of church are clearly evident. The number of stacked segmentations is not limited, of course.

Fig. 2(a) shows that even very small searching windows (kernels) with $\sigma_s < 10$ in the 481×321 pixel image completely reduces the problem of over-segmentation. There is no need to use spatial bandwidth with the size of hundreds of pixels. The better speed is achieved, because we carry out a small number of fast segmentations instead of one very slow segmentation.

It is obvious that only the image of stacked segmentations (Fig. 2(a)) is useless and it has to be processed to create one useful result. In Fig. 2(b), the result of merging the segments is visible. Many approaches are possible. If we want only edges, we can use simple edge following algorithms. If we need a real segmentation, another approach should be used. In LxMS, we use our own *segment merging* algorithm. We pick all pairs of pixels from the image and sum how many times they were in the same segment from m executed segmentations. The threshold t lower than the number of stacked segmentations is set. If the sum



(a) Stacked image from three segmentations (b) Result after basic processing

Fig. 2. Stacked segmentation and result of merging the segments

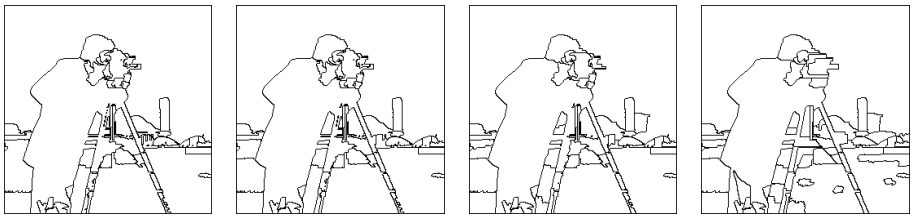
is higher or equal to the threshold, both pixels are inserted into one segment. For example, we have three segmentations and we can set the threshold value to 2. If two random pixels were in the same segment at least in two of three segmentations, they belong to one segment. It can be clearly seen in Fig. 3. After carrying out this proces, we should remove very small segments; see the small spots in 2(b).

(a) $\sigma = 5$ (b) $\sigma = 7$ (c) $\sigma = 9.8$

Fig. 3. Merging of segments. For example, if two random pixels are twice in one segment from three possible segmentations (Fig. 3(a) and Fig. 3(c)), they are given the same segment label. The number of segmentations and necessary number of the same assignments to segments is adjustable. It does not need to be 3 and 2 like in this example.

Intuitively, we should try to find the pairs between all pixels in the whole image. It leads to high complexity of $O(n^2)$ that is the same as complexity of MS and BMS algorithm. We observed that approx. $20 - 40\%$ of computational time is spent on this merging the segments with such a naive approach. It is obvious that there is almost no possibility to obtain two pixels in one segmentation if their spatial distance is larger than the spatial bandwidth (if they are not covered by the kernel, they will be hardly assigned to one attractor). Therefore, we do not need to check all pixels with all pixels in the whole image, but only with their close neighbourhood given by σ parameter. Our experiments showed that the results were the same with such an acceleration and complexity dropped to $O({\sigma_s}^2)$.

If we limit the maximal distance for merging more, we prevent the merging of the points that are divided by an another segment spatially between them.



(a) $n = \sigma$ (b) $n = 3$ (c) $n = 2$ (d) $n = 1$

Fig. 4. Segmentation merging with limiting the neigbourhood for searching pairs of pixels belonging to the same segment. Parameter n is the radius of this neigbourhood.

It should not be acceptable in statistical usage of MS method but sometimes it can be useful in digital images. This method is sort of a trade off. Some details are suppressed (segments have more regular shapes) but also does not connect corresponding parts of segments which can divide one object to more pieces and also can make additional artifacts. Therefore, we recommend to use $n = \sigma$. In Fig. 4, reduction of small segments with the size smaller than 50 pixels was used. These small segments emerge in the places where boundaries of true segments differ very slightly in different segmentations. These pixels are not assigned to any segment and create their own small segment. Such small segments are assigned to a neighboring segment.

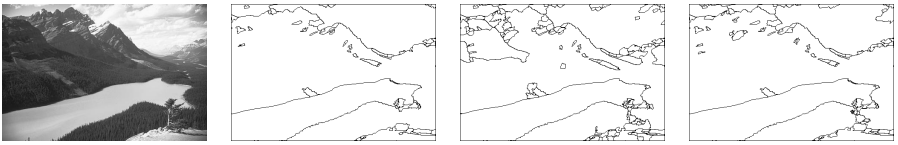
Algorithm 1. Layered Mean Shift

Input: Dataset X, spatial bandwidth σ_s, range (luminance) bandwidth σ_r, bandwidth multiplier l, number of segmentations m, threshold t.
Output: A clustered dataset X_s

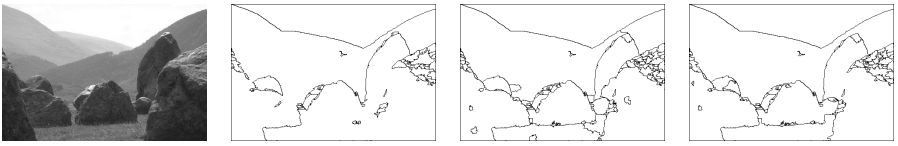
```
Set index i = 0
repeat
    Evaluate MS (or BMS) segmentation X_i with bandwidths σ_s and σ_r, where i is a index of segmentation
    Multiplicate σ_s by l and increase index i by 1
until index i = m
for all pixels x_j do
    for all pixels x_k in circle neighbourhood of pixel x_j with radius of σ_s do
        Sum the number of segmentations X_i where pixels x_j and x_k belong to the same segmentation
        If the sum is equal or higher than a threshold t, both pixels x_j and x_k are assigned to the same segment
    end for
end for
Eliminate segments with size smaller than a preset threshold (fraction of image area).
```

4 Experiments

In this section, the experiments with our algorithm are provided. We tested LBMS algorithm in comparison with the original BMS and we studied the segmentation quality as well as the speed of algorithm. Also, the hierarchical version of BMS (called HBMS) was tried. We use removal of small segments with the size smaller than a preset threshold (for example 1/5000 of image area). The bandwidth range and the number of stacked segmentations are mentioned for each test. We use images from Berkeley Image Dataset [12].



(a) original image (b) 4; 1.4; 3-2/24 (c) 4; 1.25; 4-3/24 (d) 4; 1.4; 4-3/24



(e) original image (f) 5; 1.4; 3-2/24 (g) 5; 1.25; 4-3/24 (h) 5; 1.4; 4-3/24

Fig. 5. Comparison of LBMS segmentations. The range bandwidth σ_r is the number after the slash. "The first number stands for the value of σ_s in the first segmentation, the second number is the multiplier. The notation $3-2$ says that 3 segmentations were processed and pixels that were 2 or more times in the same segmentation have been merged.

In Fig. 5, we see that different parameters lead to slightly different segmentations but we can not fully determine general influence of parameters to the segmentation quality. Of course, the larger number of processed segmentations causes a higher computational time. The higher values of multiplier increase computational time too because of higher values of σ_s parameters in the following segmentations. A small difference of spatial bandwidths between the stacked segmentations causes the increase of the number of segments. The same effect can be seen if we increase the number of stacked segmentations (of course, it can be lowered by lowering the threshold). We can say that the higher number of stacked segmentations often produces the resultant segmentation with a higher quality of details (see the incomplete stones in Fig. 5(f) and better result in Fig. 5(h)) but also with slightly more visible over-segmentation.

We can see several examples in Fig. 6 and processing times in Table 2. It is obvious that LBMS does not suffer from over-segmentation even if the kernel sizes were below $\sigma_s = 10$. Both BMS and HBMS used $\sigma_s = 20$, but there is a visible over-segmenation in both results. If we want to reduce this effect, we have to

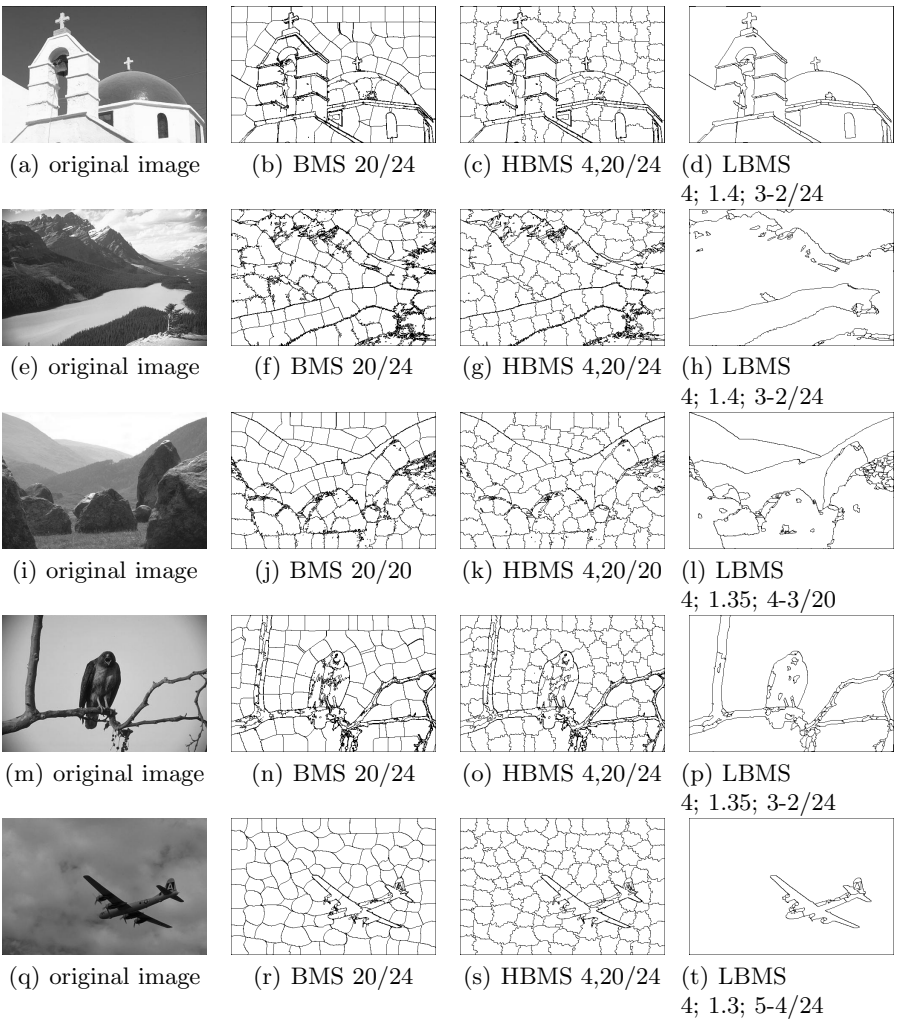


Fig. 6. Range bandwidth σ_r is the number after the slash. In BMS, the number before the slash is the spatial bandwidth σ_s. In HBMS, the numbers 4,20 mean that the first stage used $\sigma_s = 4$ and the second one used $\sigma_s = 20$. The first number in the LBMS notation is the value σ_s in the first segmentation and the second number is the bandwidth multiplier. The notation $3 - 2$ says that 3 segmentations were processed and pixels that were 2 or more times in the same segmentation have been grouped.

Table 1. Comparison of the numbers of segments and speed depending on parameters

image	bandwidth σ_s	multiplier	segmentations	threshold	time	segments
mountains	4	1.4	3	2	56.2 s	85
mountains	4	1.25	4	3	70.6 s	178
mountains	4	1.4	4	3	108.5 s	119
stones	5	1.4	3	2	87.2 s	90
stones	5	1.25	4	3	117.6 s	150
stones	5	1.4	4	3	171.6 s	121

Table 2. Comparison of the numbers of segments (seg) and the computational time (t[s]) depending on the algorithm

	BMS		HBMS		LBMS	
image	t[s]	seg	t[s]	seg	t[s]	seg
church	158.1 s	274	9.6 s	253	58.7 s	114
mountains	158.1 s	238	10.1 s	262	56.2 s	85
stones	155.9 s	198	9.7 s	185	106.3 s	149
bird	150.6 s	260	10.4 s	265	59.8 s	103
airplane	185.0 s	145	9.4 s	146	168.6 s	27

enlarge the kernel size. That would lead to enormous increase of computational time (quadratically). The segmentation is subjectively visually very nice and the computational times are much better than in BMS. On the other hand, the hierarchical approaches are still much faster but they suffer from over-segmentation. In many images, only 3 stacked segmentations are sufficient but a higher number of segmentations could be useful in images with more noticeable textures. Enlarging the number of executed segmentations would cause the increase of computational time. The more simple the images are, the smaller number of segmentations and smaller kernel size has to be used. Small bandwidths are often sufficient because they also produce different segmentation boundaries in flat areas and the same boundaries on the true edges of detected objects.

In Fig. 7, you can see five images segmented by various mean-shift methods. We used MS, BMS, EMS, their hierarchical versions HMS, HBMS, HEMS and their layered versions LMS, LBMS and LEMS. The first image is a synthetic image with smooth background gradient and smooth shapes. MS had very big problem to segment it because of zero gradient of underlying structure. The data point can not move and image is segmented only around the edges, where the non-zero gradient of density exists. The second image is the noisier version of the first image. Therefore, it can be segmented by MS. The following three images are the real-life images from the Berkeley Image Database [12]. One stage algorithms (MS, BMS and EMS) used the spatial bandwidth $\sigma_s = 25$. All of the hierarchical approaches were used in their 3-stage versions using bandwidths of $\sigma_{s_1} = 3.5$, $\sigma_{s_2} = 12$ and $\sigma_{s_3} = 50$. The layered version used two possible configurations. The first one is $3-2$, where 3 stages were processed and the pixels were merged into

