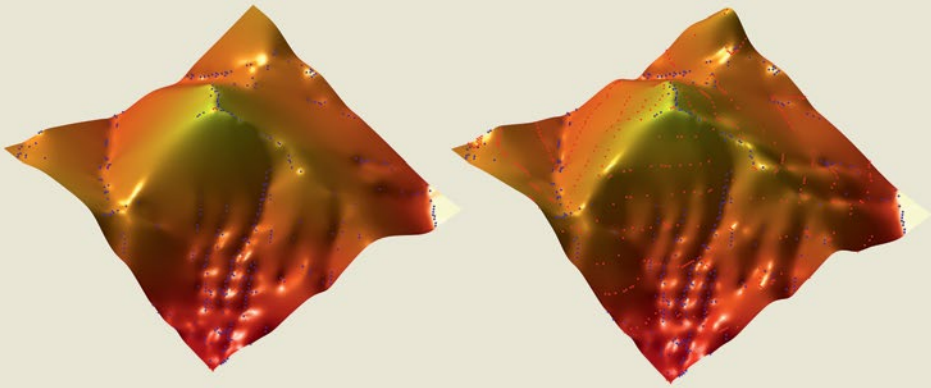
Mathematics and Visualization



Xue-Cheng Tai · Egil Bae
Marius Lysaker *Editors*

Imaging, Vision and Learning Based on Optimization and PDEs

IVLOPDE, Bergen, Norway,
August 29 – September 2, 2016

Springer

Mathematics and Visualization

More information about this series at http://www.springer.com/series/4562

Xue-Cheng Tai • Egil Bae • Marius Lysaker
Editors

Imaging, Vision and Learning Based on Optimization and PDEs

IVLOPDE, Bergen, Norway, August 29 – September 2, 2016

Springer

Editors
Xue-Cheng Tai
Department of Mathematics
University of Bergen
Bergen, Norway

Egil Bae
Norwegian Defence Research
Establishment
Kjeller, Norway

Marius Lysaker
University of South-Eastern Norway
Porsgrunn, Norway

ISSN 1612-3786 ISSN 2197-666X (electronic)
Mathematics and Visualization
ISBN 978-3-319-91273-8 ISBN 978-3-319-91274-5 (eBook)
https://doi.org/10.1007/978-3-319-91274-5

Library of Congress Control Number: 2018956324

Mathematics Subject Classification (2010): 35-XX, 49-XX, 65-XX, 90-XX

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

Preface

It has become an established paradigm to formulate problems within image processing and computer vision as partial differential equations (PDEs), variational problems or high-dimensional optimization problems. This compact, yet expressive framework makes it possible to incorporate a variety of desired properties of the solutions and to design algorithms based on well-founded mathematical theory. Applications range from early image formation through compressive sensing, to low-level image enhancement, restoration and feature detection, and to higher-level image understanding, segmentation and classification. More recently, the same tools have also shown great promise in more general applications of data analysis and machine learning.

In August 2016, we organized a conference titled Imaging, Vision and Learning based on Optimization and PDEs (IVLOPDE) in the city of Bergen, Norway. The conference was intended to foster collaboration and exchange of new ideas within these mathematical techniques for the broad range of applications in imaging science, computer vision and machine learning. The 5-day event included invited presentations from 18 internationally leading experts within the field, and 16 contributed poster presentations. Plenty of time was also set aside for informal discussions, such as a full-day excursion to the fjords and mountains of western Norway. After the conference, the participants were invited to submit full papers based on their presentations, or based on new ideas that may have emerged during the course of the event. Each submitted article was evaluated by 2–3 expert reviewers, who were mostly selected among the other invited speakers. The reviewers provided a lot of useful feedback, which in some cases led to new insights that the authors incorporated in revised versions of their articles.

This book constitutes the conference post-proceedings of IVLOPDE. It contains 11 original research articles that were selected from the submitted full papers after the peer-review process. The articles present various novel techniques and analytical results within optimization, variational models and PDEs, together with experimental results on applications ranging from early image formation to high-level image and data analysis. To guide the reader, the articles have been divided into four topical sections: (I) image reconstruction from incomplete data, (II)

image enhancement, restoration and registration, (III) 3D image understanding and classification, and (IV) machine learning and big data analysis. Each section features a balance of theoretically oriented articles and application oriented articles. The book will benefit all researchers within mathematics, imaging science or data science who would like to become more familiar with this active field, as well as experts who would like to learn about the latest developments.

We would like to thank all the reviewers for their valuable suggestions and careful evaluations, the Research Council of Norway (through ISP-matematikk project 239033/F20) for providing funding for the conference, and Ruth Allewelt and Martin Peters at Springer for their excellent support and patience while we were preparing this book.

Bergen, Norway — Xue-Cheng Tai
Kjeller, Norway — Egil Bae
Porsgrunn, Norway — Marius Lysaker
April 2018

Contents

Part I
Image Reconstruction from Incomplete Data

Chapter 1
Adaptive Regularization for Image Reconstruction from Subsampled Data

Michael Hintermüller, Andreas Langer, Carlos N. Rautenberg, and Tao Wu

Abstract Choices of regularization parameters are central to variational methods for image restoration. In this paper, a spatially adaptive (or distributed) regularization scheme is developed based on localized residuals, which properly balances the regularization weight between regions containing image details and homogeneous regions. Surrogate iterative methods are employed to handle given subsampled data in transformed domains, such as Fourier or wavelet data. In this respect, this work extends the spatially variant regularization technique previously established in Dong et al. (J Math Imaging Vis 40:82–104, 2011), which depends on the fact that the given data are degraded images only. Numerical experiments for the reconstruction from partial Fourier data and for wavelet inpainting prove the efficiency of the newly proposed approach.

Introduction

Image restoration is one of the fundamental tasks in image processing. The quality of the obtained reconstructions depends on several input factors: the quality of the given data, the choice of the regularization term or prior, and the proper balance

M. Hintermüller (✉)
Weierstrass Institute, Berlin, Germany
e-mail: michael.hintermueller@wias-berlin.de

A. Langer
Universität Stuttgart, Stuttgart, Germany
e-mail: andreas.langer@mathematik.uni-stuttgart.de

C. N. Rautenberg
Humboldt-Universität zu Berlin, Berlin, Germany
e-mail: carlos.rautenberg@math.hu-berlin.de

T. Wu
Technische Universität München, Garching, Germany
e-mail: tao.wu@tum.de

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_1

of data fidelity versus filtering, among others. The goal of the present paper is to reconstruct an image, defined over the two-dimensional Lipschitz (image) domain Ω, from contaminated data f, defined over the data domain Λ. Given the original image $\widehat{u} : \Omega \to \mathbb{R}$, the data formation model is assumed to be

$$f = K\widehat{u} + \eta, \tag{1.1}$$

where $K\widehat{u}$ represents possibly subsampled data which results from a linear sampling strategy and η is related to white Gaussian noise (with zero mean). As we later describe, η is given by white Gaussian noise in the numerical tests, and in the function space setting we assume it to be a zero mean L^2 function. A more precise description of the data formation model is postponed until section "Problem Settings and Notations".

A popular approach to image restoration rests on variational methods, i.e., the characterization of the reconstructed image u as the solution of a minimization problem of the type

$$\min_u \quad \Phi(u; f) + \alpha R(u), \tag{1.2}$$

where $\Phi(\cdot; f)$ represents a data fidelity term, $R(\cdot)$ an appropriate filter or prior, and $\alpha > 0$ a regularization parameter which balances data fidelity and filtering. The choice of Φ is typically dictated by the type of noise contamination. As long as Gaussian noise is concerned, following the maximum likelihood we choose

$$\Phi(u; f) = \frac{1}{2}\|Ku - f\|^2_{L^2(\Lambda)}.$$

On the other hand, R encodes prior information on the underlying image. For the sake of edge preservation, we choose

$$R(u) = |Du|(\Omega), \tag{1.3}$$

i.e., the total variation of a function u (see Eq. (1.5) below for its definition). Then the resulting model (1.2) becomes the well-known Rudin-Osher-Fatemi (ROF) model [31] which has been studied intensively in the literature; see, e.g., [5, 6, 8, 14, 21, 24, 29, 32, 33] as well as the monograph [38] and many references therein.

It is well known that the proper choice of α is delicate. A general guideline is the following one: Large α favorably removes noise in homogeneous image regions, but it also compromises image details in other regions; Small α, on the other hand, might be advantageous in regions with image details, but it adversely retains noise in homogeneous image regions. For an automated choice of α in (1.2) several methods have been devised; see for example [10, 18, 20, 34, 40] and the references therein, and see [22, 25] for the spatially distributed α methods. We note that instead of considering (1.2) one may equivalently study $\lambda\Phi(u; f) + R(u)$ with

$\lambda = 1/\alpha$. Based on this view, in [2], a piecewise constant function λ over the image domain is considered: The partitioning of the imagine domain is done via pre-segmentation and λ is computed by an augmented-Lagrangian-type algorithm. While still operating in a deterministic regime, [2] interestingly uses a spatially variant (more precisely a piecewise constant) parameter function λ.

Later it was noticed that stable choices of λ (or respectively α) have to incorporate statistical properties of the noise. In this vein, in [1, 15] automated update rules for λ based on statistics of local constraints were proposed. For statistical multiscale methods we refer to [16, 17, 26]. A different approach has been proposed in [35] for image denoising only, where non-local means [4] has been used to create a non-local data fidelity term. While the methods in [1, 15, 23] are highly competitive in practice, the adjustment of λ requests the output of K to be a deteriorated image which is again defined over Ω. This, however, limits the applicability of these approaches in situations where K involves transformation of an image into a different type of data output space. Particular examples of such transformations include wavelet or Fourier transforms. It is therefore the goal of this paper to study the approach of [15] in the context of reconstructing from such non-image data, possibly coupled with subsampling for the sake of fast data acquisition.

Here we also mention other spatially weighted total variation methods from the existing literatures. Very often these methods, different from [15, 23] (and also the present paper), weight the total variation locally by certain edge indicators. In [9, 42, 43] the difference of the image curvature was used as an edge indicator, while alternatively the (modified) difference of eigenvalues of the image Hessian was considered by Yan et al. [41] and Ruan et al. [30]. Recently, the authors in [27, 28] used similar edge indicators to weight the total variation anisotropically under the framework of quasi-variational inequalities.

The rest of the paper is organized as follows. Section "Problem Settings and Notations" describes in detail the problem settings and the notations. Our adaptive regularization approach is presented in section "Adaptive Regularization Approach". Section "Numerical Experiments" concludes the paper with numerical experiments on reconstruction of partial Fourier data and wavelet inpainting.

Problem Settings and Notations

In the data formation model (1.1), we shall consider the continuous linear operator K as a composition of two linear operators, i.e., $K = S \circ T$. More precisely, $T : L^2(\Omega) \to L^2(\Lambda)$ is a linear orthogonal transformation which preserves the inner product, i.e., $\langle u, v\rangle_{L^2(\Omega)} = \langle Tu, Tv\rangle_{L^2(\Lambda)}$ for any $u, v \in L^2(\Omega)$. Typical examples of T include Fourier and orthogonal wavelet transforms. Further, we denote the subsampling domain by $\tilde{\Lambda}$, which is assumed to be a (measurable) subset of Λ of finite positive measure, i.e., $0 < |\tilde{\Lambda}| < \infty$. Such a $\tilde{\Lambda}$ may arise in application cases where there is no access to the complete measured data over Λ, but only to a

reduced version of it. Define $\mathbf{1}_{\tilde{\Lambda}}$ as the characteristic function on $\tilde{\Lambda}$, i.e., $\mathbf{1}_{\tilde{\Lambda}}$ equals 1 on $\tilde{\Lambda}$ and 0 elsewhere. Then the so-called subsampling operator $S : L^2(\Lambda) \to L^2(\Lambda)$ is defined by $(Sf)(y) = \mathbf{1}_{\tilde{\Lambda}}(y) f(y)$ almost everywhere (a.e.) on Λ. It is worth mentioning that S is an orthogonal projection which satisfies idempotency, i.e., $S^2 = S$, and self-adjointness, i.e., $S^* = S$, and that the range of S, denoted by Ran S, is a closed subspace of $L^2(\Lambda)$. In this setting, we consider the noise η as an arbitrary oscillatory function in Ran S with

$$\int_{\tilde{\Lambda}} \eta \, dy = 0, \quad \text{and} \int_{\tilde{\Lambda}} |\eta|^2 dy = \sigma^2 |\tilde{\Lambda}|, \tag{1.4}$$

for some $\sigma > 0$. As a direct consequence, the data f according to (1.1) also lies in Ran S.

For $u \in L^1(\Omega)$, the total variation term $|Du|(\Omega)$ in (1.3) is defined as follows:

$$|Du|(\Omega) := \sup \left\{ \int_{\Omega} u \ \text{div} \ p \, dx : \ p \in C_0^1(\Omega; \mathbb{R}^2), \ \|p\|_{L^\infty(\Omega)} \leq 1 \right\}. \tag{1.5}$$

Here, $C_0^1(\Omega; \mathbb{R}^2)$ denotes the set of all $\mathbb{R}^2$-valued continuously differentiable functions on Ω with compact support.

Adaptive Regularization Approach

The focus of this paper is to reconstruct a high-quality image from subsampled data in a non-image data domain using an adaptive regularization approach. The present section is structured as follows. In section "ROF-Model and Surrogate Iteration", we introduce the surrogate iteration method for solving the ROF-model [31]. Then in section "Hierarchical Spatially Adaptive Algorithm" we incorporate spatially adaptive regularization into the surrogate iteration. We further accelerate the spatial adaptive algorithm by hierarchical decomposition.

ROF-Model and Surrogate Iteration

Our variational paradigm is chosen to follow Rudin et al. [31], which allows to preserve edges in images. Further, due to the properties of the noise term η in (1.4), the ROF-model restores the image by solving the following constrained optimization problem:

$$\begin{aligned} &\text{minimize (min) } |Du|(\Omega) \quad \text{over } u \\ &\text{subject to (s.t.) } \int_{\tilde{\Lambda}} Ku\, dy = \int_{\tilde{\Lambda}} f\, dy, \\ &\qquad\qquad \int_{\tilde{\Lambda}} |Ku - f|^2 dy = \sigma^2 |\tilde{\Lambda}|. \end{aligned} \tag{1.6}$$

Usually (1.6) is addressed via the following unconstrained optimization problem:

$$\min_u |Du|(\Omega) + \frac{\lambda}{2} \int_{\tilde{\Lambda}} |Ku - f|^2\, dy \tag{1.7}$$

for a given constant $\lambda > 0$. Note that, since $Ku - f \in \operatorname{Ran} S$, the objective in (1.7) remains unchanged if the integration in the second term of the objective is performed over Λ rather than $\tilde{\Lambda}$. Assuming that K does not annihilate constant functions, one can show that there exists a constant $\lambda \geq 0$ such that the constrained problem (1.6) is equivalent to the unconstrained problem (1.7); see [6].

Our purpose is to modify the objective in (1.7) in order to handle a spatially variant parameter λ over the image domain Ω and the operator $K\colon L^2(\Omega) \to L^2(\Lambda)$. Note that this can not be done directly by inserting λ on the integral over $\tilde{\Lambda}$ in (1.7) since we require λ to be defined over Ω. Hence, instead of tackling (1.7) directly we introduce a so-called *surrogate functional* $\mathbb{S}$ [12]. In this vein, for given $a \in L^2(\Omega)$, $\mathbb{S}$ is defined as

$$\begin{aligned} \mathbb{S}(u, a) &:= |Du|(\Omega) + \frac{\lambda}{2}\Big(\|Ku - f\|^2_{L^2(\Lambda)} + \delta \|u - a\|^2_{L^2(\Omega)} - \|K(u - a)\|^2_{L^2(\Lambda)} \Big) \\ &= |Du|(\Omega) + \frac{\lambda\delta}{2} \|u - f_K(a)\|^2_{L^2(\Omega)} + \phi(a, K, f, \lambda), \end{aligned} \tag{1.8}$$

with

$$f_K(a) := a - \frac{1}{\delta} K^*(Ka - f) \in L^2(\Omega),$$

where we assume $\delta > 1$. Since $\|S^*\| = \|S\| \leq 1$ and $\|T^*\| = \|T\| = 1$, we have $\|K\| \leq 1 < \delta$. We note that here and below $\|\cdot\|$ denotes the operator norm $\|\cdot\|_{\mathscr{L}(L^2(\Omega))}$. We also emphasize that ϕ is a function independent of u. It is readily observed that minimization of $\mathbb{S}(u, a)$ over u is no longer affected by the action of K. Rather, minimizing $\mathbb{S}(u, a)$ for fixed a resembles a typical image denoising problem. In order to approach a solution of (1.7), we consider the following iteration.

Surrogate Iteration: Choose $u^{(0)} \in L^2(\Omega)$. Then compute for $k = 0, 1, 2, \ldots$

$$u^{(k+1)} := \arg\min_u |Du|(\Omega) + \frac{\delta}{2}\int_\Omega \lambda |u - f_K^{(k)}|^2 dx. \tag{1.9}$$

with $f_K^{(k)} := f_K(u^{(k)})$.

It can be shown that the iteration (1.9) generates a sequence $(u^{(k)})_{k\in\mathbb{N}}$ which converges to a minimizer of (1.7); see [12, 13]. Moreover, the minimization problem in (1.9) is strictly convex and can be efficiently solved by standard algorithms such as the primal-dual first-order algorithm [5], the split Bregman method [19], or the primal-dual semismooth Newton algorithm [24].

For a constant $\lambda > 0$, the above iteration can be formulated as a forward-backward splitting algorithm: Let $F_1(u) := |Du|(\Omega)$ and $F_2(u) := \frac{\lambda}{2}\int_\Omega |Ku - f|^2 dx$, and define the proximal operator

$$\mathrm{prox}_{\gamma,F_1}(u) := \mathrm{argmin}_w \left(F_1(w) + \frac{1}{2\gamma}\int_\Omega |u - w|^2 dx \right).$$

Then, (1.9) is equivalent to

$$u^{(k+1)} = \mathrm{prox}_{\frac{1}{\delta\lambda},F_1}\left(u^k - \frac{1}{\delta\lambda}\nabla F_2(u^k) \right).$$

A different scenario is present if instead we consider a spatially adapted λ as we do next.

Hierarchical Spatially Adaptive Algorithm

The problem in (1.9) is related, via Lagrange multiplier theory, to the globally constrained minimization problem

$$\min_u |Du|(\Omega) \quad \text{s.t.} \int_\Omega |u - f_K^{(k)}|^2 dx \le A, \tag{1.10}$$

where $A > 0$ is a constant depending on σ and K; see [6]. In order to enhance image details while preserving homogeneous regions, we localize the constraint in (1.10), which leads to the modified variational model:

$$\min_u |Du|(\Omega) \quad \text{s.t. } \mathscr{S}(u) \le A \quad \text{a.e. in } \Omega. \tag{1.11}$$

Here the local variance term $\mathscr{S}(u)(\cdot) := \int_\Omega w(\cdot,x)|u - f_K(u)|^2(x)dx$ is defined for some given localization filter w. A popular choice for w, utilized in what follows, is a window type filter. Thus the constraint in (1.11) with $u = u^{(k+1)}$ reads

$$\mathscr{S}(u^{(k+1)})(\cdot) = \int_\Omega w(\cdot,x)\left|u^{(k+1)} - u^{(k)} + \frac{1}{\delta}K^*(Ku^{(k)} - f)\right|^2(x)dx \le A. \tag{1.12}$$

Given the convergence result, as $k \to \infty$, for scalar λ alluded to in connection with (1.9), one expects the term $u^{(k+1)} - u^{(k)}$ to vanish. This indicates that $\int_\Omega w(\cdot,x)|\frac{1}{\delta}K^*(Ku^{(k)} - f)|^2(x)dx \le A$ is expected in the limit. This consideration leads to the following pointwisely constrained optimization problem:

$$\min_u |Du|(\Omega) \quad \text{s.t.} \int_\Omega w(\cdot,x)\left|\frac{1}{\delta}K^*(Ku - f)\right|^2(x)dx \le A \quad \text{a.e. in } \Omega. \tag{1.13}$$

Next we discuss the choice of A. In view of the (global) estimate for the *backprojected residual* $K^*(K\widehat{u} - f)$, i.e.,

$$\|K^*(K\widehat{u} - f)\|^2_{L^2(\Omega)} \le \|K^*\|^2\|K\widehat{u} - f\|^2_{L^2(\Lambda)} \le \sigma^2|\tilde{\Lambda}|,$$

we thus choose

$$A := \frac{\sigma^2|\tilde{\Lambda}|}{\delta^2}.$$

In deriving the above inequalities, we have used the facts that $\|K^*\| = \|K\| \le 1$ and $\|K\widehat{u} - f\|^2_{L^2(\Lambda)} = \sigma^2|\tilde{\Lambda}|$.

In a discrete setting, we now describe a strategy, based on a statistical local variance estimator, to adapt the spatially variant regularization parameter λ. The idea behind considering a spatially varying λ (instead of a constant one) is motivated by the fact that the constraint in (1.13) is spatially dependent, in contrast to the one in (1.10); see [15] for further discussion. For this purpose, consider a discrete image u defined over the discrete 2D index set Ω_h (of cardinality $|\Omega_h|$), whose nodes lie on a regular grid of uniform mesh size $h := \sqrt{1/|\Omega_h|} \in \mathbb{N}$. The total variation of a discrete image u is denoted by $|Du|(\Omega_h)$; see (1.15) below for a precise definition. We also define the residual image associated with $f_K(\cdot)$ by

$$r(u) := f_K(u) - u.$$

Concerning the filter w associated with $\mathscr{S}$ in (1.11), we exemplarily choose the mean filter pertinent to a square window centered at x. For this reason and in our discrete setting, we define the averaging window

$$\Omega^\omega_{i,j} := \left\{(i + hs, j + ht) : s,t \in \left[-\frac{\omega-1}{2}, \frac{\omega-1}{2}\right] \cap \mathbb{Z}\right\},$$

where $\omega > 1$ is an odd integer representing the window size, and then compute the estimated local variance at $(i, j) \in \Omega_h$ by

$$\mathscr{S}^{\omega}(u)_{i,j} := \frac{1}{\omega^2} \sum_{(\tilde{i},\tilde{j})\in\Omega^{\omega}_{i,j}} \left| r(u)_{\tilde{i},\tilde{j}} \right|^2 .$$

Given the reconstruction u_n associated with λ_n, we use $\mathscr{S}^{\omega}(u_n)$ to check whether λ_n should be updated or it already yields a successful reconstruction u_n. In particular, motivated by Dong et al. [15], we intend to increase λ_n at the pixels where the corresponding local variance violates the upper estimate A. More specifically, we utilize the following update rule:

$$(\lambda_{n+1})_{i,j} = \frac{\zeta_n}{\omega^2} \sum_{(\tilde{i},\tilde{j})\in\Omega^{\omega}_{i,j}} \min\left\{\bar{\lambda}, \left((\lambda_n)_{\tilde{i},\tilde{j}} + \rho_n \|\lambda_n\|_{\ell^\infty}\left(\sqrt{\tilde{\mathscr{S}}^{\omega}(u_n)_{\tilde{i},\tilde{j}}/A} - 1\right)\right)\right\}. \tag{1.14}$$

Here

$$\tilde{\mathscr{S}}^{\omega}(u)_{i,j} := \begin{cases} \mathscr{S}^{\omega}(u)_{i,j}, & \text{if } \mathscr{S}^{\omega}(u)_{i,j} > A, \\ A, & \text{otherwise,} \end{cases}$$

$\bar{\lambda} > 0$ is a prescribed upper bound, and $\|\lambda_n\|_{\ell^\infty}$ is a scaling factor suggested in [15]. Two step-size parameters, $\zeta_n > 1$ and $\rho_n > 0$, will allow a backtracking procedure should λ_{n+1} be overshot by (1.14), on which we refer to the HSA algorithm below for a more detailed account.

We are now ready to present our (basic) spatially adaptive (SA) image reconstruction algorithm.

SA Algorithm: Initialize $u_0 \in \mathbb{R}^{\Omega_h}$, $\lambda_1 \in \mathbb{R}^{\Omega_h}_+$, $n := 1$. Iterate as follows until a stopping criterion is satisfied:

1) Set $u_n^{(0)} := u_{n-1}$. For each $k = 0, 1, 2, \ldots$, compute $u_n^{(k+1)}$ according to

$$u_n^{(k+1)} := \arg\min_u |Du|(\Omega_h) + \frac{\delta h^2}{2} \sum_{(i,j)\in\Omega_h} (\lambda_n)_{i,j} \left| (u - f_n^{(k)})_{i,j} \right|^2,$$

with $f_n^{(k)} := u_n^{(k)} - \frac{1}{\delta} K^*(K u_n^{(k)} - f)$. Let u_n be the outcome of this iteration.

2) Update λ_{n+1} according to (1.14). Set $n := n + 1$.

Following [15] we further accelerate the SA algorithm by employing a hierarchical decomposition of the image into scales. This idea, introduced by Tadmor, Nezzar and Vese in [36, 37], utilizes concepts from interpolation theory to represent a noisy image as the sum of "atoms" $u_{(l)}$, where every $u_{(l)}$ extracts features at a scale finer than the one of the previous $u_{(l-1)}$. This method acts like an iterative regularization scheme, i.e., up to some iteration number $\bar{l}$ the method yields improvement on reconstruction results with a deterioration (due to noise influence and ill-conditioning) beyond $\bar{l}$.

Here we illustrate the basic workflow of hierarchical decomposition in a denoising problem (i.e., where K equals the identity). Given the exponential scales $\{\zeta^l \lambda_0 : l = 0, 1, 2, \ldots\}$ with $\lambda_0 \in \mathbb{R}_+^{\Omega_h}$ and $\zeta > 1$, the hierarchical decomposition operates as follows:

1. Initialize $u_0 \in \mathbb{R}^{\Omega_h}$ by

$$u_0 := \arg\min_u |Du|(\Omega_h) + \frac{h^2}{2} \sum_{(i,j)\in\Omega_h} (\lambda_0)_{i,j} \left|(u - f)_{i,j}\right|^2 .$$

2. For $l = 0, 1, \ldots$, set $\lambda_{l+1} := \zeta \lambda_l$ and $v_l := f - u_l$. Then compute

$$d_l := \arg\min_u |Du|(\Omega_h) + \frac{h^2}{2} \sum_{(i,j)\in\Omega_h} (\lambda_{l+1})_{i,j} \left|(u - v_l)_{i,j}\right|^2 ,$$

and update $u_{l+1} := u_l + d_l$.

Now we incorporate such a hierarchical decomposition into the SA algorithm, which we shall refer to as the hierarchical spatially adaptive (HSA) algorithm. We note that all minimization (sub)problems in the HSA algorithm are solved by the primal-dual Newton method in [24]. There, the original ROF-model is approximated by a variational problem posed in $H_0^1(\Omega)$ via adding an additional regularization term $\frac{\mu}{2}\|\nabla u\|^2_{L^2(\Omega)}$, with $0 < \mu \ll 1/(\operatorname{ess\,sup} \lambda)$, to the objective and assuming, without loss of generality, homogeneous Dirichlet boundary conditions. In this case, the (discrete) total variation is given by

$$|Du|(\Omega_h) = h \sum_{(i,j)\in\Omega_h} \Big(|u_{i+1,j} - u_{i,j}| + |u_{i,j+1} - u_{i,j}| \Big), \tag{1.15}$$

with $u_{i,j} = 0$ whenever $(i, j) \notin \Omega_h$. We refer to [24] for a detailed account of this algorithm.