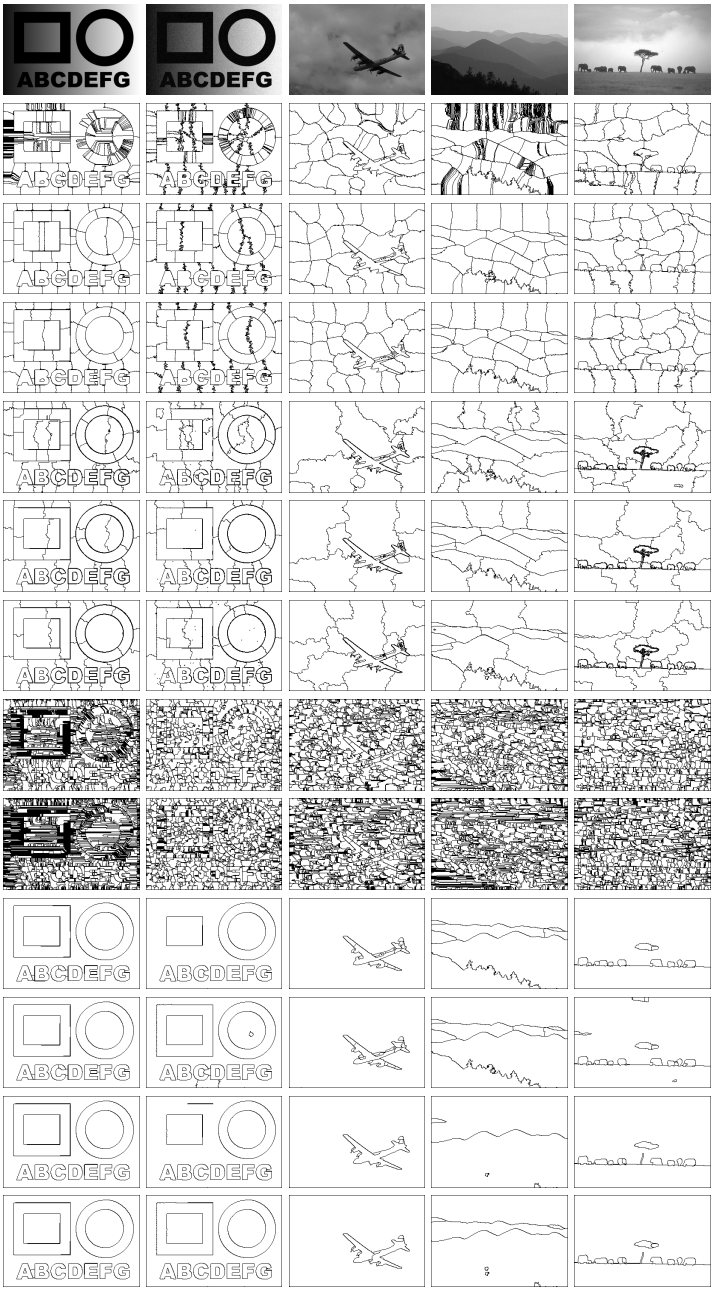


Fig. 7. Rows 1: the original image; 2/3/4: MS/BMS/EMS (spatial bandiwdth $\sigma_s = 25$); 5/6/7: HMS/HBMS/HEMS ($\sigma_s = 3.5/12/50$); 8/10/12: LMS/LBMS/LEMS (3-2 stages, $\sigma_s = 4$, multiplier of the bandwidth $mul = 1.3$); 9/11/13: LMS/LBMS/LEMS (4-3 stages, $\sigma_s = 4$, $mul = 1.3$); the notation is similar to Fig. 6.

a larger segment if they both were at least twice in the same segment. The last configuration was analogically $4-3$. It is obvious that LMS is unusable as the original MS gives an unstable result that can not be easily merged. LBMS and LEMS usually give a stable result around the most visible edges and, therefore, the layered approach is very beneficial here. In our additional tests, it has been shown that LMS needs at least $\sigma_{s_1} = 9$ for satisfactory result.

Although the largest spatial bandwidth was 6.7 in $3-2$ configuration or 8.7 in $4-3$ configuration, it definitely outperforms all other algorithms in the area of the over-segmentation problem. We can enlarge the spatial bandwidth in the classical and hierarchical algorithms to decrease the over-segmentation but it will lead to much longer computational time and potentially inaccurate results (the large σ_s will cover the large image or even the whole image and the spatial term will be unimportant - all the pixels with the same brightness in the image will be grouped even if they are separated by another segments). Such a situation does not happen in layered algorithms because of small spatial merging bandwidths.

Table 3. Comparison of the speed (t[s]) depending on algorithm

	synth. image1	synth. image 2	airplane	mountains	savana
MS	2185.35 s	2014.93 s	2112.62 s	3348.3 s	2999.28 s
BMS	82.29 s	94.34 s	107.2 s	80.58 s	103.35 s
EMS	1061.62 s	1129.69 s	629.86 s	543.26 s	753.63 s
HMS3	67.9 s	20.18 s	23.71 s	24.81 s	24.29 s
HBMS3	7.56 s	7.63 s	6.63 s	6.96 s	6.84 s
HEMS3	27.14 s	33.94 s	16.63 s	14.41 s	18.86 s
LMS3/2	318.67 s	123.08 s	282.11 s	345.74 s	222.02 s
LBMS3/2	21.21 s	19.43 s	24.08 s	23.22 s	21.62 s
LEMS3/2	79.41 s	75.21 s	70.12 s	72.05 s	73.28 s
LMS4/3	603.13 s	280.99 s	538.54 s	695.87 s	442.88 s
LBMS4/3	36.75 s	36.22 s	40.52 s	37.88 s	34.78 s
LEMS4/3	385.45 s	318.73 s	125.6 s	152.81 s	127.33 s

Table 3 shows the speed of all the algorithms. The hierarchical approaches are the fastest but they still suffer from the over-segmentation problem. The layered versions are 2.5 to 4-times slower (with the exception of LEMS4-3 in the synthetic images) with no over-segmentation problem. There rises a question whether this trade off is acceptable or not.

5 Conclusion

The layered mean-shift methods showed that they are relatively fast methods that primarily reduce the over-segmentation problem even with very small kernel sizes. They are very well suited for images with not very noticeable textures. In other cases, the number of stacked segmentations should be enlarged to achieve

a proper segmentation. Mean-shift, blurring mean-shift and evolving mean-shift approaches can be embedded into the LxMS method but it has been shown that general MS is not a very good choice. In LMS case, it needs much larger initial spatial bandwidth. The next goal is to improve the grouping of pixels in the stacked segmentations to achieve smaller a sensitivity to stronger textures.

Acknowledgements. This work was partially supported by the grant SP 2013 / 185 of VŠB-TU of Ostrava, FEECS.

References

1. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory 21, 32–40 (1975)
2. Carreira-Perpiñán, M.: Fast nonparametric clustering with Gaussian blurring mean-shift. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, pp. 153–160. ACM, New York (2006)
3. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 790–799 (1995)
4. Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1197–1203 (1999)
5. Comaniciu, D., Ramesh, V., Meer, P.: The variable bandwidth mean shift and data-driven scale selection. In: IEEE International Conference on Computer Vision, vol. 1, p. 438 (2001)
6. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
7. Zhao, Q., Yang, Z., Tao, H., Liu, W.: Evolving mean shift with adaptive bandwidth: A fast and noise robust approach. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009, Part I. LNCS, vol. 5994, pp. 258–268. Springer, Heidelberg (2010)
8. Yang, Z., Zhao, Q., Liu, W.: Neural signal classification using a simplified feature set with nonparametric clustering. Neurocomput. 73, 412–422 (2009)
9. Vatturi, P., Wong, W.K.: Category detection using hierarchical mean shift. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 847–856. ACM, New York (2009)
10. DeMenthon, D., Megret, R.: Spatio-Temporal Segmentation of Video by Hierarchical Mean Shift Analysis. Technical Report LAMP-TR-090,CAR-TR-978,CS-TR-4388,UMIACS-TR-2002-68, University of Maryland, College Park (2002)
11. Šurkala, M., Mozdřeň, K., Fusek, R., Sojka, E.: Hierarchical blurring mean-shift. In: Blanc-Talon, J., Kleihorst, R., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2011. LNCS, vol. 6915, pp. 228–238. Springer, Heidelberg (2011)
12. 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)

Partial Optimality via Iterative Pruning for the Potts Model

Paul Swoboda[1], Bogdan Savchynskyy[2], Jörg Kappes[1], and Christoph Schnörr[1,2]

[1] Image and Pattern Analysis Group,
[2] Heidelberg Collaboratory for Image Processing,
University of Heidelberg, Germany

Abstract. We propose a novel method to obtain a part of an optimal *non-relaxed integral* solution for energy minimization problems with Potts interactions, known also as the minimal partition problem. The method empirically outperforms previous approaches like MQPBO and Kovtun's method in most of our test instances and especially in hard ones. As a starting point our approach uses the solution of a commonly accepted convex relaxation of the problem. This solution is then iteratively pruned until our criterion for partial optimality is satisfied. Due to its generality our method can employ any solver for the considered relaxed problem.

1 Introduction

1.1 Problem Formulation

Continuous Model. Consider the minimal partition problem

$$\min_{u\in BV(\Omega;\{1,\dots,k\})} \int_{\Omega} |Du| + W(x,u(x))dx\ . \tag{1.1}$$

This problem and approximation algorithms for it are discussed in [12] for the discrete case, and in [7] and [15] for the continuous case.

Minimizing discretizations of the above problem is NP-hard for $n \geq 3$, therefore it is common to resort to a convex relaxation. Introduce

$$\begin{aligned} u^i(x) &\geq 0\ ,\ i = 1\dots,k\,, \\ \textstyle\sum_{i=1}^k u^i(x) &= 1\ ,\ x \in \Omega\,, \end{aligned} \tag{1.2}$$

or equivalently $u(x) \in \Delta_k$, $x \in \Omega$, and minimize (1.1) over (1.2). In general a minimizer u^* of the relaxed problem will not be binary anymore, but for some $x \in \Omega$ it may still hold true. A natural question is: Is there a minimizer $\tilde{u}$ of the original NP-hard problem (1.1) and such a subset $A \subset \Omega$, that $\tilde{u}(x) = u^*(x)$ for $x \in A$? In other words, is u^* partially optimal or *persistent* on some set A? How can we determine such a set A?

Finding persistency is not only theoretically interesting, but it also allows in many cases to solve the problem w.r.t. the remaining non-persistent variables with other methods as done in [3] and thereby to obtain its complete globally optimal solution. Moreover, solving the problem with respect to the non-persistent variables is simplified by the fact, that the latter are often weakly connected or/and form small connected components.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 477–488, 2013.

Discrete Model. For solving problem (1.1) in practice, one must discretize it. There are many possible ways to do so, see e.g. [7] or [15]. We consider a discretization, which introduces anisotropies, but can be stated for general graphs $G = (\Omega, E)$:

$$J(u) = \sum_{a \in \Omega} \langle c(a), u(a) \rangle + \sum_{(a,b) \in E} \sum_{l=1}^{k} \alpha_{a,b} |u^l(a) - u^l(b)| \tag{1.3}$$

with $\alpha_{a,b} > 0$, $(a,b) \in E$, $c(a) \in \mathbb{R}^k$ and $u(a) \in \{0,1\}^k$, $a \in \Omega$, satisfies additionally (1.2). The discrete problem (1.3) is also known as a Potts model.

Note that for a grid graph, uniform weights $\alpha_{a,b}$ and c being a local average of W, this is a particular discretization of the minimal partition problem (1.1). One can systematically approximate the minimal partition problem with graphs of higher connectivity with the method presented in [6], while still solving a problem of the form (1.3).

1.2 Related Work and Contribution

The task of finding persistent variables in labeling problems has been studied and many approaches have been proposed [5,9,10,13,14,16,17,20]. To our best knowledge the earliest paper concerning itself with persistency is [16], which states a persistency criterion for the stable set problem and verifies it for every solution of a certain relaxation, which the roof duality method in [5] uses and which is also the basis for the well known QPBO-algorithm [5, 17]. Roof Duality has been extended for Multi-Label problems in [13, 20] and for higher order binary problems in [10]. A different approach, specialized for Potts models, is pursued in [14], where possible labelings are tested for persistency.

MQPBO. In [13] the authors transform the multilabeling problem into a quadratic binary problem. Their transformation is dependent upon choosing a label order and their results are so as well. It is not known how to choose an optimal label ordering to obtain the maximum number of persistent variables. For actually solving their problem they use a relaxation which is an outer relaxation of the local polytope [18, Prop. 1]. One can show that *the relaxation we use is strictly tighter than theirs and our approach also generalizes to tighter relaxations as well.* Experimentally we are able to label a much higher percentage of points persistently. In case of high regularization weights our approach can *determine a substantially higher number of persistently labeled variables*. While the model [18] can solve more general interaction potentials, it needs significantly more memory for the interaction terms. For the Potts model our approach consumes *substantially less memory*, if a suitable algorithm for solving the relaxation is used.

Kovtun's Approach [14] consists in searching for partially optimal labelings by constructing auxiliary problems, solving these and testing for persistent variables. Each of the auxiliary problems is however less tight than the relaxation we use. Also experimentally, *we could usually label more variables than this method.*

1.3 Organization

We present

- a new persistency criterion for ensuring a labeling to be partially optimal, see Section 2,

- an algorithm for finding the provably biggest labeling, such that the persistency criterion is satisfied, see Section 2,
- experimental validation of our approach and a comparison to existing methods and new ways to use them together, see Section 3.

For the sake of clarity of presentation we have moved the proofs of all theoretical statements to Section 5.

1.4 Notation

We reserve the variable k for the number of classes in problem (1.3) and denote by $e_i = (0,\dots,0,1,0,\dots,0)^\top$ the i^{th} basis vector of $\mathbb{R}^k$ with a 1 in its i^{th} component. Let $\mathscr{E}_k := \{e_1,\dots,e_k\}$ contain the k unit basis vectors and denote by $\Delta_k = \mathrm{conv}(\mathscr{E}_k)$ its convex hull. For notational convenience we introduce two labeling spaces. Let

$$D = \{u : \Omega \to \mathscr{E}_k\}, \tag{1.4a}$$

$$D' = \{u : \Omega \to \Delta_k\}. \tag{1.4b}$$

The set D consists of all integral labelings and is nonconvex. The set D' is its convex hull, consists of functions defined by (1.2) and is used as feasible set in the relaxations.

The characteristic function of a set C is defined as $\delta_C(x) = \begin{cases} 0 & ,x \in C \\ \infty & ,x \notin C \end{cases}$.

For a subset of nodes $A \subset \Omega$ let the functional restricted to A be

$$J_A(u) = \textstyle\sum_{a\in A}\langle u(a), c(a)\rangle + \sum_{(a,b)\in E}\sum_{l=1}^{k} \alpha_{a,b}|u^l(a) - u^l(b)| \ . \tag{1.5}$$

For $A \subset \Omega$ let its boundary be given by

$$\partial A = \{a \in A \ : \ \exists b \in \Omega\backslash A \text{ s.t. } (a,b) \in E\} \ . \tag{1.6}$$

Definition 1. *For a* boundary term $w : \partial A \to \mathscr{E}_k$ *let* the functional restricted to A with the boundary term w *be defined by*

$$J_{A,w}(u) = J_A(u) + \sum_{(a,b)\in E: a\in A, b\in\Omega\backslash A} 2\alpha_{a,b}\langle w(a), u(a)\rangle \ . \tag{1.7}$$

2 Persistency for the Discretized Potts Problem

First we propose a criterion for partial optimality. Suppose we have found an integer labeling on a set A, optimal for J_A, which is not affected by what happens on its complement $\Omega\backslash A$. Then it is immediate that the labeling is partially optimal. We propose in the following Lemma 1 a sufficient condition for this situation. More specifically, a binary minimizer of $J_{A,w} + \delta_{D'}$, which conforms to the boundary conditions w on ∂A is partially optimal.

Lemma 1. *Let $w : \partial A \to \mathscr{E}_k$ be given. Suppose $u^* : A \to \mathscr{E}_k$ is optimal for the functional*

$$J_{A,w}(u) + \delta_D(u) \tag{2.1}$$

and $u^*(a) = w(a)\ \forall a \in \partial A$. Then there exists a labeling $\tilde{u} : \Omega \to \mathscr{E}_k$ which is optimal for $J(u) + \delta_D$ and such that $\tilde{u}_{|A} = u^*$.