HSA Algorithm: Input parameters $\delta > 1$, $\omega \in 2\mathbb{N}+1$. Initialize $u_0 \in \mathbb{R}^{\Omega_h}$, $\lambda_1 \in \mathbb{R}_+^{\Omega_h}$ (sufficiently small), $\zeta_0 > 1$, $\rho_0 > 0$.

1) Set $u_0^{(0)} := u_0$. For each $k = 0, 1, 2, \ldots, \kappa_0$, compute $u_0^{(k+1)}$ by

$$u_0^{(k+1)} := \arg\min_u |Du|(\Omega_h) + \frac{\delta h^2}{2} \sum_{(i,j)\in\Omega_h} (\lambda_1)_{i,j} \left|(u - f_0^{(k)})_{i,j}\right|^2,$$

with $f_0^{(k)} := u_0^{(k)} - \frac{1}{\delta} K^*(K u_0^{(k)} - f)$. Let u_1 be the outcome of this iteration, and set $n := 1$.

2) Set $v_n := f - K u_{n-1}$ and $d_n^{(0)} := 0$. For each $k = 0, 1, 2, \ldots, \kappa_n$, compute $d_n^{(k+1)}$ by

$$d_n^{(k+1)} := \arg\min_u |Du|(\Omega_h) + \frac{\delta h^2}{2} \sum_{(i,j)\in\Omega} (\lambda_n)_{i,j} \left|(u - f_n^{(k)})_{i,j}\right|^2,$$

with $f_n^{(k)} := d_n^{(k)} - \frac{1}{\delta} K^*(K d_n^{(k)} - v_n)$. Let d_n be the outcome of this iteration, and update $u_n := u_{n-1} + d_n$.

3) Evaluate the (normalized) data-fitting error

$$\theta_n := \frac{\|K u_n - f\|_{\ell^2}^2}{\sigma^2 |\tilde{\Lambda}_h|}.$$

If $\theta_n > 1$, then set $\tilde{n} := n$, $\zeta_n := \zeta_{n-1}$, $\rho_n := \rho_{n-1}$, and continue with step 4;
If $0.8 \le \theta_n \le 1$, then return u_n, λ_n and stop;
If $\theta_n < 0.8$, then set $u_n := u_{\tilde{n}}$, $\lambda_n := \lambda_{\tilde{n}}$, $\zeta_n := \sqrt{\zeta_{n-1}}$, $\rho_n := \rho_{n-1}/2$, and continue with step 4.

4) Update λ_{n+1} according to formula (1.14). Set $n := n + 1$ and return to step 2.

We also remark that the initial $\lambda_1 \in \mathbb{R}_+^{\Omega_h}$ should be sufficiently small such that the resulting normalized data-fitting error θ_1 is much larger than 1. Then the HSA iterations are responsible for (monotonically) lifting up λ_n in a spatially adaptive fashion as described earlier in this paper. Such a lifting is performed until the data-fitting error $\|K u_n - f\|_{\ell^2}^2 / |\tilde{\Lambda}_h|$ approaches the underlying noise level σ^2. If the data-fitting error drops too far below σ^2, then the algorithm may suffer from overfitting the noisy data. In this scenario, we backtrack on λ_n through potential reduction of ζ_n and ρ_n; see step 3 of the HSA algorithm.

Numerical Experiments

In this section, we present numerical results of the newly proposed HSA algorithm for two applications, namely reconstruction from partial Fourier data and wavelet inpainting. All experiments reported here were performed under Matlab. The image intensity is scaled to the interval $[0, 1]$ in advance of our computation. For the HSA algorithm, we always choose the following parameters: $\delta = 1.2$, $\omega = 11$, $\zeta_0 = 2$, $\rho_0 = 1$, $\bar{\lambda} = 10^6$, $u_0 = K^* f$. In the primal-dual Newton algorithm [24], we choose the H^1-regularization parameter $\mu = 10^{-4}$, the Huber smoothing parameter $\gamma = 10^{-3}$, and terminate the overall Newton iterations as soon as the initial residual norm is reduced by a factor of 10^{-4}. Besides, the maximum iteration numbers $\{\kappa_n\}$ for the surrogate iterations are adaptively chosen such that $\|d_n^{(\kappa_n)} - d_n^{(\kappa_n - 1)}\|_{\ell^2} \leq 10^{-6}\sqrt{|\Omega_h|}$.

The images restored by HSA are compared, both visually and quantitatively, with the ones restored by the variational model in (1.7) with *scalar-valued* λ. For quantitative comparisons among restorations, we evaluate their peak signal-to-noise ratios (PSNR) [3] and also the structural similarity measures (SSIM) [39]; see Table 1.1. To optimize our choice for each scalar-valued λ, we adopt a bisection procedure, up to a relative error of 0.02, i.e., $|\lambda^{k+1} - \lambda^k| < 0.02\lambda^k$, to maximize the following weighted sum of the PSNR- and SSIM-values of the resulting scalar-λ restoration

Table 1.1 Comparisons with respect to PSNR and SSIM

		Cameraman				Knee			
Fourier		Scalar-valued λ		HSA		Scalar-valued λ		HSA	
σ	#rad'l	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.05	75	26.7895	0.8051	**26.9559**	**0.8124**	30.4347	0.8247	**30.5399**	**0.8290**
0.05	90	**27.5399**	0.8215	27.5020	**0.8262**	30.8355	0.8337	**30.9442**	**0.8389**
0.05	105	28.1553	0.8307	**28.1667**	**0.8346**	31.1155	0.8402	**31.3328**	**0.8478**
0.1	75	24.9336	0.7576	**25.1809**	**0.7639**	28.2375	0.7570	**28.4896**	**0.7639**
0.1	90	25.2738	0.7666	**25.7072**	**0.7775**	28.4811	0.7627	**28.7140**	**0.7721**
0.1	105	25.6780	0.7740	**26.2317**	**0.7843**	28.5856	0.7662	**28.8373**	**0.7745**

		Cameraman				Barbara			
Wavelet		Scalar-valued λ		HSA		Scalar-valued λ		HSA	
σ	s.r.	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.05	2.5%	24.0319	0.7388	**24.5702**	**0.7436**	22.9489	0.6174	**24.4184**	**0.6777**
0.05	5%	26.5279	**0.7969**	**27.1539**	0.7964	24.6622	0.6922	**26.1698**	**0.7438**
0.05	10%	28.6248	**0.8374**	**29.5812**	0.8351	26.5317	0.7645	**27.8187**	**0.8083**
0.1	2.5%	23.7416	0.7301	**24.1510**	**0.7326**	22.7299	0.6067	**24.0291**	**0.6605**
0.1	5%	25.7625	0.7786	**26.5195**	**0.7791**	24.1307	0.6733	**25.2635**	**0.7095**
0.1	10%	27.3033	**0.8136**	**27.5671**	0.7937	25.4469	0.7410	**26.3245**	**0.7591**

Bold values in the table correspond to the best in their respective classes

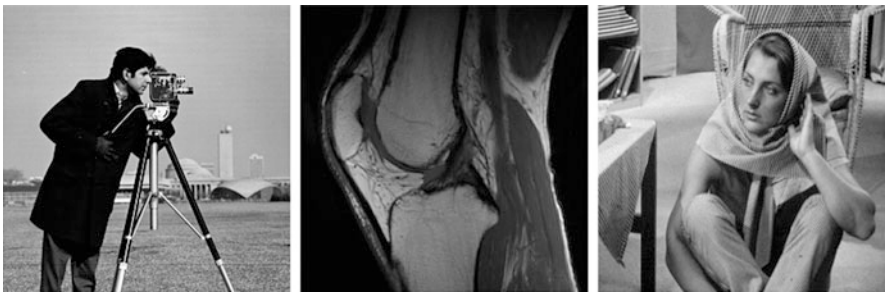


Fig. 1.1 Test images (from left to right): "Cameraman", "Knee", and "Barbara"

$$\frac{\mathrm{PSNR}(\lambda)}{\max\{\mathrm{PSNR}(\tilde{\lambda}) : \tilde{\lambda} \in I\}} + \frac{\mathrm{SSIM}(\lambda)}{\max\{\mathrm{SSIM}(\tilde{\lambda}) : \tilde{\lambda} \in I\}}$$

over the interval $I = [10^2, 10^5]$. The maximal PSNR and SSIM in the above formula are pre-computed up to a relative error of 0.001. The original images used for our numerical tests are given in Fig. 1.1.

Reconstruction of Partial Fourier Data

In magnetic resonance imaging, one aims to reconstruct an image which is only sampled by partial Fourier data and additionally distorted by additive white Gaussian noise of zero mean and standard deviation σ. Here the data-formation operator is given by $K = S \circ T$, where T is a 2D (discrete) Fourier transform and S represents a downsampling of Fourier data. In particular, we consider S which picks Fourier data along radial lines centered at zero frequency.

Our experiments are performed for the test images "Cameraman" and "Knee" with $\sigma \in \{0.05, 0.1\}$ and #radials $\in \{75, 90, 105\}$ respectively. In these experiments, we have always initialized HSA with $\lambda_1 = 100$. The resulting restorations via the total-variation method with scalar-valued λ and via our HSA method are both displayed in Figs. 1.2, 1.3 and 1.4. We also show the ultimate spatially adapted λ from HSA in each test run, where the light regions in the λ-plot correspond to high values of λ and vice versa. It is observed that the values of λ in regions containing detailed features (e.g. the camera and the tripod in "Cameraman") typically outweigh its values in more homogeneous regions (e.g. the background sky in "Cameraman"). As a consequence, this favorably yields a sharper background-versus-detail contrast in the restored images via HSA. In Fig. 1.3, we observe face and camera of "Cameraman" reconstructions with a better performance of our HSA method. According to the quantitative comparisons reported in Table 1.1, HSA almost always outperforms scale-valued λ in terms of PSNR and SSIM. As a side remark, it is also observed that the spatially adapted λ via HSA is able to capture more features of the underlying image at a lower noise level (Fig. 1.4).

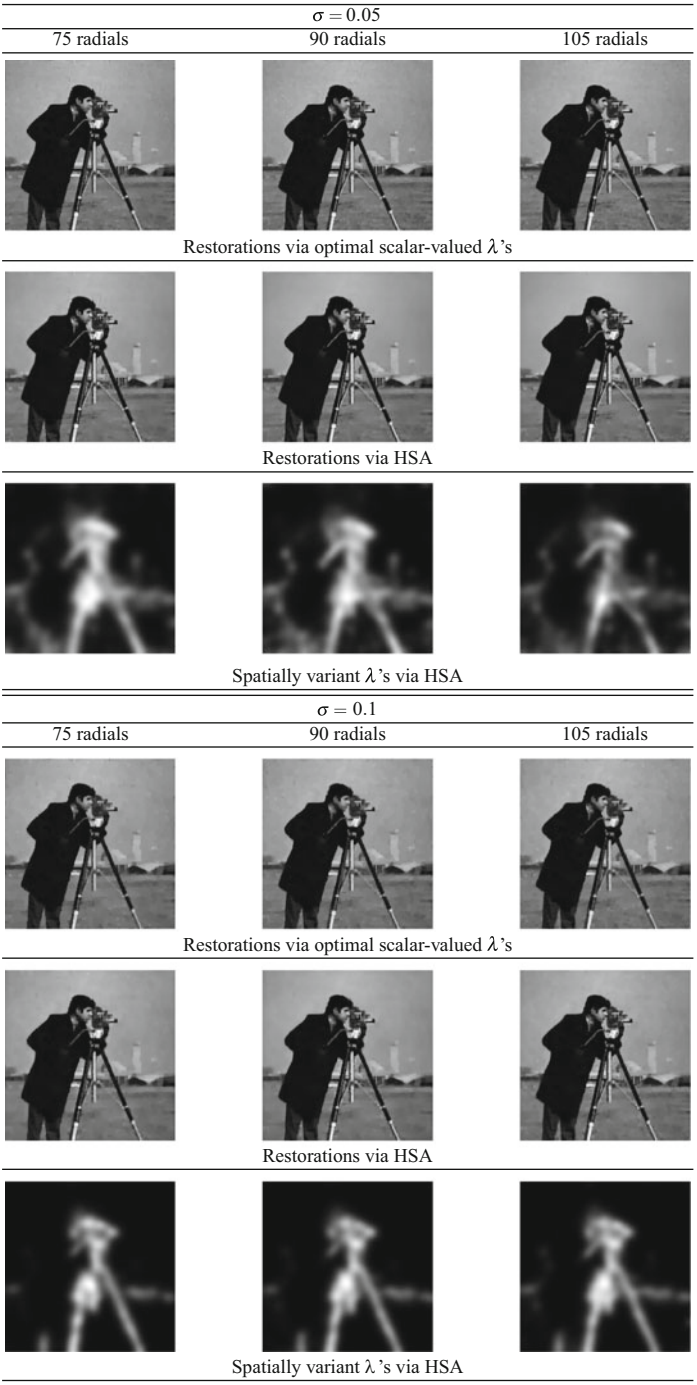


Fig. 1.2 Reconstruction of partial Fourier data on "Cameraman"

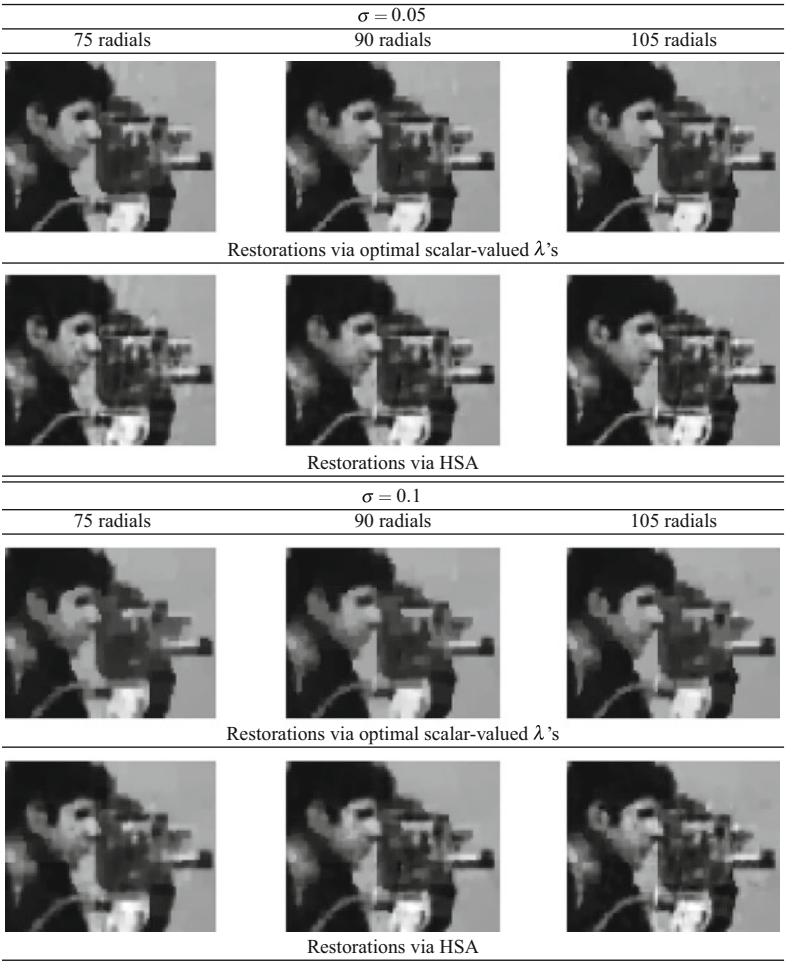


Fig. 1.3 Zoom in for reconstructions of partial Fourier data on "Cameraman"

To test the robustness of HSA, we perturb our choices of the window size ω and the initial choice of λ in our experiments. In Fig. 1.5, we report the resulting PSNRs and SSIMs of such sensitivity tests on the particular Fourier-Cameraman example with $\sigma = 0.05$ and #radials $= 90$. It is observed that HSA behaves relatively stable with different choices of ω. On the other hand, one should be cautioned that the results of HSA deteriorate as the initial λ is chosen too large. Nevertheless, among all initial λ's smaller than a certain threshold (in this case 200), smaller choices do not always claim advantages over larger ones.

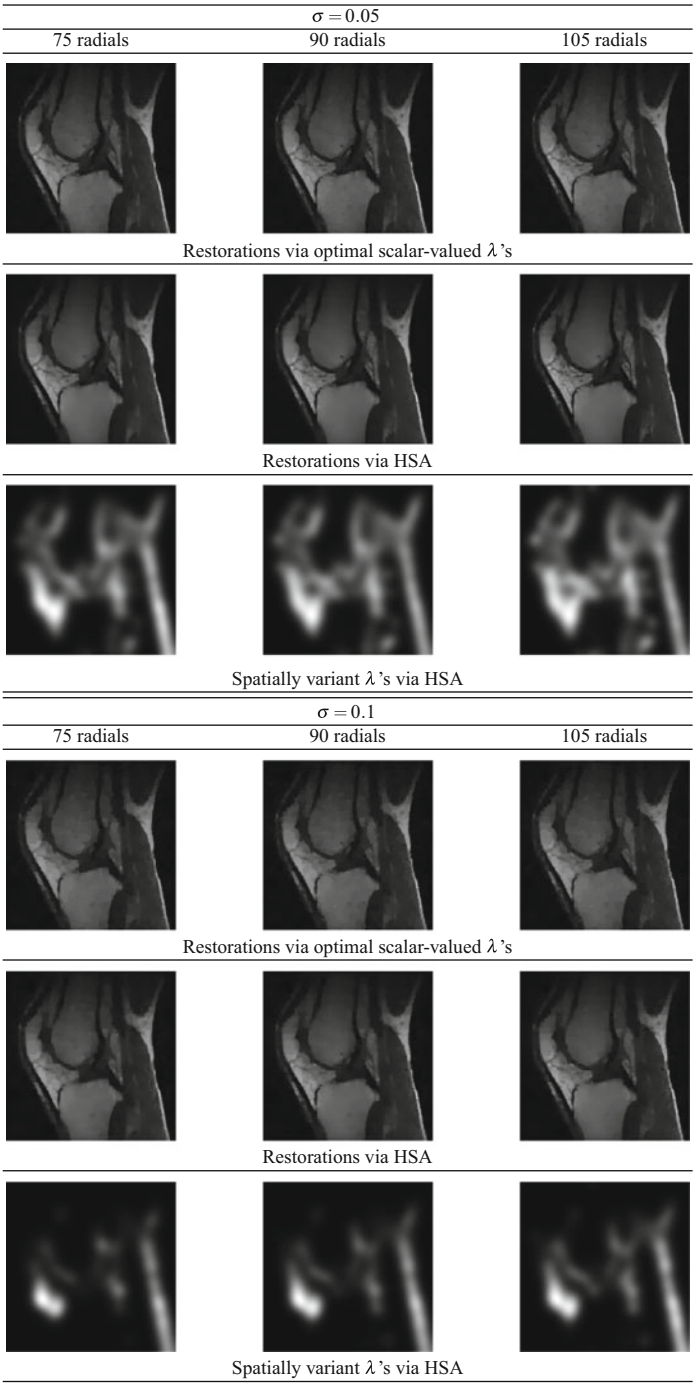


Fig. 1.4 Reconstruction of partial Fourier data on "Knee"

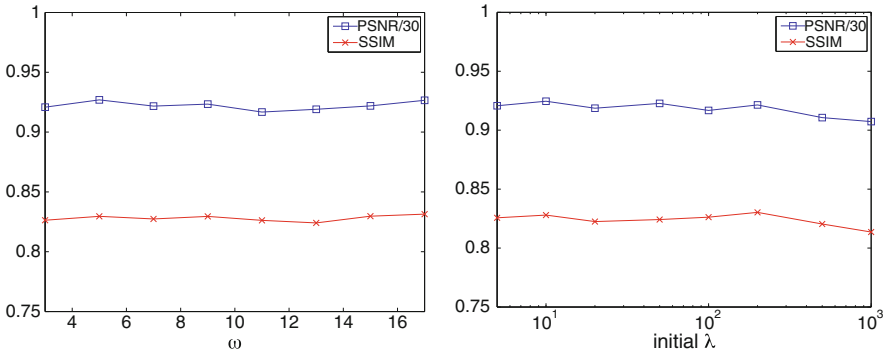


Fig. 1.5 Sensitivity test: image = "Cameraman", $\sigma = 0.05$, #radials = 90

Wavelet Inpainting

Wavelet inpainting is about restoring missing wavelet coefficients due to lossy compression or error-prone data transmission; see, e.g., [7, 44]. Here we consider the scenario where a test image is compressed by storing the largest Daubechies-4 wavelet coefficients [11] in magnitude only up to a small sampling rate (s.r.), namely s.r. $\in \{2.5\%, 5\%, 10\%\}$. The compressed wavelet coefficients are further contaminated by additive white Gaussian noise of mean zero and standard deviation $\sigma \in \{0.05, 0.1\}$. For wavelet inpainting, we have initialized HSA with $\lambda_1 = 10$. The experiments are performed for the test images "Cameraman" and "Barbara", and the corresponding results, both restored images and the adapted λ's, are shown in Figs. 1.6 and 1.7. A detailed view of the face region of "Barbara" is given in Fig. 1.8, where the results are in favor of our HSA algorithm.

In the wavelet-Cameraman example, the results via scalar-valued λ's and HSA are almost identical to human eyes. Even though, HSA always outperforms the scale-valued λ in terms of PSNR, while the SSIM-comparison is somewhat even; see Table 1.1. Interestingly, the adapted λ's in this example exhibit patterns analogous to the ones in the Fourier-Cameraman example.

Our HSA method gains more advantages when it is applied to the "Barbara" image with a stronger cartoon-texture contrast than "Cameraman". In Fig. 1.7, it is witnessed that the restored images via scalar-valued λ's suffer from undesirable staircase effects. In comparison, spatially adapted λ's yield significant improvements on the restorations, even in the cases where the pattern of λ is less transparent due to lack of data or strong noise. In Table 1.1, the PSNR- and SSIM-comparisons also dominantly favor the HSA method.

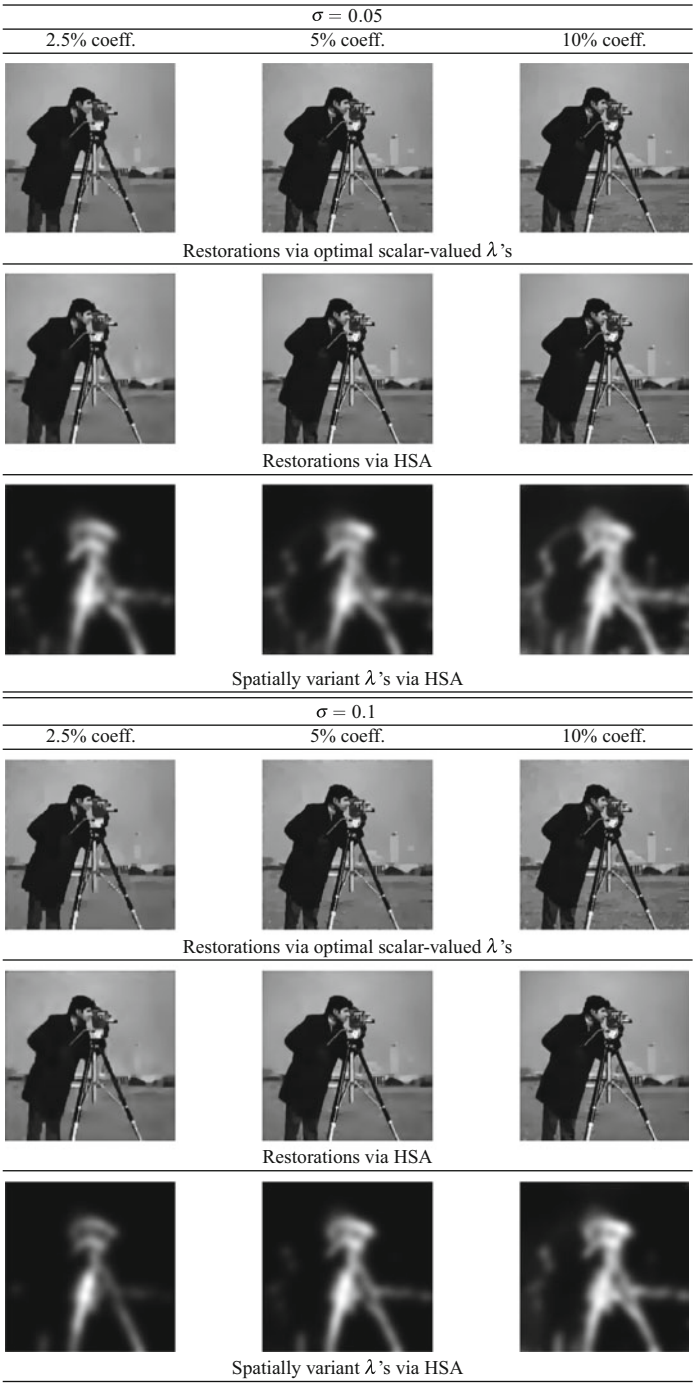


Fig. 1.6 Wavelet inpainting on "Cameraman"

Fig. 1.7 Wavelet inpainting on "Barbara"

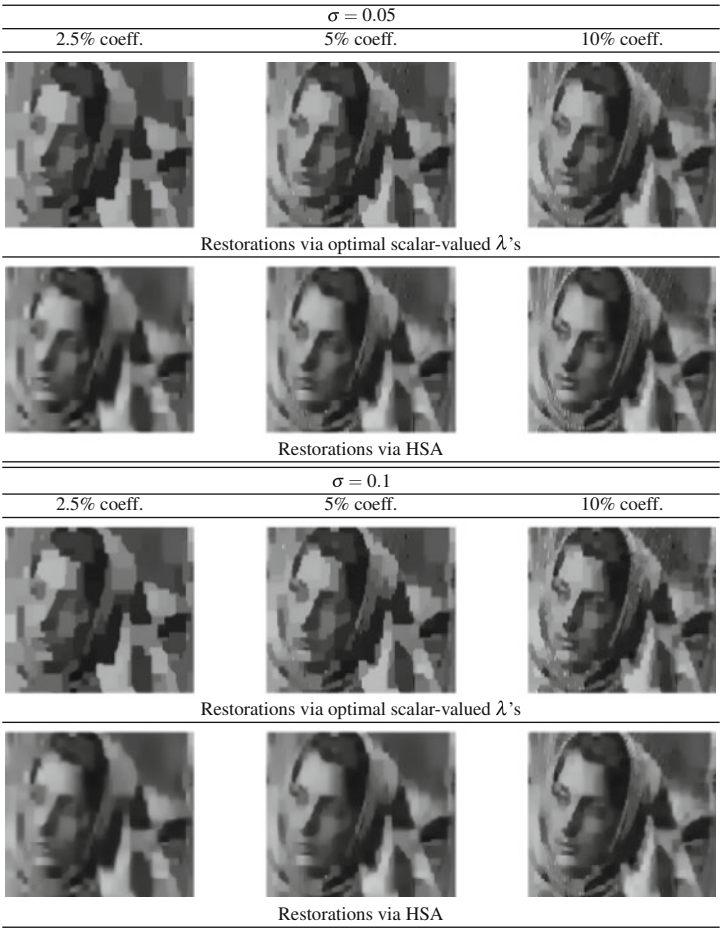


Fig. 1.8 Zoomed view on wavelet inpainting on "Barbara"

Qualitative Relation to Other Spatially Distributed Parameter Methods

A particular feature of the HSA algorithm is that it allows to assign a spatially variant parameter λ, on the image domain Ω, associated to a data fidelity term that is determined by an integral over $\tilde{\Lambda} \subset \Lambda$, a non-image domain. Such configuration renders certain variational methods with spatially variant λ's not applicable: For example, the SATV algorithm developed in [15] requires K to map into functions over the image domain Ω. This obstacle has been overcome in [22] and [25] where the spatially variant parameter is not longer related to the data fidelity term, but rather to the regularization functional. Specifically, a parameter $\alpha : \Omega \to \mathbb{R}$ in the

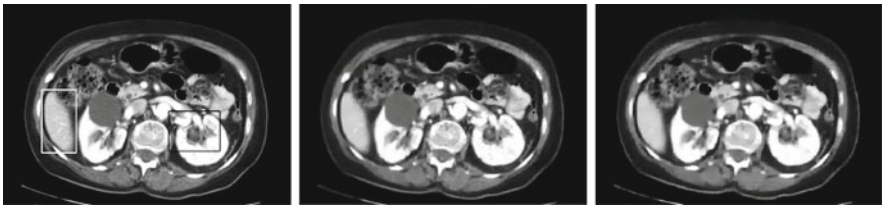
model

$$\min_u \int_\Omega \alpha(x)|Du| + \frac{1}{2}\int_{\tilde{\Lambda}} |Ku - f|^2\, dy, \tag{1.16}$$

is automatically selected based on a bilevel formulation involving also localized variance estimators.