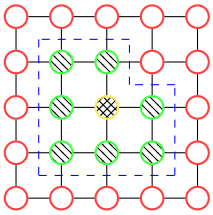


Fig. 1. An exemplaric graph for the setting of Lemma 1. The blue dashed line encloses the set A, the green nodes with the diagonal pattern have a boundary labeling w while the yellow node with the crosshatch pattern is in the interior of A and thus has no boundary labeling.

For computational purposes we must relax the functional $J_{A,w} + \delta_D$. Still the statement of Lemma 1 essentially holds:

Corollary 1. *Let $w : \partial A \to \mathscr{E}_k$ be given. Suppose the integral labeling $u^* : A \to \mathscr{E}_k$, is optimal for the relaxed functional*

$$J_{A,w}(u) + \delta_{D'}, \tag{2.2}$$

and the boundary condition

$$u^*(a) = w(a),\ a \in \partial A\,, \tag{2.3}$$

holds. Then there exists $\tilde{u} : \Omega \to \mathscr{E}_k$ optimal for $J(u) + \delta_D$ such that $\tilde{u}_{|A} = u^*$, i.e. u^* is partially optimal.

Corollary 1 forms the basis for Algorithm 1, which constructs a set of persistent variables. The algorithm is initialized with the whole set Ω and recursively shrinks it by removing variables taking non-integral values or not conforming to the boundary condition (2.3). The process stops, when there is an optimizer u^* fulfilling the conditions of Corollary 1 for the remaining set A^*.

Algorithm 1. Finding persistent variables

Data: $G = (\Omega, E)$, $c : \Omega \to \mathbb{R}^k$,
$\alpha_{a,b} \in \mathbb{R}_+ : (a,b) \in E$
Result: $A^* \subset \Omega$, $u^* : A^* \to \mathscr{E}_k$
Initialize:
$A^0 = \Omega$;
$\tilde{w}^0 = 0$;
Choose a $\tilde{u}^0 \in \operatorname{argmin}_u J_{A^0,\tilde{w}^0}(u) + \delta_{D'}$;
$t = 1$;
while $\tilde{u}^t \notin D$ *or* $\tilde{u}^t_{|\partial A^t} \neq \tilde{w}^t$ **do**
$W^t = \{b \in \partial A^{t-1} : \tilde{u}^{t-1}(b) \neq \tilde{w}^{t-1}(b)\}$;
$A^t = \{a \in A^{t-1} : \tilde{u}^{t-1}(a) \in \mathscr{E}_k\} \backslash W^t$;
$\tilde{w}^t = \tilde{u}^t_{|\partial A^t}$;
Choose a $\tilde{u}^t \in \operatorname{argmin}_u J_{A^t,\tilde{w}^t}(u) + \delta_{D'}$;
$t = t + 1$;
end
$A^* = A^t$;
$u^* = \tilde{u}^t$;

Solve the relaxed problem over Ω without boundary conditions

Shrink the set A^t by removing variables taking non-integral values or not conforming to the current boundary condition

In each iteration the set A^t shrinks. Since Ω is finite, the algorithm converges in at most $|\Omega|$ steps. If the algorithm stops, then we have that

$$u^* \in \operatorname{argmin}_u J_{A^*,w} + \delta_{D'}, \tag{2.4}$$

$w = u^*_{|\partial A^*}$ and $u^* \in D$. Hence u^*, w and A fulfill the conditions of Corollary 1, which proves persistency. In what follows we will show, that under a mild technical assumption Algorithm 1 is in some sense optimal, i.e. it delivers the *greatest* persistent set conforming to Corollary 1.

Assumption 1. *There is a unique solution of $J_{A^t,w^t} + \delta_{D'}$ for each A^t and each w^t : $\partial A^t \to \mathscr{E}_k$ obtained during iterations of Algorithm 1.*

Definition 2. *A subset $A \subset \Omega$ is the* greatest *persistent set for the functional $J + \delta_{D'}$, if for all other sets $A' \subset \Omega$ fulfilling the conditions of Corollary 1, it follows that $A' \subset A$.*

Theorem 1. *Under Assumption 1 Algorithm 1 returns the greatest persistent set A^*.*

Remark 1. If Assumption 1 does not hold, then Algorithm 1 is not deterministic and the obtained set A^* is not necessarily the greatest persistent set. The simplest example of such a situation occurs if the relaxation $J + \delta_{D'}$ is tight, but has several integer solutions. Any convex combination of these solutions will form a non-integral solution. However this fact cannot be recognized by our method and hence these entries of the solution will not be marked as persistent.

Remark 2. Algorithms similar to Algorithm 1 can be applied also to tighter relaxations, e.g. when one includes cycle inequalities similar to [19]. All our results are independent of the specific relaxation and the method one uses to optimize the relaxed problems (2.2). One can show that the persistent variables one obtains with a tighter relaxation are a superset of persistent variables one gets from a weaker relaxation.

At first glance it may seem that Algorithm 1 is not very efficient, due to the need to compute optimal solutions to possibly many slightly differing problems in the iterations of the while loop in Algorithm 1. The procedure is however significantly accelerated with a warm start, i.e. initializing the algorithm with the variables from the previous iteration. We discuss time issues in Section 3.

3 Experiments

We compare to the following methods:

- **MQPBO:** see section 1.2 and [13,18]. As in [13] we fix a label order before running the MQPBO algorithm. The labels are ordered according to the strength of the local data term. Note that contrary to our and Kovtun's approach, MQPBO can also detect persistency for labels which will *not* belong to any optimal solution.
- **Kovtun:** see section 1.2 and [14].
- **KMQPBO:** We first run Kovtun's method and then we run MQPBO on the variables which could not be found persistent by Kovtun's method.

– **KMQPBO100:** We first run Kovtun's method and then run sequentially 100 iteration of the MQPBO algorithm with a randomly sampled label order and accumulate persistent variables.

Note that KMQPBO100 is an improvement upon the remaining (K)MQPBO methods and obviously also over Kovtun's method. MQPBO's results really depend upon the label ordering and therefore finding a good or somehow optimal label order for specific problems remains an interesting task. This can be seen in the experiments, where KMQPBO100 outperforms KMQPBO. While it would be favourable for KMQPBO to optimize over all possible permutation per variable, this is computationally not tractable. The problem of choosing the best label order has not been dealt with in literature, as far as we know.

Our method uses the fast primal-dual method from [8] for minimizing the relaxations $J_{A^t,w^t} + \delta_{D'}$.

For illustrating the strength of our method we present three datasets for the Potts model. The datasets are explained in greater detail in [11] and are available on the accompanying website [2]. The first dataset contains segmentation problems, for which we are given a few prototypical color vectors. The distance between each pixel's color value and each prototypical vector is measured, thereby obtaining the local data term $c(a)$ in (1.3). The regularization strength $\alpha_{a,b}$ in (1.3) was set uniformly. See Table 1 for the results. The number of classes is written in parentheses behind the instance name and the numbers denote the percentage of persistently labelled variables. The image dimensions are usually 360×240 or slightly less. We have also included a time plot for the clownfish dataset in Table 1, see Fig. 2.

Table 1. Results for segmentation problems with prototypical color vectors. Entries in the table denote the percentage of persistently labelled variables. The numbers in parentheses behind the dataset name denote the number k of classes.

Dataset	Our method	KMQPBO	KMQPBO100	Kovtun	MQPBO
clownfish (12)	**0.9852**	0.7659	0.9495	0.7411	0.0467
crops (12)	**0.9308**	0.6486	0.8803	0.6470	0.0071
fourcolors(4)	**0.9993**	0.6952	0.7010	0.6952	0.0
lake (12)	**0.9998**	0.7613	0.9362	0.7487	0.0665
palm (12)	**0.8514**	0.6866	0.7192	0.6865	0.0
penguin (8)	**0.9999**	0.9240	0.9471	0.9199	0.0103
peacock (12)	0.1035	0.0559	**0.1234**	0.0559	0.0
snail (3)	**0.9997**	0.9786	0.9819	0.9778	0.5835
strawberry-glass (12)	**0.9639**	0.5502	0.5997	0.5499	0.0

The second dataset consists of segmentation problems of a simulated brain scan with 5 prototypical vectors. The brain images were generated with the simulator [1]. The local data terms $c(a)$ in (1.3) were computed as for the first dataset and the regularization strength $\alpha_{a,b}$ in (1.3) was set uniformly as well. Instances can become very huge due to the volumes being three-dimensional. See Table 2 for results. The dimensions of the problems are denoted in the left column, while the entries denote the percentage of the variables determined to be persistent.

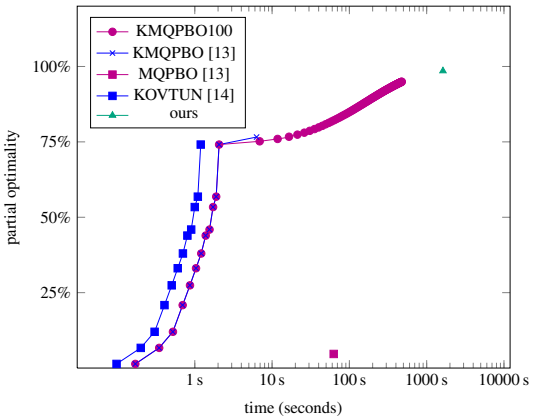


Fig. 2. Percentage of partial optimal variables of the 5 compared methods over time for the clownfish dataset. The method of Kovtun provides partial optimality very fast. KMQPBO100 finds more and more partially optimal variables, which illustrates that the performance of MQBPO depends on the label order. While our pruning method gives the best result, our current research implementation is not competitive with respect to time.

Table 2. Results for the simulated brain scan dataset. Entries in the table denote the percentage of persistently labelled variables. The numbers in the left column denote the image dimensions. The number k of classes is 5. Entries denoted by † indicate that the instance could not be solved with the specified method for implementation reasons.

Dataset	Our method	KMQPBO	KMQPBO100	Kovtun	MQPBO
$181 \times 217 \times 20$	0.9968	0.9993	**0.9994**	0.9235	0.3886
$181 \times 217 \times 26$	0.9969	**1**	0.9996	0.9322	0.3992
$181 \times 217 \times 36$	**0.9967**	†	†	0.9363	0.4020
$181 \times 217 \times 60$	**0.9952**	†	†	0.9496	0.4106

The third dataset consists of object segmentation problems with the local data terms $c(a)$ in (1.3) denoting the probability of pixels to belong to object classes. The regularization strength $\alpha_{a,b}$ in (1.3) is chosen to be inversely proportional to the image gradient. The data was taken from [4]. It turned out that the relaxation we use was already tight for the data. Hence we could determine all variables in the initial run. This illustrates experimentally, that the relaxations used in Kovtun's method and in MQPBO are less tight than our relaxation. See Table 3 for results.

Most often we could label over 95% of all variables persistently with our method and outperform the other tested approaches. Note that *we can use an arbitrary algorithm for solving the problem (2.2) in contrast to the approaches based on roof duality*. It is very noteworthy, that KMQPBO and KMQPBO100 outperform our approach in the brain scan dataset as well as in one instance of the color segmentation dataset. Although the relaxation we use is tighter than MQPBO's, all the integral variables MQPBO obtains are persistent. In contrast, only a subset of the integral variables of the solution to the relaxation we use are found persistent. Hence, it may occur that in some instances

Table 3. Results for the object segmentation dataset. Entries in the table denote the percentage of persistently labelled variables. The numbers in the left column the number of classes k.

Dataset	Our method	KMQPBO	KMQPBO100	Kovtun	MQPBO
Plane (4)	**1**	0.9833	0.9833	0.9833	0.0002
Bikes (5)	**1**	0.9570	0.9570	0.9569	0.0
Road (5)	**1**	0.9579	0.9579	0.9579	0.0
Building (7)	**1**	0.8051	0.8053	0.8051	0.0
Car (8)	**1**	0.9902	0.9904	0.9902	0.0002

(K)MQPBO(100) can label more variables persistently although our approach yields more integral variables which however cannot be proved persistent. The same reasoning applies to Kovtun's method, so possibly some variables could be found persistent in Kovtun's method and hence also in KMQPBO and KMQPBO100 which could not be verified to be persistent in our approach.

Kovtun's method from [14] is very fast, usually faster than solving the relaxation of problem (1.3). Therefore using a layered approach by first applying Kovtun's Partial Optimal Labeling Search from [14] and then applying our approach on the remaining variables will result in at least the same number of persistent variables while still retaining a very fast runtime. For properly comparing the approaches however, we have used our method on its own.

4 Conclusion and Outlook

We have presented a method for finding persistent variables for the Potts model, which outperforms other approaches to this problem with respect to the number of persistent variables found. The presented method can use an arbitrary algorithm for minimizing a relaxed labeling problem and generalizes to tighter relaxations as well.

In future we will address the problem of finding persistent variables for arbitrary graphical models and the discretized minimal partition problem with better discretizations.

Acknowledgments. This work has been supported by the German Research Foundation (DFG) within the program "Spatio-Temporal Graphical Models and Applications in Image Analysis", grant GRK 1653. The authors also would like to thank Marco Esquinazi for helpful discussions.

References

1. Brainweb: Simulated brain database, `http://brainweb.bic.mni.mcgill.ca/brainweb/`
2. OpenGM inference library, `http://hci.iwr.uni-heidelberg.de/opengm2/`
3. Alahari, K., Kohli, P., Torr, P.H.S.: Reduce, reuse & recycle: Efficiently solving multi-label MRFs. In: CVPR (2008)

4. Alahari, K., Kohli, P., Torr, P.H.S.: Dynamic hybrid algorithms for MAP inference in discrete MRFs. PAMI 32(10), 1846–1857 (2010)
5. Boros, E., Hammer, P.L.: Pseudo-Boolean optimization. Discrete Applied Mathematics 123(1-3), 155–225 (2002)
6. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: ICCV, pp. 26–33. IEEE Computer Society (2003)
7. Chambolle, A., Cremers, D., Pock, T.: A convex approach for computing minimal partitions. Technical report, Centre des Mathematiques Appliquees, Ecole Polytechnique, Palaiseau, Paris, France (2008)
8. 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)
9. Hammer, P.L., Hansen, P., Simeone, B.: Roof duality, complementation and persistency in quadratic 0-1 optimization. Math. Programming 28, 121–155 (1984)
10. Kahl, F., Strandmark, P.: Generalized roof duality. Discrete Applied Mathematics 160(16-17), 2419–2434 (2012)
11. 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: CVPR (2013)
12. Kleinberg, J., Tardos, É.: Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. J. ACM 49(5), 616–639 (2002)
13. Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov, V., Torr, P.: On partial optimality in multi-label MRFs. In: ICML, pp. 480–487 (2008)
14. Kovtun, I.: Partial optimal labeling search for a NP-hard subclass of (max,+) problems. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 402–409. Springer, Heidelberg (2003)
15. Lellmann, J., Schnörr, C.: Continuous multiclass labeling approaches and algorithms. SIAM J. Imag. Sci. 4(4), 1049–1096 (2011)
16. Nemhauser, G.L., Trotter, L.E.: Vertex packings: Structural properties and algorithms. Mathematical Programming 8, 232–248 (1975), doi:10.1007/BF01580444
17. Rother, C., Kolmogorov, V., Lempitsky, V.S., Szummer, M.: Optimizing binary MRFs via extended roof duality. In: CVPR (2007)
18. Shekhovtsov, A., Kolmogorov, V., Kohli, P., Hlavac, V., Rother, C., Torr, P.: LP-relaxation of binarized energy minimization. Research Report CTU–CMP–2007–27, Czech Technical University (2008)
19. Sontag, D.: Approximate Inference in Graphical Models using LP Relaxations. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science (2010)
20. Windheuser, T., Ishikawa, H., Cremers, D.: Generalized Roof Duality for Multi-Label Optimization: Optimal Lower Bounds and Persistency. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 400–413. Springer, Heidelberg (2012)

5 Appendix

Lemma 1

Proof. Let $\overline{u} : \Omega \backslash A \to \mathscr{E}_k$ be optimal for the functional

$$J_{\Omega \backslash A}(u) - \sum_{a \in (\Omega \backslash A), b \in A, (a,b) \in E} 2\alpha_{a,b} \langle w(b), u(a) \rangle + \delta_D. \tag{5.1}$$

Then

$$\tilde{u}(a) = \begin{cases} u^*(a) \ , a \in A \\ \overline{u}(a) \ , a \notin A \end{cases} \tag{5.2}$$

is optimal for $J + \delta_D$. Let $u' : \Omega \to \mathscr{E}_k$ be another labeling. If we will show, that $J(\tilde{u}) \leq J(u')$, it will prove the lemma. Indeed, taking into account that $u^*(a) = w(a)$, $a \in \partial A$ we have

$$\begin{aligned} & J(\tilde{u}) \\ &= J_A(u^0) + J_{\Omega\setminus A}(\overline{u}) + \textstyle\sum_{b\in\Omega\setminus A, a\in A,(a,b)\in E} \sum_{l=1}^k \alpha_{a,b} |\tilde{u}^l(a) - \tilde{u}^l(b)| \\ &= J_A(u^0) + J_{\Omega\setminus A}(\overline{u}) + \textstyle\sum_{b\in\Omega\setminus A, a\in A,(a,b)\in E} \sum_{l=1}^k \alpha_{a,b} |w^l(a) - \tilde{u}^l(b)| \\ &= J_A(u^0) + J_{\Omega\setminus A}(\overline{u}) + \textstyle\sum_{b\in\Omega\setminus A, a\in A,(a,b)\in E} 2\alpha_{a,b} (1 - \mathbb{1}_{\overline{u}(b)=w(a)}) \\ &= J_A(u^0) + J_{\Omega\setminus A}(\overline{u}) + \textstyle\sum_{b\in\Omega\setminus A, a\in A,(a,b)\in E} 2\alpha_{a,b} \langle w(a), u^0(a) - \overline{u}(b)\rangle \\ &\leq J_A(u'_{|A}) + J_{\Omega\setminus A}(u'_{|\Omega\setminus A}) + \textstyle\sum_{b\in\Omega\setminus A, a\in A,(a,b)\in E} 2\alpha_{a,b} \langle w(a), u'(a) - u'(b)\rangle \\ &= J_A(u'_{|A}) + J_{\Omega\setminus A}(u'_{|\Omega\setminus A}) + \textstyle\sum_{b\in\Omega\setminus A, a\in A,(a,b)\in E} 2\alpha_{a,b} (\mathbb{1}_{w(a)=u'(a)} - \mathbb{1}_{w(a)=u'(b)}) \\ &\leq J_A(u'_{|A}) + J_{\Omega\setminus A}(u'_{|\Omega\setminus A}) + \textstyle\sum_{b\in\Omega\setminus A, a\in A,(a,b)\in E} \alpha_{a,b} \sum_{l=1}^k |u'(a) - u'(b)| \\ &= J(u') , \end{aligned} \tag{5.3}$$

which finalizes the proof.

Corollary 1

Proof. $u^* \in \operatorname{argmin} J_{A,w} + \delta_{D'}$ and $u^* \in D \Rightarrow u^* \in \operatorname{argmin} J_{A,w} + \delta_D$. Now apply Lemma 1 and get statement of the corollary.

To prove Theorem 1 we will require the following three lemmas.

Lemma 2. *Let $\alpha \in \mathscr{E}_k$, $\beta \in \Delta_k$. Then*

$$2\langle \alpha, \alpha - \beta \rangle = \sum_{l=1}^k |\alpha^l - \beta^l| \tag{5.4}$$

Proof. Without loss of generality assume $\alpha = e_i$. Then

$$\begin{aligned} 2\langle \alpha, \alpha - \beta \rangle = 2 - 2\beta^i = 1 - \beta^i + 1 - \beta^i \\ = 1 - \beta^i + \textstyle\sum_{l\neq i} b^l = |1 - \beta^i| + \sum_{l\neq i} |b^l| = \sum_{l=1}^k |\alpha^l - \beta^l| \end{aligned} \tag{5.5}$$

Lemma 3. *Let $\alpha, \beta \in \Delta_k$ and $e_i \in \mathscr{E}_k$. Then*

$$2\langle e_i, \alpha - \beta \rangle \leq \sum_{l=1}^k |\alpha^l - \beta^l|. \tag{5.6}$$

Proof.

$$\begin{aligned} 2\langle e_i, \alpha - \beta \rangle = 2(\alpha^i - \beta^i) = (\alpha^i - \beta^i) + (1 - \textstyle\sum_{l\neq i} a^l) - (1 - \sum_{l\neq i} \beta^l) \\ = (\alpha^i - \beta^i) + \textstyle\sum_{l\neq i} (\beta^l - a^l) \leq \sum_{l=1}^k |\alpha^l - \beta^l|. \end{aligned} \tag{5.7}$$

Note that equality holds in (5.6) iff $a^i \geq \beta^i$ and $\alpha^j \leq \beta^j$ for all $j \neq i$.

Lemma 4. *Let $A \subset \Omega$, $\overline{u}\colon A \to \mathscr{E}_k$ and $\overline{w}\colon \partial A \to \mathscr{E}_k$ satisfy the persistency criterion of Corollary 1.*

Then for any $B \supset A$ and boundary condition $\tilde{w}$ satisfying $\tilde{w}(a) = \overline{w}(a)$ for all $a \in \partial A \cap \partial B$, such $\tilde{u}$ exists, that

$$\tilde{u} \in \operatorname{argmin}_u J_{B,\tilde{w}}(u) + \delta_{D'} \tag{5.8}$$

and $\tilde{u}_{|A} = \overline{u}$ and $\tilde{w}(a) = \tilde{u}(a)$ for $a \in \partial A$.

Proof. Let u' be optimal for the functional

$$J_{B\setminus A}(u) - \sum_{(a,b)\in E\colon\, b\in B\setminus A, a\in A} 2\alpha_{a,b}\langle w(a), u(b)\rangle + \sum_{(a,b)\in E\colon\, a\in B\setminus A, b\in B\setminus\Omega} 2\alpha_{a,b}\langle \tilde{w}(a), u(a)\rangle + \delta_{D'}. \tag{5.9}$$

Let

$$\tilde{u}(a) = \begin{cases} \overline{u}(a) & , a \in A \\ u'(a) & , a \in B\setminus A\,. \end{cases} \tag{5.10}$$

The lemma will be proved when we show that $\tilde{u} \in \operatorname{argmin}_u J_{B,\tilde{w}}(u) + \delta_{D'}(u)$. For this let $\hat{u}$ be arbitrary relaxed labeling from $\in \delta_{D'}$. The following inequalities then hold:

$$\begin{aligned}
& J_{B,\tilde{w}}(\tilde{u}) && (5.11)\\
=& J_A(\overline{u}) + J_{B\setminus A}(u') && (5.12)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in A, b\in B\setminus A} \sum_{l=1}^{k} \alpha_{a,b} |\overline{u}^l(a) - (u')^l(b)| && (5.13)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in B, b\in\Omega\setminus B} 2\alpha_{a,b}\langle \tilde{w}(a), \tilde{u}(a)\rangle && (5.14)\\
=& J_A(\overline{u}) + J_{B\setminus A}(u') && (5.15)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in A, b\in B\setminus A} \sum_{l=1}^{k} \alpha_{a,b} |\overline{u}^l(a) - (u')^l(b)| && (5.16)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle \overline{w}(a), \overline{u}(a)\rangle && (5.17)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in B\setminus A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle \tilde{w}(a), u'(a)\rangle && (5.18)\\
\overset{(*)}{=}& J_A(\overline{u}) + J_{B\setminus A}(u') && (5.19)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in A, b\in B\setminus A} 2\alpha_{a,b}\langle \overline{w}(a), \overline{u}(a) - u'(b)\rangle && (5.20)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle \overline{w}(a), \overline{u}(a)\rangle && (5.21)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in B\setminus A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle \tilde{w}(a), u'(a)\rangle && (5.22)\\
=& J_A(\overline{u}) + J_{B\setminus A}(u') && (5.23)\\
+& \textstyle\sum_{(a,b)\in E\colon\, a\in A, b\in\Omega\setminus A} 2\alpha_{a,b}\langle \overline{w}(a), \overline{u}(a)\rangle && (5.24)\\
-& \textstyle\sum_{(a,b)\in E\colon\, a\in A, b\in B\setminus A} 2\alpha_{a,b}\langle \overline{w}(a), u'(b)\rangle && (5.25)
\end{aligned}$$

$$+\sum_{(a,b)\in E:\ a\in B\setminus A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle\tilde{w}(a), u'(a)\rangle \tag{5.26}$$

$$\overset{(**)}{\leq} J_A(\hat{u}_{|A}) + J_{B\setminus A}(\hat{u}_{|B\setminus A}) \tag{5.27}$$

$$+\sum_{(a,b)\in E:\ a\in A, b\in\Omega\setminus A} 2\alpha_{a,b}\langle\overline{w}(a), \hat{u}(a)\rangle \tag{5.28}$$

$$-\sum_{(a,b)\in E:\ a\in A, b\in B\setminus A} 2\alpha_{a,b}\langle\overline{w}(a), \hat{u}(b)\rangle \tag{5.29}$$

$$+\sum_{(a,b)\in E:\ a\in B\setminus A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle\tilde{w}(a), \hat{u}(a)\rangle \tag{5.30}$$

$$= J_A(\tilde{u}_{|A}) + J_{B\setminus A}(\hat{u}_{|B\setminus A}) \tag{5.31}$$

$$+\sum_{(a,b)\in E:\ a\in A, b\in B\setminus A} 2\alpha_{a,b}\langle\overline{w}(a), \hat{u}(a) - \hat{u}(b)\rangle \tag{5.32}$$

$$+\sum_{(a,b)\in E:\ a\in A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle\overline{w}(a), \hat{u}(a)\rangle \tag{5.33}$$

$$+\sum_{(a,b)\in E:\ a\in B\setminus A, b\in\Omega\setminus B} 2\alpha_{a,b}\langle\tilde{w}(a), \hat{u}(a)\rangle \tag{5.34}$$

$$\overset{(***)}{\leq} J_A(\hat{u}_{|A}) + J_{B\setminus A}(\hat{u}_{|B\setminus A}) \tag{5.35}$$

$$+\sum_{(a,b)\in E:\ a\in A, b\in B\setminus A} \sum_{l=1}^{k} \alpha_{a,b}|\hat{u}^l(a) - \hat{u}^l(b)| \tag{5.36}$$

$$+\sum_{(a,b)\in E:\ a\in B, b\in\Omega\setminus B} 2\alpha_{a,b}\langle\tilde{w}(a), \hat{u}(a)\rangle \tag{5.37}$$

$$= J_{B,\tilde{w}}(\hat{u}), \tag{5.38}$$

where $(*)$ is due to lemma 2, $(**)$ is due to the optimality of $\overline{u}$ and u' for the respective functionals and $(***)$ is due to lemma 3.

Theorem 1

Proof. We prove the statement that A^* is the greatest persistent set by showing the following claim to hold true:

Claim: Assume that for some $\overline{A} \subset \Omega$ there exists a persistent labeling $\overline{u}$. Then in each iteration of the algorithm $\overline{A} \subset A^i$ holds. Furthermore $\overline{u} = u^*_{|\overline{A}}$ and hence $\overline{w}(a) = w^*(a)$ for $a \in \partial\overline{A} \cap \partial A^*$, where $w^* = u^*_{|\partial A^*}$.

In the initialization step the claim clearly holds true, since $A^0 = \Omega$ and Lemma 4 ensures that there exists $\tilde{u}^0 \in \operatorname{argmin} J_\Omega + \delta_{D'}$ such that $\overline{u} = \tilde{u}^0_{|\overline{A}}$ and $\overline{w}(a) = \tilde{w}^0(a)$ for all $a \in \partial\overline{A} \cap \partial\Omega = \varnothing$ is an empty condition. Finally assumption 1 gives us that there is only one such minimizer $\tilde{u}^0$, so the claim holds initially.

Now assume the claim to hold for $i-1$. We need to show that it also holds for i. For this just invoke Lemma 4 with $A = \overline{A}$, $B = A^{i-1}$ and $\tilde{w} = \tilde{w}^{i-1}$. The conditions of Lemma 4 hold by assumption on $i-1$. Lemma 4 now ensures existence of $\tilde{u}^i \in \operatorname{argmin}_u J_{A^{i-1},\tilde{w}^{i-1}}(u) + \delta_{D'}(u)$ with the required properties. Again by Assumption 1, $\tilde{u}^i$ is unique, so we are done.

Inspecting the proof of the claim above, we see that Assumption 1 is necessary because otherwise the labels for nodes in $\overline{A}$ could possibly change during iterations of the algorithm or be convex sums of optimal persistent labellings in which case they would be discarded from the sets A^i at some point.

Wimmelbild Analysis with Approximate Curvature Coding Distance Images

Julia Bergbauer[1] and Sibel Tari[2,⋆]

[1] TU Munich, Garching bei Muenchen, DE-85747
[2] Middle East Technical University, Ankara, TR-06800
stari@metu.edu.tr

Abstract. We consider a task of tracing out target figures hidden in teeming figure pictures come to known as *Wimmelbild(er)*. Wimmelbild is a popular genre of visual puzzles; a timeless classic for children, artists and cognitive scientists. Particularly suited to the considered task, we propose a diffuse representation which serves as a heuristic approximation mimicking curvature coding distance images. Curvature coding distance images received increased attention in recent years. Typically, they are computed as solutions to variants of Poisson PDE. The proposed approximation is based on erosion of the white space (background) followed by isotropic averaging, hence, does not require solving a PDE.

Keywords: Poisson PDE and its variants, level sets, non-linear diffusion, figure-hunt games, teeming figure pictures, applications of variational and PDE methods.