For non-negative constants λ and α in (1.7) and (1.16), respectively, it holds true that if $\lambda = \alpha^{-1}$ solutions of the optimization problems are identical. In the spatially variant case for λ and α, the relationship between the parameters does no longer hold exactly when automatically chosen via local variance estimates of reconstructions, i.e., we only expect $\lambda(x) \simeq (\alpha(x))^{-1}$ for $x \in \Omega$. One specific difference between the two approaches is related to the fact that λ is only required to be essentially bounded while (in the function space setting) α requires to have higher regularity for the objective in (1.16) to be well-defined. The latter translates into the need of having an additional regularization term for the smoothing of α in the upper level objective. This has a clear consequence in differences for λ and α for the HSA algorithm and the bilevel method in [22] and [25], respectively: λ seems to be able to have more variability than α on Ω. On the other hand, although λ and α have, in general, high and low values on details, respectively, α seems to decrease on edges more drastically, while λ has slower transitions there. In particular, the previous explains how the selection of α is a preferable choice over the one of λ in images with large homogeneous regions, sharp edges, and corners, and vice versa for images with significant number of details on small regions and certain textures. A quantitative analysis for such differences is beyond the scope of the paper, and an active research direction.

In order to compare with the HSA method, we consider the spatially distributed method described in [22] and [25], where α is chosen in (1.16) via a bilevel formulation. We define $K = S \circ T$ to collect Fourier coefficients along 120 radial lines centered at zero frequency, and take data distorted by additive white Gaussian noise of zero mean and standard deviation $\sigma = 0.05$. For the bilevel scheme, we utilized the same configuration and parameters as in [25], where the local variance bounds are chosen as in (#2); see [25] for all details. For the HSA algorithm, we use the same setup as above in the reconstruction of partial Fourier data. When K maps into functions over the image domain, it was observed that the bilevel formulation provides, in general, reconstructions with better SSIM than the ones from the SATV algorithm in [15]. However, the SATV performs better in terms of PSNR than the bilevel scheme. This same behavior is observed between the HSA algorithm and the bilevel one. Reconstructions for both methods are given in Fig. 1.9, and we take zoom views of the two framed regions in the "Chest" image for further detail comparison in Fig. 1.10. Finally, in Fig. 1.11, we observe the α parameter of the bilevel scheme, and λ^{-1} where λ is HSA parameter. As fine details are hard to observe in the black and white images, we have included red colored plots of the surfaces associated with both parameters with a specific light effect to show such details.



(a) (b) (c)

Fig. 1.9 Fourier inpainting: "Chest". (**a**) "Chest" image. (**b**) Bilevel restoration. PSNR: 28.8837—SSIM:**0.8406**. (**c**) HSA restoration. PSNR:**29.2488**—SSIM:0.8282

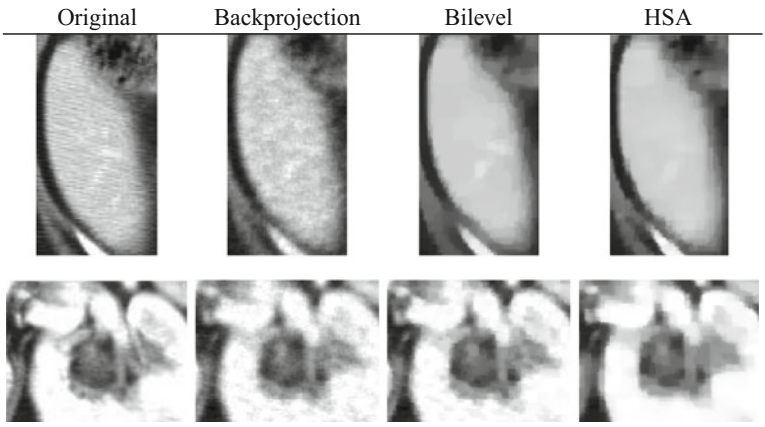
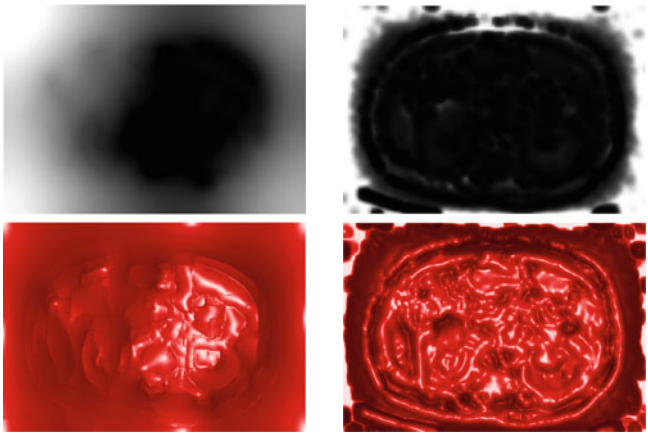


Fig. 1.10 "Chest": zoomed views



(a) (b)

Fig. 1.11 Fourier inpainting parameters. (**a**) α in bilevel. (**b**) λ^{-1} in HSA

Conclusion

In this work, it has been shown that spatially adapted data fidelity weights help to improve the quality of restored images. The automated adjustment of the local weights is developed based on the localized image residuals. Such a parameter adjustment scheme can be further accelerated by employing hierarchical decompositions, which aim at decomposing an image into so-called atoms at different scales. The framework of the paper is suitable for subsampled data in non-image domain, in particular incomplete coefficients from orthogonal Fourier- and wavelet transforms as illustrated in the numerical experiments.

Acknowledgements This research was supported by the Austrian Science Fund (FWF) through START-Project Y305 "Interfaces and Free Boundaries" and SFB-Project F3204 "Mathematical Optimization and Applications in Biomedical Sciences", the German Research Foundation DFG through Project HI1466/7-1 "Free Boundary Problems and Level Set Methods", as well as the Research Center MATHEON through Project C-SE15 "Optimal Network Sensor Placement for Energy Efficiency" supported by the Einstein Center for Mathematics Berlin.

References

1. A. Almansa, C. Ballester, V. Caselles, G. Haro, A TV based restoration model with local constraints. J. Sci. Comput. **34**, 209–236 (2008)
2. M. Bertalmio, V. Caselles, B. Rougé, A. Solé, TV based image restoration with local constraints. J. Sci. Comput. **19**, 95–122 (2003)
3. A. Bovik, *Handbook of Image and Video Processing* (Academic, San Diego, 2000)
4. A. Buades, B. Coll, J.-M. Morel, A review of image denoising algorithms, with a new one. Multiscale Model. Simul. **4**, 490–530 (2005)
5. A. Chambolle, An algorithm for total variation minimization and applications. J. Math. Imaging Vision **20**, 89–97 (2004)
6. A. Chambolle, P.-L. Lions, Image recovery via total variation minimization and related problems. Numer. Math. **76**, 167–188 (1997)
7. T.F. Chan, J. Shen, H.-M. Zhou, Total variation wavelet inpainting. J. Math. Imaging Vision **25**, 107–125 (2006)
8. Q. Chang, I.-L. Chern, Acceleration methods for total variation based image denoising. SIAM J. Appl. Math. **25**, 982–994 (2003)
9. Q. Chen, P. Montesinos, Q.S. Sun, P.A. Heng, D.S. Xia, Adaptive total variation denoising based on difference curvature. Image Vis. Comput. **28**, 298–306 (2010)
10. K. Chen, E.L. Piccolomini, F. Zama, An automatic regularization parameter selection algorithm in the total variation model for image deblurring. Numer. Algorithms **67**, 73–92 (2014)
11. I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992)
12. I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**, 1413–1457 (2004)
13. I. Daubechies, G. Teschke, L. Vese, Iteratively solving linear inverse problems under general convex constraints. Inverse Prob. Imaging **1**, 29–46 (2007)
14. D.C. Dobson, C.R. Vogel, Convergence of an iterative method for total variation denoising. SIAM J. Numer. Anal. **34**, 1779–1791 (1997)
15. Y. Dong, M. Hintermüller, M. Rincon-Camacho, Automated regularization parameter selection in multi-scale total variation models for image restoration. J. Math. Imaging Vis. **40**, 82–104 (2011)

16. K. Frick, P. Marnitz, Statistical multiresolution Dantzig estimation in imaging: fundamental concepts and algorithmic framework. Electron. J. Stat. **6**, 231–268 (2012)
17. K. Frick, P. Marnitz, A. Munk, Statistical multiresolution estimation for variational imaging: with an application in Poisson-biophotonics. J. Math. Imaging Vision **46**, 370–387 (2013)
18. R. Giryes, M. Elad, Y.C. Eldar, The projected GSURE for automatic parameter tuning in iterative shrinkage methods. Appl. Comput. Harmon. Anal. **30**, 407–422 (2011)
19. T. Goldstein, S. Osher, The split Bregman method for ℓ_1 regularized problems. SIAM J. Imaging Sci. **2**, 1311–1333 (2009)
20. C. He, C. Hu, W. Zhang, B. Shi, A fast adaptive parameter estimation for total variation image restoration. IEEE Trans. Image Process. **23**, 4954–4967 (2014)
21. M. Hintermüller, K. Kunisch, Total bounded variation regularization as bilaterally constrained optimization problem. SIAM J. Appl. Math. **64**, 1311–1333 (2004)
22. M. Hintermüller, C.N. Rautenberg, Optimal selection of the regularization function in a weighted total variation model. Part I: modelling and theory. J. Math. Imaging Vision **59**(3), 498–514 (2017)
23. M. Hintermüller, M.M. Rincon-Camacho, Expected absolute value estimators for a spatially adapted regularization parameter choice rule in L^1-TV-based image restoration. Inverse Prob. **26**, 085005 (2010)
24. M. Hintermüller, G. Stadler, An infeasible primal-dual algorithm for total bounded variation-based Inf-convolution-type image restoration. SIAM J. Sci. Comput. **28**, 1–23 (2006)
25. M. Hintermüller, C.N. Rautenberg, T. Wu, A. Langer, Optimal selection of the regularization function in a weighted total variation model. Part II: algorithm, its analysis and numerical tests. J. Math. Imaging Vision **59**(3), 515–533 (2017)
26. T. Hotz, P. Marnitz, R. Stichtenoth, L. Davies, Z. Kabluchko, A. Munk, Locally adaptive image denoising by a statistical multiresolution criterion. Comput. Stat. Data Anal. **56**(33), 543–558 (2012)
27. F. Lenzen, F. Becker, J. Lellmann, S. Petra, C. Schnörr, A class of quasi-variational inequalities for adaptive image denoising and decomposition. Comput. Optim. Appl. **54**, 371–398 (2013)
28. F. Lenzen, J. Lellmann, F. Becker, C. Schnörr, Solving quasi-variational inequalities for image restoration with adaptive constraint sets. SIAM J. Imaging Sci. **7**, 2139–2174 (2014)
29. S. Osher, M. Burger, D. Goldfarb, J. Xu, W. Yin, An iterative regularization method for total variation-based image restoration. Multiscale Model. Simul. **4**, 460–489 (2005)
30. Y. Ruan, H. Fang, Q. Chen, Semiblind image deconvolution with spatially adaptive total variation regularization. Math. Probl. Eng. **2014**(606170), 8 (2014)
31. L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
32. D. Strong, T. Chan, Spatially and scale adaptive total variation based regularization and anisotropic diffusion in image processing. Technical report, UCLA (1996)
33. D. Strong, T. Chan, Edge-preserving and scale-dependent properties of total variation regularization. Inverse Prob. **19**, 165–187 (2003)
34. D. Strong, J.-F. Aujol, T. Chan, Scale recognition, regularization parameter selection, and Meyer's G norm in total variation regularization. Technical report, UCLA (2005)
35. C. Sutour, C.-A. Deledalle, J.-F. Aujol, Adaptive regularization of the NL-means: application to image and video denoising. IEEE Trans. Image Process. **23**, 3506–3521 (2014)
36. E. Tadmor, S. Nezzar, L. Vese, A multiscale image representation using hierarchical (BV, L^2) decompositions. Multiscale Moel. Simul. **2**, 554–579 (2004)
37. E. Tadmor, S. Nezzar, L. Vese, Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation. Commun. Math. Sci. **6**, 1–26 (2008)
38. C.R. Vogel, *Computational Methods for Inverse Problems*. Frontiers in Applied Mathematics, vol. 23 (SIAM, Philadelphia, 2002)
39. Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004)
40. Y.-W. Wen, R.H. Chan, Parameter selection for total -variation-based image restoration using discrepancy principle. IEEE Trans. Image Process. **21**, 1770–1781 (2012)

41. L. Yan, H. Fang, S. Zhong, Blind image deconvolution with spatially adaptive total variation regularization. Opt. Lett. **37**, 2778–2780 (2012)
42. Q. Yuan, L. Zhang, H. Shen, Multiframe super-resolution employing a spatially weighted total variation model. IEEE Trans. Circuits Syst. Video Technol. **22**, 379–392 (2012)
43. Q. Yuan, L. Zhang, H. Shen, Regional spatially adaptive total variation super-resolution with spatial information filtering and clustering. IEEE Trans. Image Process. **22**, 2327–2342 (2013)
44. X. Zhang, T.F. Chan, Wavelet inpainting by nonlocal total variation. Inverse Prob. Imaging **4**, 191–210 (2010)

Chapter 2
A Convergent Fixed-Point Proximity Algorithm Accelerated by FISTA for the ℓ_0 Sparse Recovery Problem

Xueying Zeng, Lixin Shen, and Yuesheng Xu

Abstract We propose an approximation model of the original ℓ_0 minimization model arising from various sparse signal recovery problems. The objective function of the proposed model uses the Moreau envelope of the ℓ_0 norm to promote the sparsity of the signal in a tight framelet system. This leads to a non-convex optimization problem involved the ℓ_0 norm. We identify a local minimizer of the proposed non-convex optimization problem with a global minimizer of a related convex optimization problem. Based on this identification, we develop a two stage algorithm for solving the proposed non-convex optimization problem and study its convergence. Moreover, we show that FISTA can be employed to speed up the convergence rate of the proposed algorithm to reach the optimal convergence rate of $\mathcal{O}(1/k^2)$. We present numerical results to confirm the theoretical estimate.