1 Introduction

Level set methods have been successfully applied to knowledge based segmentation; bringing in an ability to deal with physically corrupted incomplete data. Most typically, shape knowledge is coded via signed distance transform, embedding the $1-D$ shape boundary as the zero-level set of a function defined on a connected bounded open subset of $\Re^2$ [9,8]. It is also possible to replace the sharp interface model in level set based segmentation methods with diffuse ones [12,10], decaying exponentially and coding curvature.

In recent years, there is a growing interest in exponentially decaying curvature coding distance images with examples including [1,5,2,11,12].

The idea is to replace a point set S denoting possibly incomplete object boundaries with a smooth function $\nu : R \to \Re$ where $R \subset \Re^2$ is an open connected bounded set s.t. $R \supset S$. The function ν is the solution of

$$\nabla \cdot (\nabla \nu) - \alpha^2 \nu = 0 \tag{1}$$

subject to boundary conditions:

$$\nu\Big|_S = 1 \quad \text{and} \quad \partial_\eta \nu\Big|_{\partial R} = 0 \tag{2a,b}$$

⋆ Corresponding author.

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 489–500, 2013.

where ∂R denotes the boundary of R and $\partial_\eta \nu$ the derivative of ν in the normal direction to ∂R. Interestingly, from [12],

$$\nu(x,y) \approx \frac{1}{2\alpha^2}\left(2\alpha + curv(x,y)\right)\frac{\partial \nu}{\partial \eta} + O\left(\frac{1}{\alpha^3}\right) \tag{3}$$

where $curv(x,y)$ is the curvature of the level curve passing through the point (x,y) at (x,y). Eq. 3 indicates a reciprocal relationship between the level curve curvature and the gradient of ν. That is, unlike the usual distance transform, ν is an implicit coder of the curvature, a valuable geometric feature, without explicit estimation of higher order derivatives. (One of the original goals in proposing ν was to bridge low level and high level vision [11,12]).

In this paper, in the setting of a specific task, namely tracing out target figures hidden in teeming figure pictures, we present a much simplistic way of obtaining an analogous (curvature dependent) behavior in a band around S. Our computation does not require the computation of the entire ν function on the entire domain R by solving a PDE.

Preliminary experiments are highly encouraging and indicate the potential of the approach.

2 The Problem Setting

Wimmelbild is a popular genre of visual puzzles. It means teeming figure picture. Abundant masses of small figures are brought together in complex arrangements to make one scene in a Wimmelbild, to be used for a figure hunt game (Fig. 1).

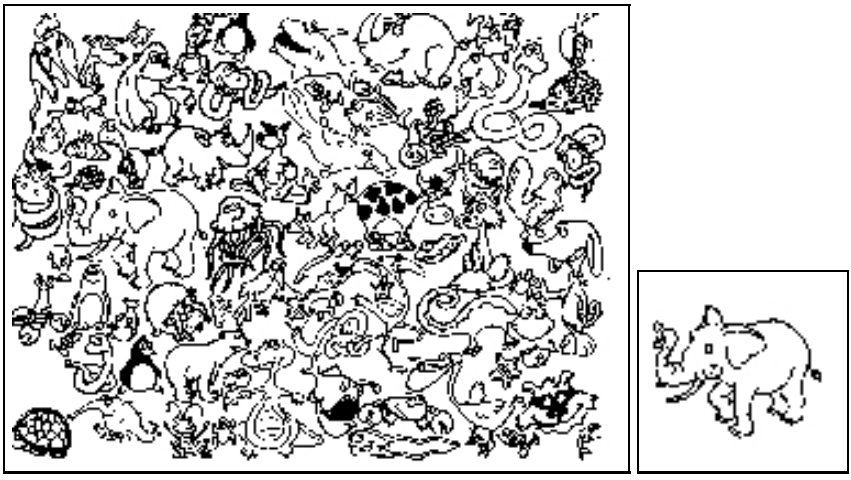


Fig. 1. An elephant in a zoo. A sample wimmelbild (left) and an elephant figure (right) to hunt for in the Zoo Wimmelbild.

Figure-hunt games have been a timeless classic for children, artists and cognitive scientists. As early as 1926 Kurt Gottschald experimented with intentionally designed hidden figures – simple drawings where simple shapes such as polygons are embedded within more complex organizations– to study the influence of experience on perception and the extent to which wholes influence the perception of parts [4].

There are of course colorful versions of these genre of visual puzzles. Interestingly, sketch-like black and white versions of teeming figure pictures pose even a bigger challenge than their colored counterparts:

Firstly, during a hunt for the elephant, suppose we somehow landed on the correct location in the picture yet hypothesized a wrong scale, say a smaller scale, such that the hypothesized elephant fits inside the white space (background) surrounded by the figural loci of the actual elephant, yielding a matching cost of zero; providing, therefore, no clue as to whether we are in the right neighborhood, on a white space, or on a region containing a figure which has no common elements with the elephant. Whereas each point on a dense picture (be it color or gray) is informative, information in sketch-like binary pictures is concentrated on loci of lower dimension, *i.e.*, curves denoting figural loci.

Secondly, as a consequence of lower dimensionality of the figural loci, final pictures could get extremely complicated. Observe that it is impossible to trace out the hidden clover in Fig. 2 using Gestalt parsing rules. The final picture is not simply a superposition of figures. Indeed, figures are first added, but then thresholded.

Therefore, the task can not be simply cast as locating a subpicture within a whole picture.

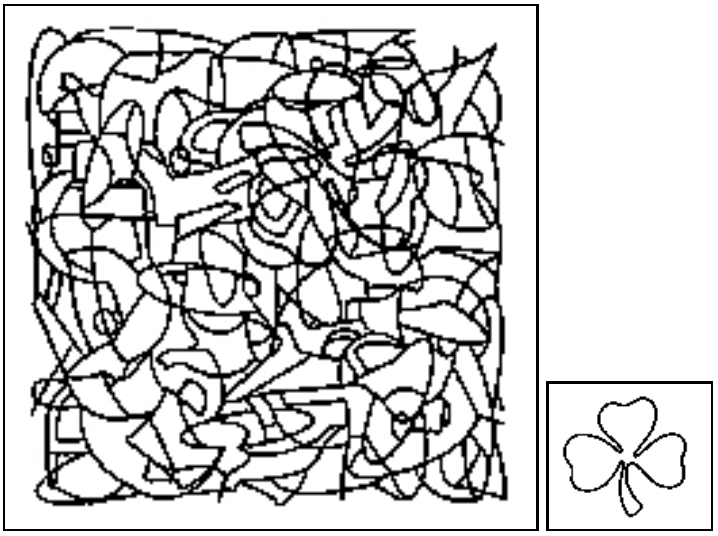


Fig. 2. Can you trace out the hidden clover?

Related Work. To the best of our knowledge, Wimmelbild(er) has not been studied within the variational and PDE methods community before. Saarbruecken group recently presented an inpainting based steganography application [7]. The nature of their problem hence their solution strategy, however, are quite different. Their goal is to hide a secret image by embedding it into arbitrary cover images. Both the secret and the cover are dense images and recovery of the secret is possible only via a password. Therefore, ordinary observer can not detect whether an image contains a secret or not. In reconstructing frescos, Fornasier et.al. [3] addressed the problem of locating small fragments within a whole; for each small piece of plaster that still showed an element of the design of the fresco, they were able to find where it belonged. Such approaches can not be used due to non-additive and non-linear nature of the binary drawings that we consider (such as Fig. 2). Curvature coding distance fields are becoming more and more commonplace. Theoretical investigations can be found in [12,1,5,2], among others. They have been used to address a variety of shape related problems, including skeleton computation and knowledge based segmentation as well as other vision problems.

3 Our Approach

The key idea is to propagate information restricted to figural loci to neighboring areas so that it becomes possible to know whether a location is close or far away from the desired locations.

We start by uniformly eroding the white space, or equivalently, dilating the figure. Hence, the drawings (both the wimmelbild and the target figure) become thicker. Then, we diffuse by computing a local isotropic average. It is sufficient to compute the local average only for the points falling on the thickened figural loci or in a slightly wider band surrounding it. This transforms the sketch-like binary drawing to a gray-tone picture which may be referred as a diffuse drawing. This diffuse drawing is an approximation to a curvature coding distance image.

The rightmost column in Fig. 3 depicts the outcome of the described procedure. The leftmost column is the original drawing and the fourth column from the left is the thickened drawing. If the averaging and the dilation radii are identical, the highest value is attained on the figural loci; from thereof values decrease as a function of distance in the normal direction. Thus, diffusion produces iso-intensity contours, each following the figural loci from a fixed distance. The lower the intensity, the further away the iso-intensity curve from the figural loci. The two columns in the middle (second and third from the left) depict the results of two different local isotropic averaging applied to the original thin drawing. There, one can not observe the distance-coding behaviour, *i.e.*, the initial thickening is a crucial step.

What makes the iso-intensity contours of our diffuse drawings further interesting is that they implicitly code curvature: Let us select two locations on the white space of the drawing such that the nearest figural point to the first location has a high curvature, while on the contrary, the nearest figural point to the

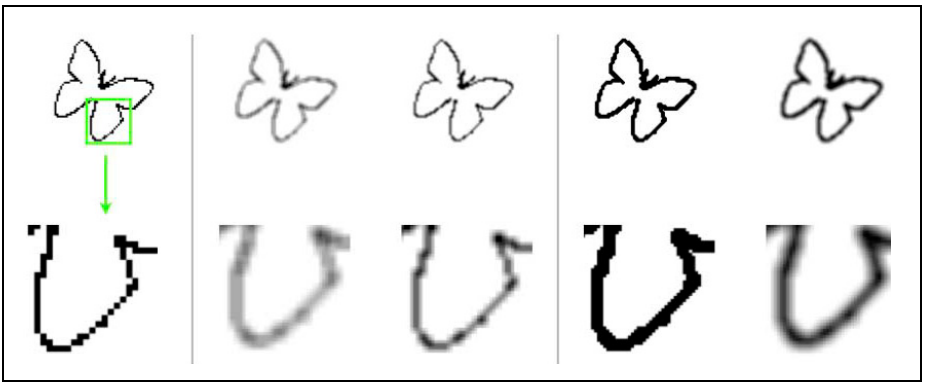


Fig. 3. Mimicking curvature coding distance via erosion of the white space (dilation of the figural loci) followed by averaging

second location is on a flat part. Let us make sure, however, that their distances to their respective nearest figure points are equal. That is each of the points have the same normal distance to the drawing. For example, let us consider the two locations marked by the intersection points of the two crosses in Fig. 4.

Note that even though the normal distances from each location to the figural loci are identical, the distances in the tangential directions are not.

In the tangential direction, indeed, the first location (marked by the cross on the upper right in Fig. 4) is closer to the figural loci, causing the average value at that location to be higher compared to the average value at the second location (marked by the cross on the lower left in Fig. 4) . Consequently, the level curve passing through the second location will not pass through the first location but through a location further down in the direction of inward normal.

As a result, within a band surrounding the figural loci, our diffuse drawing (obtained by dilation followed by isotropic diffusion) mimics a curvature coding

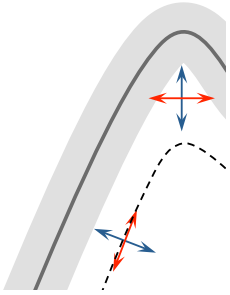


Fig. 4. Two locations of identical normal distance to the figural loci are marked by the two crosses. In the tangential direction, however, the location of which nearest figural point has a higher curvature is closer to figural loci.

distance field similar to the ν, the solution of a damped Poisson PDE [12]. In Fig. 5, we compare our diffuse drawing to a usual distance image (i.e. Eikonal PDE with constant right hand side) restricted to a band surrounding the figural loci. Whereas the effects of discretization noise remains (even amplified) in the usual distance image, the iso-intensity contours in our diffuse model *smoothly* follow the boundary.

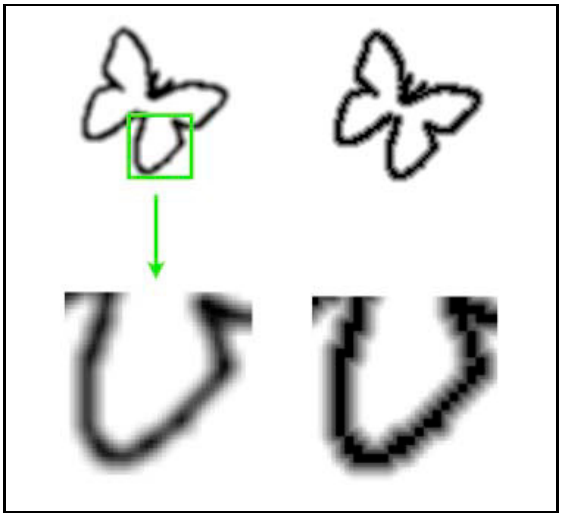


Fig. 5. Curvature coding distance image obtained by thickening + isotropic averaging (left) versus the usual distance image (right)

We avoid solving Poisson PDE or variants for two reasons. Firstly, our approximation is both easier and faster to compute. But more importantly, a Poisson based distance field, being the steady state solution to a biased diffusion equation, $\frac{\partial \nu}{\partial \tau} = \nabla \cdot (\nabla \nu) - \alpha^2 \nu$, is too much influenced by long-range interactions among opposing boundaries. This may be detrimental if several figural loci overlap as in Fig. 2.

Once the drawings (both the wimmelbild and the target figure to be hunted for) are converted to diffused forms, the best match is formulated as finding the deformation parameters (e.g. scale, location and orientation) that yield the best match. The matching cost is measured as the sum of the gray value differences between the wimmelbild and the target figure. Of course, the sum is taken over those locations that fall within the band surrounding the figural loci within which the diffuse field has been constructed. Moreover, the cost is normalized by dividing it to the number of locations contributed to its computation.

In Fig. 6, the cost calculation is illustrated. Observe how matching cost becomes informative when raw binary figures are replaced with diffuse ones.

Once the matching cost is defined, the optimizing parameters are determined via a probabilistic algorithm which returns multiple solutions. The importance of using such an algorithm is discussed in [6]. We use genetic algorithms based optimization which is readily available in Matlab environment. It minimizes an energy functional by varying its input variables. It is called via the comment:

$$[variables, energy] = \mathbf{ga}(\mathbf{fitnessfcn}, nvars, [], [], [], [], lb, ub, [], IntCon);$$
$$\mathbf{fitnessfcn} = @energyfunctional, pattern, shape, diffusingparam;$$

The *output* includes the determined value of the vector of *variables* for rotation, scaling and translation as well as the value of the corresponding minimal *energy*. As *input* it requires a **fitnessfcn**. This has to be a functional which takes the *variables* as the first input. Its single output has to be the value of the corresponding energy. Furthermore, the parameters *nvars* (number of variables), *lb* (lower bound for variables), *ub* (upper bound for variables) and *IntCon* (an interval containing the indices of variables that should be integers) have to be specified.

Each call to **ga** returns the best fit obtained after a certain number of trials. Due to randomization, multiple calls to **ga** generate multiple hypotheses. We depict all the generated hypotheses after an ordering based on the matching cost.

4 Experimental Results

In this section, we will present our experiments with the two wimmelbilder:

- hunt for the elephant in the Zoo Wimmelbild;
- hunt for the clover in a heap of overlapping contours.

Prior to that, however, we test our approach on simpler drawings of repeated patterns: Mandalas. One purpose of these supporting illustrations is to examine the robustness of the method to scale and pose variations. The second purpose is to observe whether the genetic algorithm returns correct fits more often than the wrong ones.

First, to observe the robustness with respect to scale, we consider a composition of circles of varying size. One of the circles (shown in blue color on the left column of Fig. 7) is selected as the target figure. On the right column, we depict all the circles detected after several runs of the genetic algorithm. Observe that the method can handle scale variations.

Second, to observe the robustness with respect to pose, we consider a simple Mandala pattern (Fig. 8). The two subfigures extracted from it (on the right) are to be used as targets. We expect to find 8 instances of the butterfly target and 4 instances of the second target.

The results are depicted in the left column of Fig. 9. The top row depicts the results for the *butterfly* target and the second row for the second target. The **ga** is invoked around 100 times and all of the 100 results are displayed. Each

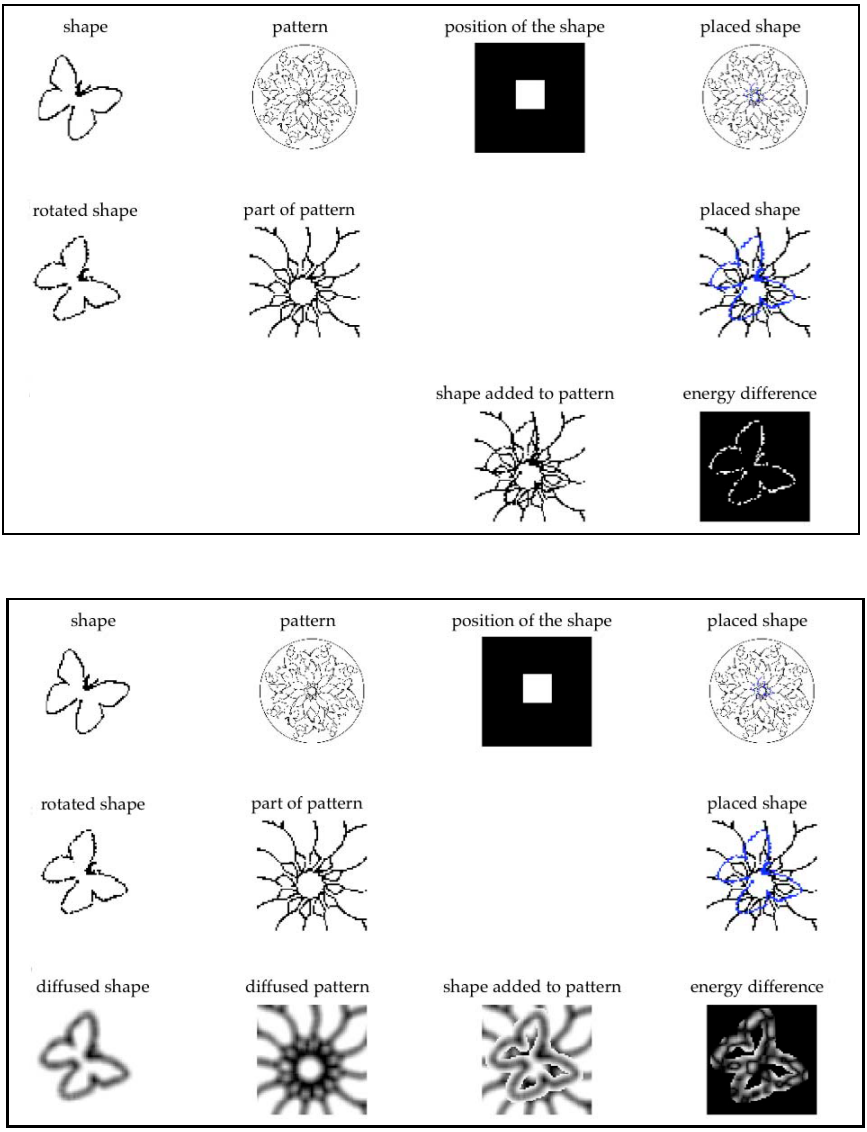


Fig. 6. Illustration of cost calculation. When raw binary drawings are used (top), it is hard to tell how well the target figure is located. Observe how matching cost becomes informative when raw binary figures are replaced with diffuse ones (bottom).

outcome of the **ga** is tone-coded based on the matching cost. Lighter the tone, lower the matching cost hence better the fit. The correct fits are found regardless of their pose. Moreover, the matching cost is significantly lower for the correct fits. This indicates the robustness of the representation to pose changes.

Third, we have tested whether the good fits (those of lower matching cost) are obtained more often than the bad fits. This is important as the algorithm is not a

deterministic one. We have performed independent **ga** runs, each run producing several hypotheses. We have then computed the average of the batches of independent runs. As the results shown in the right column compellingly demonstrate, **ga** has a tendency to return good fits more often than the bad ones.

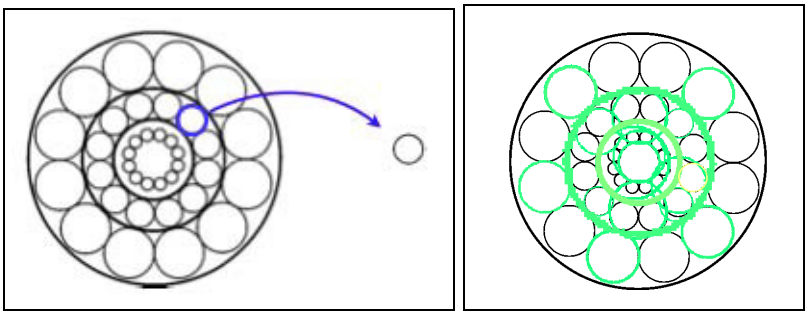


Fig. 7. Circles of varying scale

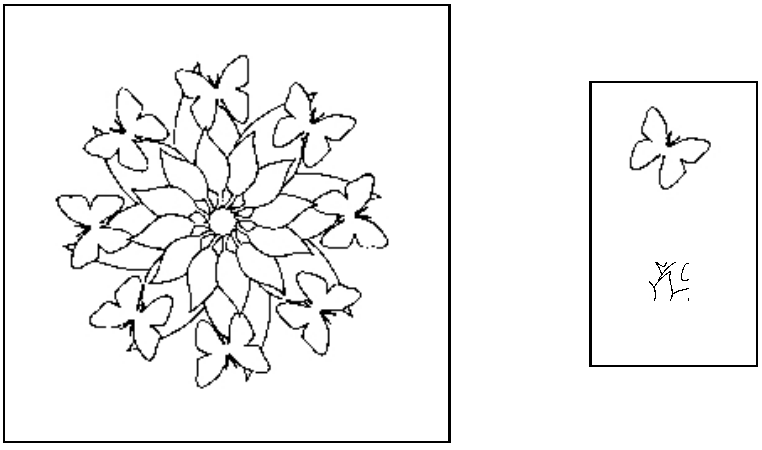


Fig. 8. A Mandala pattern and two subfigures extracted from it

Finally, we present our two figure hunt experiments. The results of the elephant hunt is given in Fig. 10. On the left, multiple hypotheses produced by **ga** are tone-mapped. The correct hypothesis is significantly lighter than all the others. On the right, the best match is shown in red, whereas the others in shades of green.

Fig. 11 depicts the results of the clover hunt. The generated hypotheses are depicted on the top left. The hypothesis with the lowest matching cost is depicted on the top right. The best hypothesis is traced with red marker on the original drawing (bottom left), and the original drawing is repeated (bottom right) as a convenience to the reader.

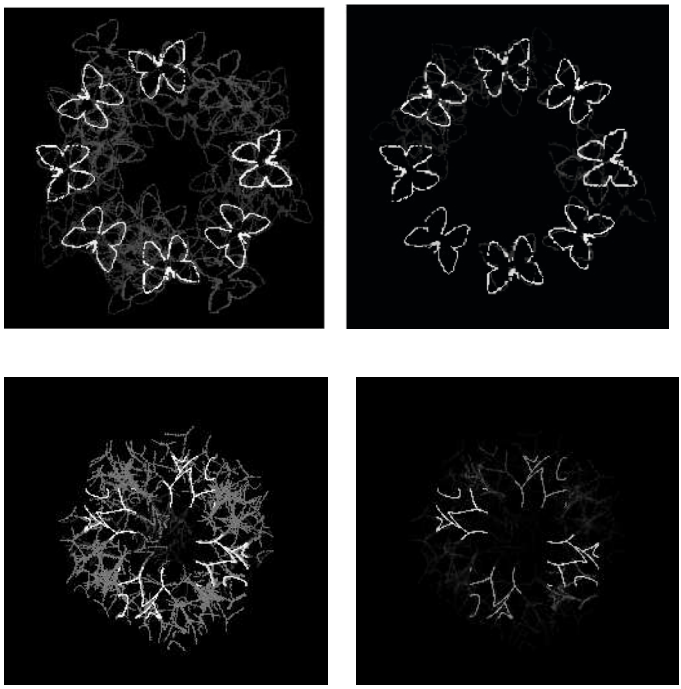


Fig. 9. (Left) several hypotheses returned by the genetic algorithm. Gray-tones reflect the matching cost. The lighter the tone, the better the fit.

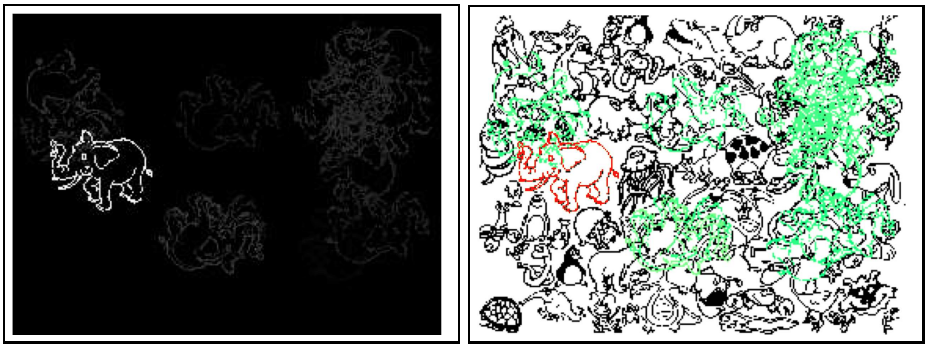


Fig. 10. Hunt for the elephant

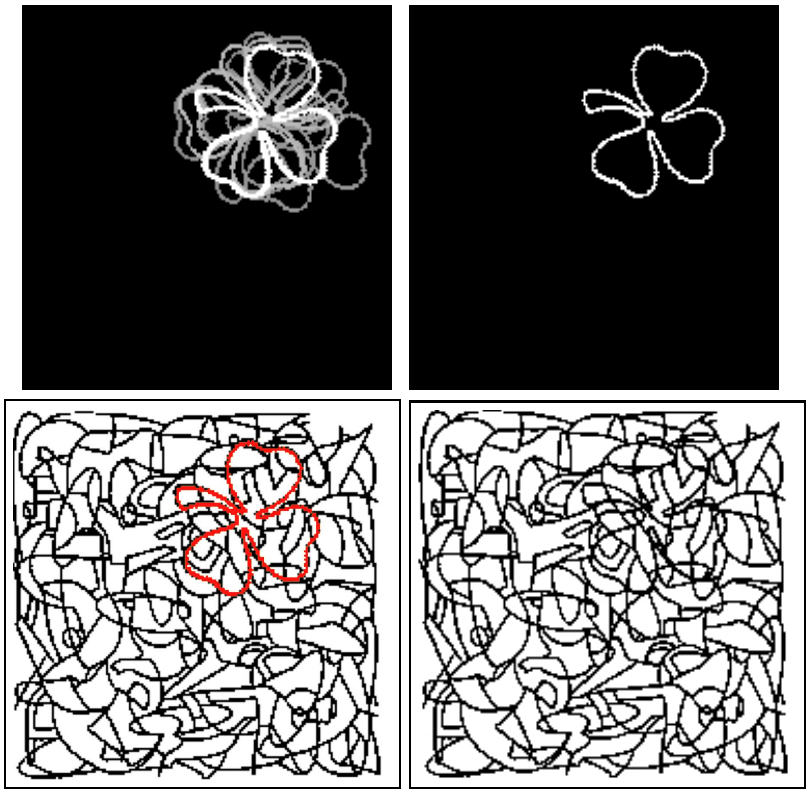


Fig. 11. Hunt for the clover

5 Summary and Conclusion

We have addressed the task of tracing out target figures in sketch-like binary teeming figure pictures. Particularly suited to the task, we propose a simple heuristic for generating diffuse drawings that imitate curvature coding distance images which are typically computed as solutions to elliptic PDEs. Our work extends the applications of diffusion based ideas to an interesting problem.

Acknowledgements. The work of JB has been completed in March 2012 as a part of MA5313 course offered by ST, while ST was spending her sabbatical at Folkmar Bornemann Group at TU Munich. At present, JB is affiliated with D. Cremers Group at the same institution. Subsequent work in writing the paper has been completed at the Middle East Technical University.

ST thanks to Folkmar Bornemann for providing a truly scientific environment and flawless support during the time of her stay, and extends her gratitude to the Alexander von Humboldt Foundation for generous financial support.