Introduction

Sparse recovery problems recently attracted considerable attention in many applications such as signal processing, machine learning and computer vision. Generally, many practical problems can be formulated as

$$u = Ax + \omega, \tag{2.1}$$

X. Zeng
School of Mathematical Sciences, Ocean University of China, Qingdao, People's Republic of China

L. Shen
Department of Mathematics, Syracuse University, Syracuse, NY, USA

Y. Xu (✉)
School of Data and Computer Science, Guangdong Province Key Lab of Computational Science, Sun Yat-sen University, Guangzhou, People's Republic of China

Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA, USA
e-mail: y1xu@odu.edu; yxu06@syr.edu

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_2

where the vector $x \in \mathbb{R}^n$ denotes the signal to be recovered, the matrix $A \in \mathbb{R}^{m\times n}$ describes the system response mechanism, and the vector $\omega \in \mathbb{R}^m$ presents noise. The sparse recovery process seeks a solution of (2.1) which can be represented using only a few atoms of some redundant system. Mathematically, it can be formulated as the following constrained ℓ_0 optimization problem:

$$\min_{x\in\mathbb{R}^n} \|Dx\|_0, \quad \text{s.t. } \|Ax - u\|_2 \leq \epsilon, \tag{2.2}$$

where the ℓ_0 norm $\|x\|_0$ counts the number of nonzero components of x, ϵ is a positive parameter corresponding to the noise level of ω in (2.1), and D is a redundant dictionary. Alternately, the constrained optimization problem (2.2) may be written in its unconstrained "Lagrange" counterpart

$$\min\left\{\frac{\lambda}{2}\|Ax - u\|_2^2 + \|Dx\|_0 : x \in \mathbb{R}^n\right\}, \tag{2.3}$$

where λ is a positive parameter related to the noise level of ω. In this paper, we assume D to be a discrete tight framelet system, that is, $D^\top D = I$ with $D^\top$ being the transpose of D and I the identity matrix.

The ℓ_0 norm provides an intuitive and easily grasped penalization of sparsity. However, it is challenging to design an efficient algorithm to solve (2.2) or (2.3) directly. A widely used alternative is the convex relaxation which replaces the ℓ_0 norm with the ℓ_1 norm which is convex to take advantage of efficient algorithms in convex optimization [2, 9, 21]. Another approach is to replace the discontinuous nonconvex ℓ_0 norm by a nonconvex but continuous penalty, which is more a principle than a method. The main motivation was to overstep the bias introduced by the ℓ_1 penalty on large coefficients of the recovered signals in the transform domain [10, 25]. Such penalties include but not limit to the ℓ_p norm for $0 < p < 1$ [14], the Log-det function [11], the Mangasarian function [17], and other nonconvex functions in [20]. It has been demonstrated that the nonconvex penalties can outperform the ℓ_1 norm in various applications such as compressive sensing, matrix recovery and image processing [3, 8, 12, 14, 16]. However, these improvements are usually achieved at the cost of much higher computational complexity compared to the cost of solving the ℓ_1 norm models.

In our previous work [24], we proposed a new nonconvex approximation model for (2.2) in which the ℓ_0 norm was approximated by its continuous Moreau envelope, and a fixed-point proximity algorithm was developed when A is a partial wavelet or Fourier matrix, that is, $AA^\top = I$. The proposed model provides significant improvement for image restoration compared to the ℓ_1 relaxed model. The remarkable properties of the Moreau envelope of the ℓ_0 norm allow us to solve the approximation model efficiently. We also found in our numerical experiments that the proposed algorithm can be significantly accelerated by the FISTA technique in [1]. Although convergence analysis of the developed algorithm for the considered case had been given in [24], from a mathematical viewpoint there is a need for

understanding convergence mechanisms for more general problems involving the ℓ_0 norm. Moreover, although FISTA has been well studied in the context of convex optimization, its acceleration effects for nonconvex optimization problems involving the ℓ_0 norm should be justified.

In this paper, we extend the study in [24] to a general model covering both constrained and unconstrained cases with an arbitrary matrix A. The model under consideration has the form

$$\min\{\|Dx\|_0 + f(\mathscr{A}x) : x \in \mathbb{R}^n\}, \tag{2.4}$$

where $\mathscr{A}$ is the affine transform with the form $\mathscr{A}x := Ax - u$, and f is a proper and lower semi-continuous convex function. Specially, by identifying f as the indicator function of $\mathscr{B}(\|\cdot\|_2, \epsilon)$ and $\frac{\lambda}{2}\|\cdot\|_2^2$, model (2.5) can be specified to models (2.2) and (2.3), respectively, where

$$\mathscr{B}(\|\cdot\|_2, \epsilon) := \{z : \|z\|_2 \le \epsilon\}$$

is the ball of radius ϵ with respect to the ℓ_2 norm. We approximate the ℓ_0 norm in (2.4) with its Moreau envelope $\mathrm{env}_{\beta\|\cdot\|_0}$, and result in the model

$$\min\{\mathrm{env}_{\beta\|\cdot\|_0}(Dx) + f(\mathscr{A}x) : x \in \mathbb{R}^n\}. \tag{2.5}$$

We shall present an equivalent formulation of problem (2.5) which possesses a variable separation structure. We shall develop a two-stage proximity algorithm for solving problem (2.5) and understand the insight of its convergence to (local) minimizers. Particularly, we shall reveal that the speedup of FISTA for the proposed algorithm is due to the fact that the second stage of the algorithm is essentially solving a convex subproblem.

This paper is organized in five sections. In section "Minimizers of the Proposed Model", we propose our approximate sparsity model using the envelope of the ℓ_0 norm. We then present an equivalent formulation of the proposed model and characterize its global and local minimizers. In section "Convergence Analysis of the Proposed Algorithms", we describe a two stage fixed-point proximity algorithm for solving the proposed model and analyze its convergence. We also prove that FISTA can speed up the proposed algorithms for solving the nonconvex optimization problems involving the ℓ_0 norm. Because this paper mainly focuses on understanding the mathematical insight of convergence of the proposed algorithm, we shall only present a simple numerical example in section "Numerical Experiments" to show the convergence and acceleration effects of the algorithms. Those who are interested in knowing more about the numerical performance of the proposed algorithm are referred to [24], where a special case of the model studied in this paper was considered. Our conclusions are drawn in section "Conclusion".

Minimizers of the Proposed Model

The goal of this section is to conduct a theoretical study on the proposed model (2.5). Specifically, we present an equivalent formulation of model (2.5), characterize its local minimizers and prove the existence of its local minimizers.

We recall the notion of the proximity operator and the Moreau envelope crucial for our algorithmic development and as well as the related convergence analysis. Let $\varphi : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper, lower semi-continuous function, and β be a parameter. The proximity operator $\text{prox}_{\beta\varphi}$ at $x \in \mathbb{R}^d$ is defined by

$$\text{prox}_{\beta\varphi}(x) := \arg\min \left\{ \frac{1}{2}\|z - x\|_2^2 + \beta\varphi(z) : z \in \mathbb{R}^d \right\}.$$

Note that $\text{prox}_{\beta\varphi}$ is a set valued function. The proximity operator and the Moreau envelop are intimately related to each other. The Moreau envelope $\text{env}_{\beta\varphi}$ at $x \in \mathbb{R}^d$ is defined by

$$\text{env}_{\beta\varphi}(x) := \min \left\{ \frac{1}{2\beta}\|z - x\|_2^2 + \varphi(z) : z \in \mathbb{R}^d \right\}.$$

As shown in [22], for any $\beta > 0$, the function $\text{env}_{\beta\varphi}$ enjoys several remarkable properties: It is a continuous finite-valued function whereas φ itself may merely be lower semi-continuous and extended real-valued. Further, it yields a family of approximations $\{\text{env}_{\beta\varphi}\}_{\beta>0}$ to the function φ and there holds

$$\lim_{\beta \to 0^+} \text{env}_{\beta\varphi}(x) = \varphi(x), \quad x \in \mathbb{R}^d.$$

These properties make $\text{env}_{\beta\|\cdot\|_0}$ to be seen as a nonconvex but continuous relaxation of the ℓ_0 norm and they motivate us to approximate the ℓ_0 norm with its Moreau envelope.

We introduce a function of two variables

$$F(y, x) := \frac{1}{2\beta}\|y - Dx\|_2^2 + \|y\|_0 + f(\mathscr{A}x), \quad (y, x) \in \mathbb{R}^q \times \mathbb{R}^n$$

and consider the model

$$\min\{F(y, x) : (y, x) \in \mathbb{R}^q \times \mathbb{R}^n\}. \tag{2.6}$$

Moreover, for $x \in \mathbb{R}^n$, we let

$$J(x) := \text{env}_{\beta\|\cdot\|_0}(Dx) + f(\mathscr{A}x).$$

By the definition of the Moreau envelope, we observe that

$$J(x) = F(y, x), \quad \text{for all} \quad y \in \text{prox}_{\beta\|\cdot\|_0}(Dx) \ \text{ and for all } \ x \in \mathbb{R}^n. \tag{2.7}$$

The next proposition confirms that the proposed model (2.5) and model (2.6) are essentially equivalent.

Proposition 2.1 *Let $f : \mathbb{R}^m \to (-\infty, +\infty]$ be a proper and lower semicontinuous convex function, $\mathscr{A} : \mathbb{R}^n \to \mathbb{R}^m$ be an affine transform and $D \in \mathbb{R}^{q\times n}$ be a tight framelet system. For any $\beta > 0$, if a pair $(y^\star, x^\star)$ is a solution of model (2.6), then*

$$y^\star \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star). \tag{2.8}$$

Moreover, a pair $(y^\star, x^\star)$ is a solution of model (2.6) if and only if $x^\star$ is a solution of model (2.5) with $y^\star$ satisfying (2.8).

Proof We prove the first part of this proposition. Suppose that $(y^\star, x^\star)$ is a solution of model (2.6) and we prove that $(y^\star, x^\star)$ satisfies the inclusion relation (2.8). It follows immediately from the hypothesis that

$$F(y^\star, x^\star) \le F(y, x^\star), \quad \text{for every} \ \ y \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star).$$

On the other hand, by the definition of the proximity operator of $\|\cdot\|_0$, one has that

$$\frac{1}{2\beta}\|y^\star - Dx^\star\|_2^2 + \|y^\star\|_0 \ge \frac{1}{2\beta}\|y - Dx^\star\|_2^2 + \|y\|_0, \quad \text{for every} \ \ y \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star).$$

Hence, we have that

$$F(y^\star, x^\star) \ge F(y, x^\star), \quad \text{for every} \ \ y \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star).$$

Summarizing the above discussion, we conclude that

$$F(y^\star, x^\star) = F(y, x^\star), \quad \text{for every} \ \ y \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star).$$

This implies that

$$\frac{1}{2\beta}\|y^\star - Dx^\star\|_2^2 + \|y^\star\|_0 = \frac{1}{2\beta}\|y - Dx^\star\|_2^2 + \|y\|_0, \quad \text{for every} \ \ y \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star).$$

By using the definition of the proximity operator of $\|\cdot\|_0$ again, we observe that $(y^\star, x^\star)$ satisfies the inclusion relation (2.8).

Next, we prove the second part of the proposition. We assume that a pair $(y^\star, x^\star)$ is a solution of model (2.6). We show by contradiction that $x^\star$ is a solution of model (2.5). Assume to the contrary that there exists a vector $\widetilde{x} \in \mathbb{R}^n$ such that $J(\widetilde{x}) < J(x^\star)$. This inequality together with (2.7) yields that

$$F(\widetilde{y}, \widetilde{x}) = J(\widetilde{x}) < J(x^\star), \quad \text{for all} \ \ \widetilde{y} \in \text{prox}_{\beta\|\cdot\|_0}(D\widetilde{x}). \tag{2.9}$$

By the first part of this proposition, we know that $y^\star \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star)$. By (2.7), we obtain that $J(x^\star) = F(y^\star, x^\star)$. This with (2.9) gives that $F(\widetilde{y}, \widetilde{x}) < F(y^\star, x^\star)$, which contradicts the assumption and proves that $x^\star$ is a solution of model (2.5).