References

1. Aubert, G., Aujol, J.F.: Poisson skeleton revisited: A new mathematical perspective. J. Math. Imaging Vis. (2012)
2. Dimitrov, P., Lawlor, M., Zucker, S.W.: Distance images and intermediate-level vision. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 653–664. Springer, Heidelberg (2012)
3. Fornasier, M., Toniolo, D.: Fast, robust and efficient 2d pattern recognition for re-assembling fragmented images. Pattern Recognition 38(11), 2074–2087 (2005)
4. Gottschald, K.: Ueber den Einfluss der Erfahrung auf die Wahrnehmung von Figuren. Psychologische Forschung 8, 261–317 (1926)
5. Gurumoorthy, K.S., Rangarajan, A.: A schrödinger equation for the fast computation of approximate euclidean distance functions. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 100–111. Springer, Heidelberg (2009)
6. Keles, H.Y., Ozkar, M., Tari, S.: Weighted shapes for embedding perceived wholes. Environment and Planning B: Planning and Design 39, 360–375 (2012)
7. Mainberger, M., Schmaltz, C., Berg, M., Weickert, J., Backes, M.: Diffusion-based image compression in steganography. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Fowlkes, C., Wang, S., Choi, M.-H., Mantler, S., Schulze, J., Acevedo, D., Mueller, K., Papka, M. (eds.) ISVC 2012, Part II. LNCS, vol. 7432, pp. 219–228. Springer, Heidelberg (2012)
8. Osher, S., Paragios, N.: Geometric Level Set Methods in Imaging,Vision,and Graphics. Springer-Verlag New York, Inc., Secaucus (2003)
9. Paragios, N.: A variational approach for the segmentation of the left ventricle in cardiac image analysis. International Journal of Computer Vision 50(3), 345–362 (2002)
10. Pien, H., Desai, M., Shah, J.: Segmentation of MR images using curve evolution and prior information. Int. J. Pattern Recogn. 11(8), 1233–1245 (1997)
11. Tari, S., Genctav, M.: From a modified Ambrosio-Tortorelli to a randomized part hierarchy tree. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 267–278. Springer, Heidelberg (2012)
12. Tari, Z., Shah, J., Pien, H.: A computationally efficient shape analysis via level sets. In: Proceedings of the Workshop on Mathematical Methos in Biomedical Image Analysis, pp. 234–243 (1996)

Defect Classification on Specular Surfaces Using Wavelets

Andreas Hahn[1], Mathias Ziebarth[2], Michael Heizmann[3], and Andreas Rieder[1]

[1] Karlsruhe Institute of Technology, Institute for Applied and Numerical Mathematics, Kaiserstr. 89-93, 76133 Karlsruhe, Germany
[2] Karlsruhe Institute of Technology, Institute for Anthropomatics, Adenauerring 4, 76131 Karlsruhe, Germany
[3] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Fraunhoferstr. 1, 76131 Karlsruhe, Germany
publications@andyhahn.de, {mathias.ziebarth,andreas.rieder}@kit.edu, michael.heizmann@iosb.fraunhofer.de

Abstract. In many practical problems wavelet theory offers methods to handle data in different scales. It is highly adaptable to represent data in a compact and sparse way without loss of information. We present an approach to find and classify defects on specular surfaces using pointwise extracted features in scale space. Our results confirm the presumption that the stationary wavelet transform is better suited to localize surface defects than the classical decimated transform. The classification is based on a support vector machine (SVM) and furthermore applicable to empirically evaluate given wavelets for specific classification tasks and can therefore be used as quality measure.

Keywords: Defect detection, Wavelet Transform, SVM, Classification, Deflectometry, Surface inspection.

1 Introduction

We inspect different transforms and several wavelet bases for the detection of defects on surfaces. The detection and moreover classification of defects is realized directly in scale space. For evaluation we choose height maps as measurements of (partially) specular surfaces as application. In contrast to matt surfaces, specular surfaces are not observable directly in the visible spectrum. Therefore the measurements are made exploiting the specular reflectivity by using deflectometry. The paper is structured as follows:

In section 2 we show previous work with similar approaches for inspection. Section 3 gives an introduction to deflectometry and the measurements of the surface used for evaluation. The underlying idea of this work is illustrated in section 4. Then a short introduction to wavelet theory, important wavelet transforms and important properties of wavelets are given in section 5. In section 6 the requirements that have to be met and the composition of the feature vector are explained. Finally in section 7 it is described how a support vector machine

A. Kuijper et al. (Eds.): SSVM 2013, LNCS 7893, pp. 501–512, 2013.

(SVM) is used to classify each point on the surface into error-free area or a specific defect type. Experimental results using deflectometric measurements from specular surfaces are shown in section 8. Section 9 summarizes the proposed method.

2 Related Work

Automated defect detection has been a wide area of research for at least twenty years. In recent works more and more publications use the wavelet decomposition for defect detection. It is a well studied tool that is widely used to analyze signals in scale space. One of its applications is the evaluation of image data in multiple scaling levels. As a surface height map can be represented as an image, methods from image classification can be adapted to our task. We list some publications which also use wavelets for feature extraction in combination with a support vector machine as classifier.

Jiang and Blunt [1] and later on Jiang et al. [2] investigate different ways to realize a wavelet transform on surface data. Their main interest is the use of the stationary wavelet transform together with complex wavelets to get good surface representations. Rosenboom et al. [3] study different wavelets to represent reconstructions of specular surfaces, obtained by deflectometry, in a sparse manner. Rajpoot and Rajpoot [4] use a discrete wavelet decomposition and reconstruction of singular details to extract local features in the spatial domain. These features are classified with a support vector machine to detect fabric defects. Ghorai et al. [5] analyze a wider variety of orthogonal wavelets to detect different defects on images of hot-rolled steel products. Their approach is to partition the surface into square areas that are classified by a support vector machine using extracted features. Zhang et al. [6] use the discrete wavelet transform to smooth images of strongly reflecting metal and to emphasize edges. Then an edge filter is applied and features based on a Fourier transform are extracted to detect metal defects.

Our method differs from the established methods to detect and classify defects on surfaces with wavelets in combination with SVMs by utilizing scale space information directly for classification. This makes our approach less complex and computationally more efficient. There are similarities to an adaptive Bayesian wavelet shrinkage approach proposed by Chipman et. al [7]. While their task is completely different as they want to suppress noise, their methods to discriminate between noise and signal are similar to our discrimination between the surface and defects. But our classification differs from theirs because we use non-parametric statistics to learn decision boundaries.

3 Deflectometry

Specular surfaces are present almost everywhere in daily life. For example surfaces on dishes, mirrors, furniture, home appliances or laquered car bodies are strongly or at least partially specular. More than on diffuse surfaces the visual impression of these surfaces is an important aesthetic property. Because of the

specular surface, not only the object itself can be seen, but moreover the reflected surrounding. In this reflection both very small flaws and large bumps can be seen and reduce the value of the object. Consequently quality inspection on specular surfaces is essential. Unfortunately classical approaches to measure zero-order surface properties are difficult to apply. Deflectometry offers methods to measure first-order surface properties and to infer the shape. Due to the measurement principle of deflectometry, it is especially sensitive to surface curvature which corresponds to the human perception of specular surfaces and is therefore appropriate to detect aesthetical defects. For the measurement, a system consisting of a camera with image plane I, a specular surface S and a screen L arranged in a triangular setup, as seen in Fig. 1, is used. The camera observes

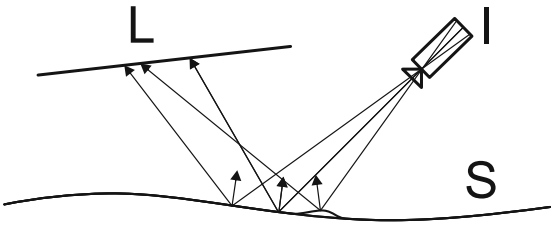


Fig. 1. Deflectometric system setup consisting of camera I, surface S and screen L

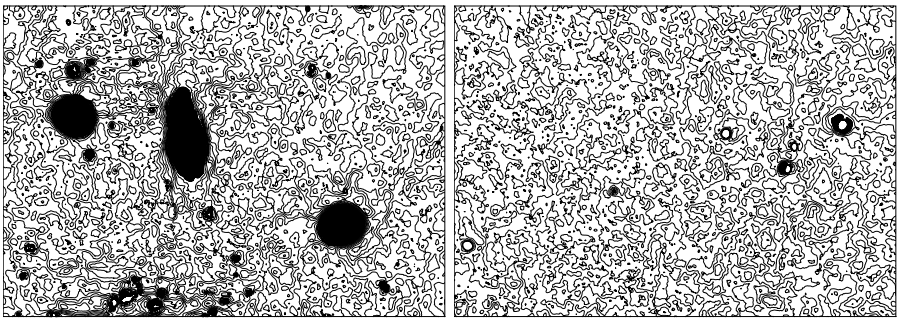
a sequence of phase shifted sinus patterns projected onto the screen over the specular surface. By decoding this information, viewing rays from the camera plane to the surface P_I can be uniquely assigned to points on the screen P_L:

$$l : P_I \mapsto P_L,\ l[u, v] = (x_L, y_L)\ . \tag{1}$$

This mapping is called deflectometric registration. It contains first order information about the surface, because the direction of each reflected ray is determined by the normal of the reflecting surface. But without knowledge of the distance between the camera and the surface, it is impossible to unambiguously reconstruct the surface from the deflectometric registration. An overview of the topic is given by Werling et al. in [8, 9]. Theoretical and practical accuracy properties are studied by Knauer [10]. In our case the surface is reconstructed using an iterative algorithm that finds a surface whose gradients match the measured gradients under consideration of some a priori known points on the surface. These a priori known points are determined by a separate measurement. Two contour plots of reconstructed surfaces are given in Fig. 2. The lines mark points of equal height, black areas are below a threshold and white areas above a threshold.

4 Defects in Frequency Domain

As shown in section 2 most approaches for defect detection use features in spatial space. Because defects can have a great variety in shape and size, one either

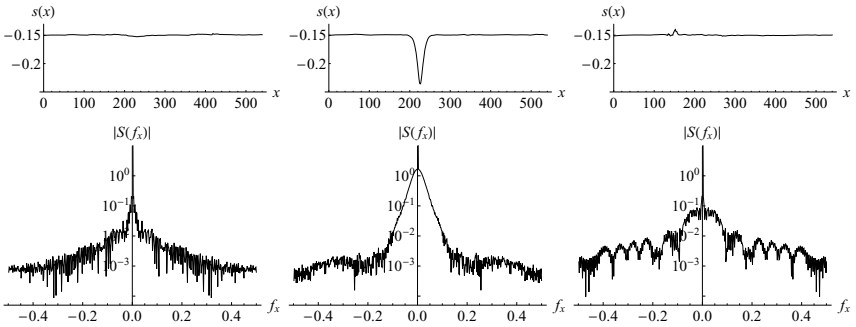


(a) Defects: 3 dents, pimples (b) Defects: pimples

Fig. 2. Contours of two reconstructed surfaces with orange peel and defects

has to find invariant features or very complex classifiers. We assume that a representation of defects in the frequency domain provides the required invariance.

This conjecture is illustrated in the following example using deflectometric measurements of lacquered metal sheets with large defect-free areas and small local defects of two different types. We used a Fourier transform to represent the windowed defect in the frequency domain. Figure 3 shows the frequency spectrum of sections through two of the defects and a section through a defect-free area. As it can be seen, the frequencies of the non-defect area mainly concentrate around zero. The large dent contains a wide band of low frequencies. In contrast to this the smaller pimple has a lot more impact on higher frequencies. Besides the observation that the occouring frequencies are depending on the size of the defect and the steepness of the flanks, it is important that the shape of the spectrum differs according to the shape of the defect. Note that this illustration of the underlying idea works, because we only transformed a selected area around the defect. When the location of the defect is unknown and one has to find its location, the Fourier transform as a global transform is not suitable.



(a) defect-free (b) dent (b) pimple

Fig. 3. Three sections through different defects and their frequency spectrum

5 Wavelet Theory

5.1 Continuous Wavelet Transform

Similar to the Fourier transform, the wavelet transform describes frequency properties. In contrast to the fourier transform, which is a global transformation, the wavelet transform gives localized frequency information, by using basis functions which only differ significantly from zero in a small interval.

The continuous wavelet transform for a function $f \in L^2(\mathbb{R})$ is defined as a convolution with the window function ψ

$$W_\psi f(s,u) = \frac{1}{\sqrt{s}} \int_{\mathbb{R}} f(t)\psi\left(\frac{t-u}{s}\right) dt = \langle f, \psi_{u,t} \rangle \ . \tag{2}$$

Its advantage is the choosable window function that is variable and limited by the Heisenberg principle. The higher the extracted frequencies are, the smaller is the window in the spatial domain. To allow the perfect reconstruction of f, ψ has to satisfy the admissibility condition [11].

5.2 Discrete Wavelet Transform

In practice the wavelet transform is usually calculated using the discrete wavelet transform (DWT). Instead of all scales only discrete, i.e. dyadic scales are considered and instead of all translations only non-overlapping translations of the wavelet are considered. For this purpose a scaling function ϕ is introduced, which represents f in approximations a:

$$a_s[u] = \int_{\mathbb{R}} f(x) \frac{1}{\sqrt{2^s}} \phi\left(\frac{x - 2^s u}{2^s}\right) dx, \ (s,u) \in \mathbb{Z}^2 \ . \tag{3}$$

The scaling function has a low-pass property. Together with a suitable wavelet function, which works as a high-pass filter, we are able to construct a multiresolution analysis. The calculation is realized using filter banks. Starting with a sampled signal $a_0[x]$ in scale $s = 0$ the approximation in the next scale is calculated using the filter h of the scaling function ϕ:

$$a_{s+1}[x] = \sum_{n=-\infty}^{\infty} h[n - 2x] a_s[n] \ . \tag{4}$$

The details lost in the approximation are extracted using the orthogonal high-pass filter g of the wavelet function ψ:

$$d_{s+1}[x] = \sum_{n=-\infty}^{\infty} g[n - 2x] a_s[n] \ . \tag{5}$$

As we want to extract features from a two-dimensional surface we use a tensor wavelet approach with a one-dimensional filtering in both directions. This results in one approximation space $a_s[x,y]$ and three detail spaces $d_{s,1}[x,y]$, $d_{s,2}[x,y]$, $d_{s,3}[x,y]$, where $x, y \in \mathbb{Z}$ determines the position on the surface, in each scale.

5.3 Stationary Wavelet Transform

For our purposes a good localization of the detail coefficients in all scales is desired. The stationary wavelet transform (SWT, desribed in [12] as algorithme à trous) achieves this by omitting the subsampling and by widening the filters on each scale. We start with the filters h_0 and g_0 on scale $s = 0$ and use dilated filters in each scale:

$$h_{s+1}[n] = \begin{cases} h_s[n/2], & n \text{ even} \\ 0, & n \text{ odd} \end{cases}, \quad g_{s+1}[n] = \begin{cases} g_s[n/2], & n \text{ even} \\ 0, & n \text{ odd} \end{cases} . \tag{6}$$

This results in a higher redundancy and consequently higher computational expenses of the transform, but is invariant to shifts of the signal. Furthermore it gives us the good localization of each coefficient in all scales. Hence it leads to better classification results as we will show in section 8.

5.4 Important Properties of Wavelets

The properties of the wavelet transform are determined by the choice of a wavelet. We discuss some important properties of wavelets and their relevance for our application.

Support. The support of a wavelet is the smallest interval containing all of its non-zero values. Its size determines the size of the interval that is influenced by a single position of the signal. As we want to distinguish between neighboring defects the size of this interval has to be as small as possible. Furthermore a small support leads to few calculations and a precise localization of defects.

Symmetry. The localization, especially in higher scales, is affected by the symmetry of the scaling function. Orthogonal wavelets cannot be symmetric and compactly supported at the same time. But it is possible to construct almost symmetric orthogonal wavelets, called symlets [13]. Biorthogonal wavelets can be constructed symmetrically.

Vanishing Moments. A wavelet ψ with p vanishing moments is orthogonal to polynomials of degree less or equal to $p - 1$:

$$\int_{\mathbb{R}} t^k \psi(t) dt = 0 \quad \text{for } 0 \leq k < p \ . \tag{7}$$

This means functions that can be described by polynomials of degree $p - 1$ appear in the approximation space only. If we want to extract high frequency parts like defects we want to ignore surface properties that can be assumed as low frequency parts. As the surface is usually modeled with splines it is piecewise polynomial with low order [14]. Thus consequently we should use wavelets with many vanishing moments. Ingrid Daubechies proved that the number of vanishing moments is proportional to the support of a wavelet. The wavelets with a

maximum of vanishing moments at minimal support are the so-called Daubechies wavelets. Usually wavelets are named by the number of their vanishing moments, e.g. the symlet 3 is a wavelet with 3 vanishing moments.

Choice of the Wavelet. On the one hand a good localization is essential for detecting even small defects. On the other hand we need enough vanishing moments to make sure the smooth parts of the surface don't appear in the detail space. Therefore we need to find a trade-off between the number of vanishing moments and the length of the support. In section 8, we study different orthogonal and biorthogonal wavelets for defect detection and classification.

6 Features

The detection and classification of defects depends on good features. With good features, we obtain sparse representations that only show problem specific properties. This sparsity is achieved with invariances under translation, curvature of the surface and scaling. Scaling and translation invariance are obtained by the multiresolution analysis of the stationary wavelet transform. Note that the used scaling function has to be symmetric, otherwise the location of a defect would shift over the scales and introduce a bias in the estimation of the location. Invariance against surface curvature depends on the number of vanishing moments of the wavelet. We use biorthogonal spline wavelets and symlets with up to five vanishing moments and invariance under curvature up to fourth order polynomials. The tensor wavelet approach we use considers only three directions and can therefore not be used to extract anisotropic features. This limitation doesn't matter in our case, because the defects we consider are approximately round.

Each pixel of the surface is associated with a value of each scale and each direction of the wavelet transform as seen in Fig. 4. On s scales together with r directions of the 2D wavelet transform we get $s \cdot r$ features for each pixel (x, y) on the surface. While for the SWT one coefficient in each scale for each point is calculated, the coefficients of the DWT describe, depending on the scale, more

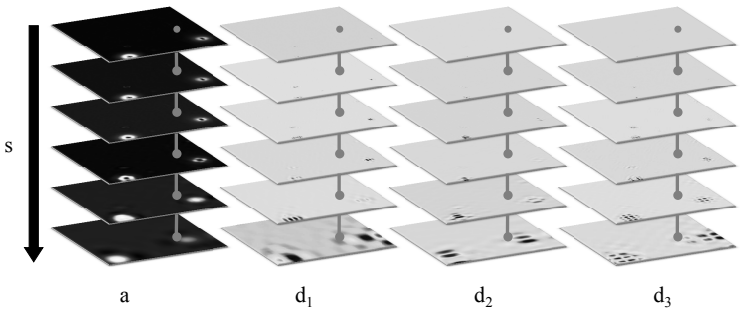


Fig. 4. Feature vector depicted as cut through all scales of the SWT

than one point on the surface. We interpolate the DWT coefficients for each point using the nearest neighbored coefficient.

The 2D tensor wavelet gives us $r = 3$ directions. We analyze $s = 5$ scales which leads us to a feature vector of 15 coefficients from (5) for each pixel:

$$\boldsymbol{d} = (d_{1,1}, d_{1,2}, d_{1,3}, d_{2,1}, \ldots, d_{5,1}, d_{5,2}, d_{5,3}) \quad . \tag{8}$$

7 Classification

The classification is performed by a support vector machine (SVM) as described by Vapnik [15]. For the discrimination of more than two classes, the SVM has to be extended. We use the free library LIBSVM by Chang and Lin [16] for our classification. They provide an implementation of Vapniks SVM with an extension to combine several two-class SVMs to one multiclass SVM. Moreover they implement an extension that allows the SVM to give probability estimates for each class.

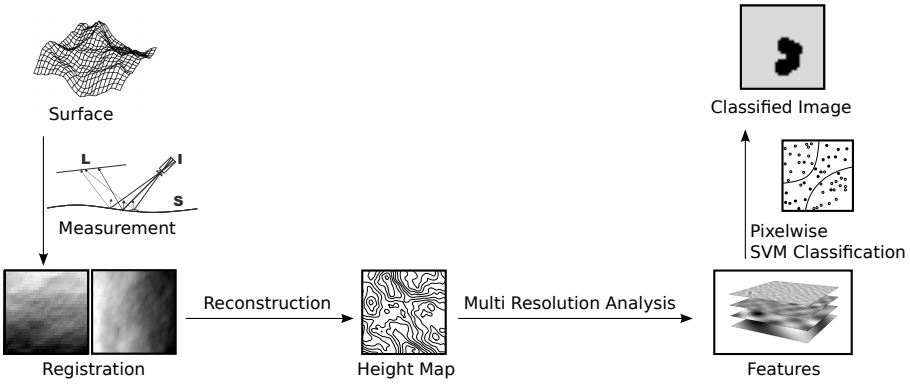


Fig. 5. Classification process

We use a radial basis function $e^{-\gamma|u-v|^2}$ as kernel function. Besides the necessity for training data the SVM needs to be parametrized with two parameters: a regularization parameter for weighting the costs of misclassifications C and the width of the kernel function γ. The optimization of these parameters $\gamma \in \{2^{i/2}\}, C \in \{2^{j/2}\}, i \in \{-4, \ldots, 40\}, j \in \{-24, \ldots, 24\}$ is realized using a five fold cross validation with 1500 feature vectors for each class and a grid search as proposed in [17].

7.1 Training of the SVM

As described in section 6 one feature vector is extracted for each point on the surface separately. We annotated some measurements to provide training data

for each class. Since the classification accuracy improves when the feature values are limited, we normalize the feature values to mean 0 and variance 1. The parameters for this normalization are determined by the training data and are applied to the testing data later on as well.

7.2 Classification

Using the probability estimates of the support vector machine $p(\boldsymbol{d} \mid c)$ for a feature vector $\boldsymbol{d}$ with class c, instead of a voting decision a maximum likelihood decision can be made:

$$\arg\max_{c} p(\boldsymbol{d} \mid c) \quad . \tag{9}$$

If additional prior information $p(c)$ is available and necessary a maximum a posteriori decision can be made. Prior information may be given by an expert, for example if only certain classes are possible. Another reason to include prior information is to connect adjacent points on the surface, for example it is unlikely that in the middle of a small defect some points belong to a larger defect.

7.3 Evaluation of Wavelets

The comparison of ground truth and the classification results obtained based on features calculated with a certain wavelet allows an empirical evaluation of that wavelet. To measure the evaluation results we use the accuracy of the results (i.e. the number of true classifications divided by the number of all classifications).

8 Results

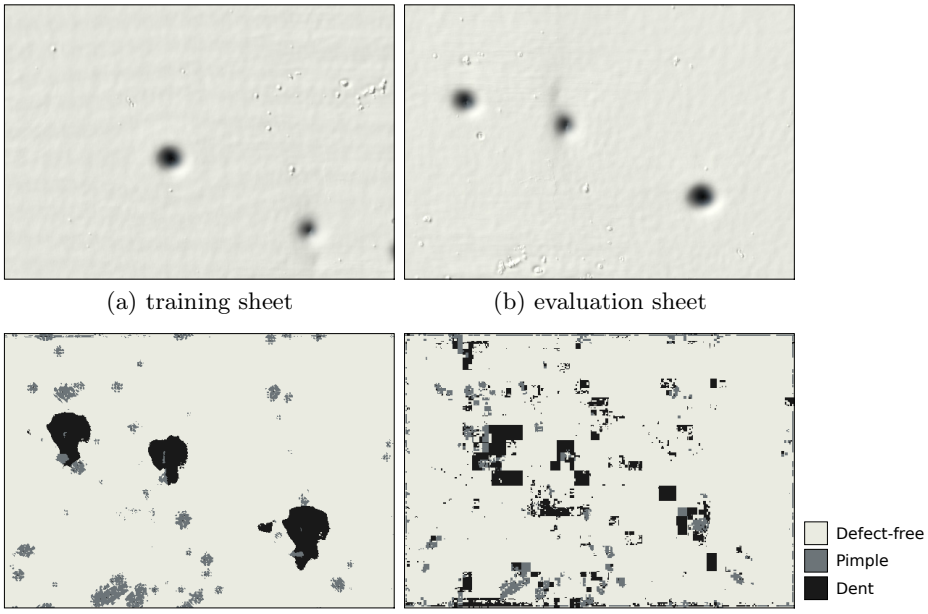
To evaluate our method we used manually classified height maps of two lacquered metal sheets on which dents and pimples occurred. First the features using one specific wavelet were determined. One sheet was solely used as training data for estimating the normalization parameters and to learn the classifier. For each class we randomly selected 1000 points on the surface to train the SVM.

The other sheet was used to test the trained classifier. Here the features of 1500 points on the surface for each class were randomly selected to evaluate the quality of the classification.

Figure 6 shows a classification of a surface classified with a biorthogonal 3.3 wavelet over five scales using an SWT and a DWT. The differences between both transforms are obvious. A first noticeable difference is the blocky classification due to the discretization of the DWT. And secondly the classification is overall worse as many misclassification's can be observed.

The results of the SWT are much better. There are still some misclassification's at the borders of the surface that result from boundary effects. Furthermore one can recognize that defects tend to be classified larger than they are in reality.

Table 1 shows the accuracy of our method with various wavelets and methods as well as the percentage of samples used as support vectors. As we used nearly



(a) training sheet (b) evaluation sheet

(c) stationary wavelet transform (d) discrete wavelet transform

Fig. 6. Relief plots of the training data (a) and the evaluation data (b). (c), (d) Classification results for (b) using the biorthogonal 3.3 wavelet.

plane surfaces the results are good with few vanishing moments. Furthermore the SVM complexity decreases by using good wavelets like Symlet 2 and the SWT.

A glance at Table 1 confirms the expectations after viewing Fig. 6. A classification using the discrete wavelet transform is very inaccurate. The comparatively good results using wavelets with few vanishing moments can be reasoned with the good localization properties. More vanishing moments and a wider support lead to worse localization and separation of neighbored defects on flat surfaces. Detailed results are shown in Tab. 2 and reveal weaknesses detecting pimples.

Table 1. Accuracy values on 4500 evaulation samples and percentage of the 3000 training samples used as support vectors

	accuracy		support vectors	
Wavelet	SWT	DWT	SWT	DWT
Biorthogonal 3.5	77.73%	55.31%	28.17%	56.40%
Biorthogonal 2.4	92.73%	70.84%	11.37%	42.40%
Biorthogonal 3.3	89.22%	57.11%	17.27%	42.47%
Symlet 2	93.60%	84.22%	11.00%	16.97%
Symlet 3	90.00%	75.20%	12.50%	24.27%
Symlet 4	88.07%	65.78%	17.07%	41.83%

Table 2. Confusion matrices. Actual classes defect-free d, pimple p and bump b. Predicted classes $\hat{d}$, $\hat{p}$ and $\hat{b}$.

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1445	29	26
p	766	655	79
b	14	88	1398

SWT Bior3.5

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1461	18	21
p	181	1243	76
b	11	20	1469

SWT Bior2.4

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1448	26	26
p	371	1109	20
b	2	40	1458

SWT Bior3.3

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1459	16	25
p	117	1351	32
b	17	81	1402

SWT Sym2

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1441	21	38
p	199	1253	48
b	90	54	1356

SWT Sym3

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1450	15	35
p	283	1135	82
b	70	52	1378

SWT Sym4

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1353	51	96
p	680	708	112
b	711	361	428

DWT Bior3.5

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1406	38	56
p	558	868	74
b	256	330	914

DWT Bior2.4

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1366	40	94
p	527	842	131
b	818	320	362

DWT Bior3.3

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1460	27	13
p	171	1303	26
b	77	396	1027

DWT Sym2

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1440	26	34
p	337	1158	5
b	434	280	786

DWT Sym3

	$\hat{d}$	$\hat{p}$	$\hat{b}$
d	1438	48	14
p	474	979	47
b	639	318	543

DWT Sym4

It can be seen that much better results can be achieved with the stationary wavelet transform. The classification quality remains on a high level even with wavelets of larger size.

9 Summary

We presented an approach for defect detection and classification directly in scale space. While being computationally efficient due to the use of the wavelet transform, the simple feature extraction and the application of the SVM, it offer numerous advantages over existing approaches. The approach was evaluated on deflectometric measurements of lacquered surfaces. Our results show that using only the small feature vectors it is possible to detect and classify two defect classes ranging over multiple scales with high accuracy. Additionally we observed that only few samples were used by the SVM, which leads to the conclusion that our features are highly representative for the given task. Furthermore our approach is suitable as a measure to evaluate the ability of a wavelet to discriminate given defect classes.

This work was financed by Baden-Württemberg Stiftung within the project MID-Wave.

References

1. Jiang, X., Blunt, L.: Third generation wavelet for the extraction of morphological features from micro and nano scalar surfaces. Wear 257, 1235–1240 (2004)
2. Jiang, X., Scott, P., Whitehouse, D.: Wavelets and their applications for surface metrology. CIRP Annals - Manufacturing Technology 57, 555–558 (2008)

3. Rosenboom, L., Kreis, T., Jüptner, W.: Surface description and defect detection by wavelet analysis. Measurement Science and Technology 22, 45102 (2011)
4. Rajpoot, K., Rajpoot, N.: Wavelets and support vector machines for texture classification. In: Proceedings of 8th International Multitopic Conference, INMIC 2004 (2004)
5. Ghorai, S., Mukherjee, A., Gangadaran, M., Dutta, P.K.: Automatic defect detection on hot-rolled flat steel products. IEEE Transactions on Instrumentation and Measurement (2012) (preprint)
6. Xue-wu, Z., Yan-qiong, D., Yan-yun, L., Ai-ye, S., Rui-yu, L.: A vision inspection system for the surface defects of strongly reflected metal based on multi-class svm. Expert Systems with Applications 38, 5930–5939 (2011)
7. Chipman, H.A., Kolaczyk, E.D., McCulloch, R.E.: Adaptive bayesian wavelet shrinkage. Journal of the American Statistical Association 92 (1997)
8. Werling, S., Mai, M., Heizmann, M., Beyerer, J.: Inspection of specular and partially specular surfaces. Metrology and Measurement Systems 16, 415–431 (2009)
9. Balzer, J., Werling, S.: Principles of Shape from Specular Reflection. Measurement 43, 1305–1317 (2010)
10. Knauer, M.C., Kaminski, J., Häusler, G.: Phase measuring deflectometry: a new approach to measure specular free-form surfaces. Optical Metrology in Production Engineering 5457, 366–376 (2004)
11. Louis, A.K., Maaß, P., Rieder, A.: Wavelets - theory and applications. Wiley (1997)
12. Mallat, S.G.: A Wavelet Tour of Signal Processing: The Sparse Way. Academic Press Elsevier (2009)
13. Daubechies, I.: Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics (1992)
14. Prautzsch, H., Böhm, W., Paluszny, M.: Bézier and B-spline techniques. Mathematics and Visualization. Springer (2002)
15. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
16. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
17. Hsu, C.W., Chang, C.C., Lin, C.J.: A Practical Guide to Support Vector Classification (2010)

Author Index