Conversely, we assume that $x^\star$ is a solution of model (2.5) and $y^\star \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star)$. We show that the pair $(y^\star, x^\star)$ is a solution of model (2.6). Once again, we prove it by contradiction. If $(y^\star, x^\star)$ is not a solution of model (2.6), then there exists a vector $\widetilde{x}$ satisfying $F(\widetilde{y}, \widetilde{x}) < F(y^\star, x^\star)$, where $\widetilde{y}$ is any vector in $\text{prox}_{\beta\|\cdot\|_0}(D\widetilde{x})$. This together with (2.7) implies that $J(\widetilde{x}) < J(x^\star)$, which violates the assumption of $x^\star$ being a solution of model (2.5).

The next proposition identifies a global minimizer of $F(\cdot, \cdot)$. This result will be used in the next section to design numerical algorithms.

Proposition 2.2 *Let $f : \mathbb{R}^m \to (-\infty, +\infty]$ be a proper and lower semicontinuous convex function, $\mathscr{A} : \mathbb{R}^n \to \mathbb{R}^m$ be a affine transform, $D \in \mathbb{R}^{q\times n}$ be a tight framelet system, and $\beta > 0$. If a pair $(y^\star, x^\star)$ is a solution of model (2.6), then*

$$y^\star \in \text{prox}_{\beta\|\cdot\|_0}(Dx^\star) \quad \text{and} \quad x^\star = \text{prox}_{\beta f\circ\mathscr{A}}(D^\top y^\star). \tag{2.10}$$

Proof Let $(y^\star, x^\star)$ be a solution of model (2.6). By Proposition 2.1, immediately we have the first inclusion of (2.10). It remains to prove the second formula of (2.10). By the Fermat rule (see, e.g., Theorem 10.1 in [22]), we have that

$$0 \in \frac{1}{\beta} D^\top (Dx^\star - y^\star) + \partial(f \circ \mathscr{A})(x^\star), \tag{2.11}$$

where ∂ denotes the Fréchet subdifferential (see, e.g. [22]). Since $D^\top D = I$ and

$$\beta\partial(f \circ \mathscr{A}) = \partial(\beta f \circ \mathscr{A}),$$

the inclusion in (2.11) leads to

$$D^\top y^\star - x^\star \in \partial(\beta f \circ \mathscr{A})(x^\star). \tag{2.12}$$

This inclusion combined with Proposition 2.6 in [18] leads to the second formula of (2.10).

We need the notion of the support of a vector. For a vector $y \in \mathbb{R}^q$, we denote by $N(y)$ the support of all nonzero components of y, that is $N(y) := \{i : y_i \neq 0\}$. For an ordered subset $\mathscr{N}$ of the ordered set $\{1, 2, \ldots, q\}$, we use $\sharp\mathscr{N}$ to denote its cardinality and $\mathscr{N}[i]$ its ith component. For a given subset $\mathscr{N}$, we let $\mathscr{S}_\mathscr{N}$ denote the set of vectors whose supports are included in $\mathscr{N}$, that is,

$$\mathscr{S}_\mathscr{N} := \{y : N(y) \subseteq \mathscr{N}\}. \tag{2.13}$$

When no ambiguity may be caused, we write $\mathscr{S}_{\mathscr{N}}$ as $\mathscr{S}$. Clearly, $\mathscr{S}$ is a convex set.

We next construct a convex optimization problem whose global minimizer is a (local) minimizer of the nonconvex optimization problem (2.6). For this purpose, we introduce a convex function

$$G(y,x) := \frac{1}{2\beta}\|y - Dx\|_2^2 + f(\mathscr{A}x), \quad (y,x) \in \mathbb{R}^q \times \mathbb{R}^n.$$

For an ordered subset $\mathscr{N}$ of $\{1,2,\ldots,q\}$, we consider the optimization problem

$$\min\{G(y,x) : y \in \mathscr{S}, x \in \mathbb{R}^n\}. \tag{2.14}$$

Clearly, since $G(\cdot,\cdot)$ is a convex function and the set $\mathscr{S} \times \mathbb{R}^n$ is a convex set, the model (2.14) is a convex optimization problem.

Proposition 2.3 *Let $f : \mathbb{R}^m \to (-\infty, +\infty]$ be a proper, lower semicontinuous convex function and continuous on its domain, $\mathscr{A} : \mathbb{R}^n \to \mathbb{R}^m$ be a affine transform, $D \in \mathbb{R}^{q\times n}$ be a tight framelet system, and $\mathscr{N}$ be an ordered subset of $\{1,2,\ldots,q\}$. If $(y^\star, x^\star) \in \mathscr{S} \times \mathbb{R}^n$ is a solution of (2.14), then $(y^\star, x^\star)$ is a local minimizer of $F(\cdot,\cdot)$ in (2.6).*

Proof To prove that $(y^\star, x^\star)$ is a local minimizer of $F(\cdot,\cdot)$, we consider all pairs $(y^\star + \triangle y, x^\star + \triangle x)$ in a small neighbourhood of $(y^\star, x^\star)$, where $\triangle y \in \mathscr{S}$ and $\triangle x \in \mathbb{R}^n$ are small in their Euclidean norms. We consider only the case of $\mathscr{A}(x^\star + \triangle x) \in \mathrm{dom}(f) := \{x \in \mathbb{R}^m : f(x) < +\infty\}$, because if this is not the case, then $f(\mathscr{A}(x^\star + \triangle x)) = +\infty$. In such a case, we clearly have that

$$F(y^\star, x^\star) \le F(y^\star + \triangle y, x^\star + \triangle x).$$

Let $\sigma_1 := \min\{|y_i^\star| : i \in N(y^\star)\}$. When $\|\triangle y\|_\infty < \sigma_1$, we have that

$$|y_i^\star + (\triangle y)_i|_0 = |y_i^\star|_0, \quad \text{for all } i \in N(y^\star). \tag{2.15}$$

We consider two cases: $\triangle y \in \mathscr{S}$ and $\triangle y \notin \mathscr{S}$.

We first consider the case that $\triangle y \in \mathscr{S}$. When $\|\triangle y\|_\infty < \sigma_1$, by (2.15), we have that $y^\star + \triangle y \in \mathscr{S}$ and $\|y^\star + \triangle y\|_0 = \|y^\star\|_0$. Therefore, $(y^\star + \triangle y, x^\star + \triangle x)$ is a feasible point of (2.14). Since $(y^\star, x^\star)$ is a solution of (2.14), we find that

$$G(y^\star, x^\star) \le G(y^\star + \triangle y, x^\star + \triangle x).$$

This implies immediately that

$$G(y^\star, x^\star) + \|y^\star\|_0 \le G(y^\star + \triangle y, x^\star + \triangle x) + \|y^\star + \triangle y\|_0.$$

By the definition of F, if $\|\triangle y\|_\infty < \sigma_1$, we have that

$$F(y^\star, x^\star) \leq F(y^\star + \triangle y, x^\star + \triangle x). \tag{2.16}$$

We now consider the case that $\triangle y \notin \mathscr{S}$. When $\|\triangle y\|_\infty < \sigma_1$, we have that

$$\|y^\star + \triangle y\|_0 \geq \|y^\star\|_0 + 1. \tag{2.17}$$

Moreover, since f is continuous on its domain and $\mathscr{A}(x^\star + \triangle y) \in \mathrm{dom}(f)$, we get that

$$\lim_{\triangle x, \triangle y \to 0} G(y^\star + \triangle y, x^\star + \triangle x) = G(y^\star, x^\star).$$

Therefore, there exists a positive number σ_2 such that whenever $\|\triangle y\|_\infty < \sigma_2$ and $\|\triangle x\|_\infty < \sigma_2$, there holds the inequality

$$G(y^\star + \triangle y, x^\star + \triangle x) \geq G(y^\star, x^\star) - 1. \tag{2.18}$$

Let $\sigma := \min\{\sigma_1, \sigma_2\}$. Combining (2.17) and (2.18), for any small perturbations $(\triangle y, \triangle x)$ such that $\|\triangle y\|_\infty < \sigma$ and $\|\triangle x\|_\infty < \sigma$, we obtain that

$$G(y^\star + \triangle y, x^\star + \triangle x) + \|y^\star + \triangle y\|_0 \geq G(y^\star, x^\star) + \|y^\star\|_0.$$

This implies that

$$F(y^\star + \triangle y, x^\star + \triangle x) \geq F(y^\star, x^\star). \tag{2.19}$$

Combining both cases (2.16) and (2.19), we conclude that $(y^\star, x^\star)$ is a local minimizer of $F(y, x)$.

We next show the existence of a local minimizer of model (2.6). To this end, we define necessary notation. By $D_{\mathscr{N}}$ we denote the submatrix of D associating with the set $\mathscr{N}$, that is,

$$D_{\mathscr{N}} := [D^\top_{\mathscr{N}[1]}, D^\top_{\mathscr{N}[2]}, \ldots, D^\top_{\mathscr{N}[\sharp\mathscr{N}]}]^\top.$$

Let $\overline{\mathscr{N}}$ be the complement of $\mathscr{N}$ in $\{1, 2, \ldots, q\}$. For a convex set $\mathscr{S}$, let $\mathrm{P}_{\mathscr{S}}$ denote the projection on $\mathscr{S}$. For a linear mapping T, let $\mathrm{Ker}(T)$ denote the null space of T.

Proposition 2.4 *For any subset $\mathscr{N}$ of $\{1, 2, \ldots, q\}$, if $\mathrm{Ker}(D_{\overline{\mathscr{N}}}) \cap \mathrm{Ker}(\mathscr{A}) = \{0\}$, then $F(\cdot, \cdot)$ in (2.6) has a (local) minimizer $(y^\star, x^\star)$ such that $N(y^\star) \subseteq \mathscr{N}$.*

Proof Since $\mathrm{Ker}(D_{\overline{\mathscr{N}}}) \cap \mathrm{Ker}(\mathscr{A}) = \{0\}$, we have that

$$\lim_{\|x\|_2 \to \infty} \|D_{\overline{\mathscr{N}}} x\|_2^2 + f(\mathscr{A}x) = +\infty.$$

That is, the objective function of the following model

$$\min\left\{\frac{1}{2}\|D_{\overline{\mathscr{N}}}x\|_2^2 + f(\mathscr{A}x) : x \in \mathbb{R}^n\right\}, \tag{2.20}$$

is coercive. Hence, model (2.20) has a solution $x^\star$. By noticing the definition of the set $\mathscr{S}$ in model (2.14), the pair $(y^\star, x^\star)$ with $y^\star := \mathrm{P}_{\mathscr{S}}(Dx^\star)$ is a solution of model (2.14). By Proposition 2.3, $(y^\star, x^\star)$ is a local minimizer of $F(\cdot,\cdot)$ in (2.6).

Convergence Analysis of the Proposed Algorithms

In this section, we describe our iterative algorithm for finding a local minimizer of problem (2.6) and study its convergence property. In particular, we shall explain how FISTA accelerates the convergence of the proposed algorithm.

Proposition 2.3 ensures that a global minimizer of problem (2.14) with a particular $\mathscr{S}$ corresponds to a local minimizer of problem (2.6). In view of the definition of the set $\mathscr{S}$ in (2.13), a local minimizer of problem (2.6) depends on the support of the sparse variable y. This property naturally motivates a two-stage algorithm to find a local minimizer of problem (2.6): We first pursue a suitable support for the sparse variable y and then solve problem (2.14) with this support.

Sparse Support Pursuit

The proposed algorithm to pursue a candidate for the support of the sparse variable y is motivated by the system of fixed-point equations in (2.10) that characterize the global minimizers of the problem (2.6). Based on the fixed-point equations (2.10) with a similar modification to that in [24], the proposed algorithm is expressed as

$$\begin{cases} y^{k+1} \in \mathrm{prox}_{\alpha\beta\|\cdot\|_0}(\alpha Dx^k + (1-\alpha)y^k) \\ x^{k+1} = \mathrm{prox}_{\beta f\circ\mathscr{A}}(D^\top y^{k+1}) \end{cases} \tag{2.21}$$

where $\alpha \in (0, 1)$ is a parameter which balances the two terms Dx^k and y^k.

The updates of both variables y and x in (2.21) at each iteration can be efficiently implemented. The first subproblem in (2.21) can be explicitly solved by using the closed-form formula of the proximity operator of the ℓ_0 norm, which may be found in [24]. By the definition of the proximity operator, with the variable y being fixed, updating the variable x in the second subproblem in (2.21) should solve the following convex optimization problem

$$\min\left\{\frac{1}{2}\|x - D^\top y\|_2^2 + \beta f(\mathscr{A}x) : x \in \mathbb{R}^n\right\}. \tag{2.22}$$

Problem (2.22) can be seen as a generalization of the well-know ROF model [23] which could be solved by several efficient algorithms in [1, 4, 6, 13, 15, 18, 19] and the reference therein.

We are now ready to present the convergence analysis of algorithm (2.21). Specifically, we shall show that the support of the sparse variable y^k generated by algorithm (2.21) remains unchanged after a finite number of iterations. We need a sequence of lemmas to establish the main theorem.

We need a function that is closely related to both functions F and G to be used as a bridge. Specifically, we define $E : \mathbb{R}^q \to \mathbb{R}$ at $y \in \mathbb{R}^q$ by

$$E(y) := \frac{1}{2\beta}\|(I - DD^\top)y\|_2^2 + \mathrm{env}_{\beta f \circ \mathscr{A}}(D^\top y). \tag{2.23}$$

In the next lemma, we reexpress the objective function F given in (2.6) in terms of E.

Lemma 2.1 *Let $y \in \mathbb{R}^q$. If $x = \mathrm{prox}_{\beta f \circ \mathscr{A}}(D^\top y)$, then*

$$G(y, x) = E(y) \ \text{ and } \ F(y, x) = E(y) + \|y\|_0. \tag{2.24}$$

Proof We establish the first equation in (2.24). Note that D is a tight framelet matrix and

$$\|y - Dx\|_2^2 = \|x - D^\top y\|_2^2 + \|(I - DD^\top)y\|_2^2.$$

By the definition of G, we obtain that

$$G(y, x) = \frac{1}{2\beta}\|(I - DD^\top)y\|_2^2 + \frac{1}{2\beta}\|x - D^\top y\|_2^2 + f(\mathscr{A}x).$$

By using the fact $x = \mathrm{prox}_{\beta f \circ \mathscr{A}}(D^\top y)$ and the definition of the Moreau envelope, we know that

$$\frac{1}{2\beta}\|x - D^\top y\|_2^2 + f(\mathscr{A}x) = \mathrm{env}_{\beta f \circ \mathscr{A}}(D^\top y).$$

Hence,

$$G(y, x) = \frac{1}{2\beta}\|(I - DD^\top)y\|_2^2 + \mathrm{env}_{\beta f \circ \mathscr{A}}(D^\top y),$$

which together with the definition of E leads to the first equation in (2.24).

We now prove the second equation in (2.24). From (2.6), for any $y \in \mathbb{R}^q$ and $x \in \mathbb{R}^n$, we have that

$$F(y, x) = G(y, x) + \|y\|_0.$$

This together with the first equation of (2.24) gives the second equation.

The next lemma establishes the quadratic function majorization property of the function E, which is crucial for our convergence analysis.

Lemma 2.2 *If E is given by (2.23), then*

$$E(t) \le E(s) + \langle \nabla E(s), t - s \rangle + \frac{1}{2\beta}\|t - s\|_2^2$$

for all $s, t \in \mathbb{R}^q$.

Proof We first show that ∇E is Lipschitz continuous with Lipschitz constant $\frac{1}{\beta}$. It can be directly verified for any $s, t \in \mathbb{R}^q$ that

$$\begin{aligned} &\|\nabla E(s) - \nabla E(t)\|_2^2 \\ =&\left\| \frac{1}{\beta}(I - DD^\top)(s - t) + \frac{1}{\beta} D((I - \mathrm{prox}_{\beta f \circ \mathscr{A}})(D^\top s) \right. \\ &\left. -(I - \mathrm{prox}_{\beta f \circ \mathscr{A}})(D^\top t)) \right\|_2^2. \end{aligned}$$

Since $D^\top D = I$ and $D^\top (I - DD^\top) = 0$, from the above identity we get that

$$\begin{aligned} &\|\nabla E(s) - \nabla E(t)\|_2^2 \\ =&\frac{1}{\beta^2}\|(I - DD^\top)(s - t)\|_2^2 + \frac{1}{\beta^2}\|(I - \mathrm{prox}_{\beta f \circ \mathscr{A}})(D^\top s) \\ &- (I - \mathrm{prox}_{\beta f \circ \mathscr{A}})(D^\top t)\|_2^2. \end{aligned}$$

Using the nonexpansiveness of operator $I - \mathrm{prox}_{\beta f \circ \mathscr{A}}$, we obtain that

$$\begin{aligned} \|\nabla E(s) - \nabla E(t)\|_2^2 &\le \frac{1}{\beta^2}\|(I - DD^\top)(s - t)\|_2^2 + \frac{1}{\beta^2} \\ \|D^\top (s - t)\|_2^2 &= \frac{1}{\beta^2}\|s - t\|_2^2. \end{aligned}$$

That is, ∇E is Lipschitz continuous with Lipschitz constant $\frac{1}{\beta}$. The conclusion follows immediately from the well-known and fundamental property of a differentiable convex function with a Lipschitz continuous gradient and Lipschitz constant $\frac{1}{\beta}$.

We need Lemma 3 from [24].

Lemma 2.3 *If $\{y^k\}$ is the sequence generated by algorithm (2.21), then the following statements hold:*

(i) $|y_i^k| \ge \sqrt{2\alpha\beta}$ *for all* $i \in N(y^k)$, *and* $|y_i^k| = 0$ *for all* $i \in \overline{N(y^k)}$.
(ii) $\|y^{k+1} - y^k\|_2 \ge \sqrt{2\alpha\beta}$ *if* $N(y^k) \ne N(y^{k+1})$.

We are now ready to establish the theorem on convergence of algorithm (2.21).

Theorem 2.1 *Let $f : \mathbb{R}^m \to (-\infty, +\infty]$ be a proper and lower semicontinuous convex function, $\mathscr{A} : \mathbb{R}^n \to \mathbb{R}^m$ be a affine transform, $D \in \mathbb{R}^{q\times n}$ be a tight framelet system, α be a number in $(0, 1)$, and β be a positive number. If $\{(y^k, x^k)\}$ is a sequence generated by algorithm (2.21), then the following statements hold:*

(i) $F(y^{k+1}, x^{k+1}) \le F(y^k, x^k)$ *for all* $k \ge 0$ *and* $\lim_{k\to\infty} \|y^{k+1} - y^k\|_2 = \lim_{k\to\infty} \|x^{k+1} - x^k\|_2 = 0$.

(ii) There exists a $K > 0$ *such that* $N(y^k) = N(y^K)$ *for all* $k \ge K$.

Proof We first prove Item (i). Since

$$x^{k+1} = \text{prox}_{\beta f\circ\mathscr{A}}(D^\top y^{k+1}),$$

by Lemma 2.1, we have that

$$F(y^{k+1}, x^{k+1}) = E(y^{k+1}) + \|y^{k+1}\|_0.$$

Combining this equation with Lemma 2.2 yields that

$$F(y^{k+1}, x^{k+1}) \le E(y^k) + \langle \nabla E(y^k), y^{k+1} - y^k\rangle + \frac{1}{2\beta}\|y^{k+1} - y^k\|_2^2 + \|y^{k+1}\|_0. \tag{2.25}$$

Noting that

$$\nabla E(y^k) = \frac{1}{\beta}(y^k - D\text{prox}_{\beta f\circ\mathscr{A}}(D^\top y^k)) = \frac{1}{\beta}(y^k - Dx^k), \tag{2.26}$$

we rewrite (2.26) with a parameter $\alpha \in (0, 1)$ to obtain the equation

$$\alpha Dx^k + (1 - \alpha)y^k = y^k - \alpha\beta\nabla E(y^k).$$

By the first equation of (2.21) and using the definition of the proximity operator, we have that

$$y^{k+1} \in \text{argmin}\left\{\frac{1}{2\alpha\beta}\|y - (y^k - \alpha\beta\nabla E(y^k))\|_2^2 + \|y\|_0 : y \in \mathbb{R}^q\right\}. \tag{2.27}$$

Expanding the quadratic term $\frac{1}{2\alpha\beta}\|y - (y^k - \alpha\beta\nabla E(y^k))\|_2^2$ in (2.27) as

$$\frac{\alpha\beta}{2}\|\nabla E(y^k)\|_2^2 + \langle \nabla E(y^k), y - y^k\rangle + \frac{1}{2\alpha\beta}\|y - y^k\|_2^2$$

and replacing the constant $\frac{\alpha\beta}{2}\|\nabla E(y^k)\|_2^2$ by $E(y^k)$ (that will not alter the minimizer) yield

$$y^{k+1} \in \text{argmin}\left\{E(y^k) + \langle \nabla E(y^k), y - y^k\rangle + \frac{1}{2\alpha\beta}\|y - y^k\|_2^2 + \|y\|_0 : y \in \mathbb{R}^q\right\}.$$

It follows immediately from Lemma 2.1 that

$$E(y^k) + \langle \nabla E(y^k), y^{k+1} - y^k \rangle + \frac{1}{2\alpha\beta}\|y^{k+1} - y^k\|_2^2 + \|y^{k+1}\|_0 \leq F(y^k, x^k). \tag{2.28}$$

Since $\alpha \in (0, 1)$, estimate (2.28) together with (2.25) leads to the first part of Item (i). Furthermore, we conclude that the sequence $\{F(y^k, x^k)\}$ converges. Moreover, from (2.25) and (2.28), we have that

$$\|y^{k+1} - y^k\|_2^2 \leq \frac{2\alpha\beta}{1-\alpha}(F(y^k, x^k) - F(y^{k+1}, x^{k+1})).$$

Therefore,

$$\lim_{k\to\infty} \|y^{k+1} - y^k\|_2 = 0.$$

By the second line in (2.21) and using the nonexpansiveness of the proximity operator, we have that

$$\lim_{k\to\infty} \|x^{k+1} - x^k\|_2 = 0.$$

This proves Item (i).

The second part of Item (i) implies that there exists a number $K > 0$ such that

$$\|y^{k+1} - y^k\|_2 < \sqrt{2\alpha\beta}, \quad \text{for all } k \geq K.$$

By Item (ii) of Lemma 2.3, sets $N(y^k)$ for all $k \geq K$ are identical as a subset of $\{1, 2, \ldots, q\}$. The proof of Item (ii) is completed.

Theorem 2.1 together with Lemma 2.3 reveals an important property of the sequence $\{y^k\}$: The index set of the components with biggest absolute values remains unchanged after a finite number of iterations. The support size of this index set depends on the β. We denote by $\mathscr{N}$ the unchanged support of y^k after a finite number of iterations. Then, $\mathscr{N}$ labels the locations of the most important transform coefficients of the recovered signal. In this sense, it is a good support for the sparse variable y which will be used in model (2.14) to recover the signal.

Recovery on the Sparse Support

After the support of the variable y keeps unchanged after a finite iteration of algorithm (2.21), we replace the proximity operator $\text{prox}_{\alpha\beta\|\cdot\|_0}$ in the first subproblem in (2.21) by the projection on the set $\mathscr{S}$ associated to the unchanged support $\mathscr{N}$.

The resulting iteration has the form of

$$\begin{cases} y^{k+1} = \mathrm{P}_{\mathscr{S}}(\alpha D x^k + (1-\alpha) y^k), \\ x^{k+1} = \mathrm{prox}_{\beta f \circ \mathscr{A}}(D^\top y^{k+1}). \end{cases} \tag{2.29}$$

It is readily seen that the closed-form formulation of $\mathrm{P}_{\mathscr{S}}$ can be given by

$$\mathrm{P}_{\mathscr{S}}(z) = \begin{cases} z_i, & \text{if } i \in \mathscr{N} \\ 0, & \text{otherwise} \end{cases}. \tag{2.30}$$

Therefore, the computational complexity of (2.29) is comparable to that of (2.21).

We shall show that the sequence $\{(y^k, x^k)\}$ generated by (2.29) converges to a solution of problem (2.14), which, by Proposition 2.3, is a local minimizer of problem (2.6).

Theorem 2.2 *If $f : \mathbb{R}^m \to (-\infty, +\infty]$ is a proper and lower semicontinuous convex function, $\mathscr{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a affine transform, $D \in \mathbb{R}^{q \times n}$ a tight framelet system, α a number in $(0, 1)$, β a positive number, and $\mathscr{S}$ a set defined in (2.13), then the sequence $\{(y^k, x^k)\}$ generated by algorithm (2.29) converges to a solution $(y^\star, x^\star)$ of problem (2.14).*

Proof By the second equation of (2.29) and (2.26), we get that

$$\alpha D x^k + (1-\alpha) y^k = y^k - \alpha\beta \nabla E(y^k).$$

Therefore, the two equations of algorithm (2.29) can be combined as

$$y^{k+1} = \mathrm{P}_{\mathscr{S}}(y^k - \alpha\beta \nabla E(y^k)). \tag{2.31}$$

Since E has the Lipschitz continuous gradient with Lipschitz constant $\frac{1}{\beta}$ and $\alpha \in (0, 1)$, the projection on a set is essentially the proximity operator of the indicator function associated with the set, the above iterative scheme (2.31) is the well-known forward-backward splitting algorithm (see, e.g., [7]) of the optimization problem

$$\min\{E(y) + \iota_{\mathscr{S}}(y) : y \in \mathbb{R}^q\}, \tag{2.32}$$

where $\iota_{\mathscr{S}}$ is the indicator function of $\mathscr{S}$. Therefore, the sequence $\{y^k\}$ converges to a solution of $y^\star$ of problem (2.32). As a direct consequence, by the second subproblem of (2.29), we get that

$$\lim_{k\to\infty} x^k = x^\star \ \text{ with } \ x^\star = \mathrm{prox}_{\beta f \circ \mathscr{A}}(D^\top y^\star).$$

We next show by contradiction that $(y^\star, x^\star)$ is a solution of problem (2.14). If this is not true, suppose a pair $(\widehat{y}, \widehat{x})$ solves problem (2.14), that is $G(\widehat{y}, \widehat{x}) < G(y^\star, x^\star)$. By the Fermat rule and Proposition 2.6 in [18], we can conclude that

$$\widehat{x} = \mathrm{prox}_{\beta f \circ \mathscr{A}}(D^\top \widehat{y}).$$

This together with Lemma 2.1 implies that $G(\widehat{y}, \widehat{x}) = E(\widehat{y})$. Moreover, since

$$x^\star = \text{prox}_{\beta f \circ \mathscr{A}}(D^\top y^\star),$$

again by Lemma 2.1, we have that $G(y^\star, x^\star) = E(y^\star)$. Therefore, we conclude that $E(\widehat{y}) < E(y^\star)$, which violates the fact that $y^\star$ is a solution of problem (2.32).

Since the iteration of algorithm (2.29) can be combined as that in (2.31), and E has the Lipschitz continuous gradient with Lipschitz constant $\frac{1}{\beta}$ and $\alpha \in (0, 1)$, the iteration (2.31) can be accelerated by the well-known FISTA technique

$$\begin{cases} y^{k+1} = \text{P}_{\mathscr{S}}(\alpha D \text{prox}_{\beta f \circ \mathscr{A}}(D^\top \tilde{y}^k) + (1-\alpha)\tilde{y}^k) \\ t_{k+1} = \frac{\sqrt{1+4t_k^2}+1}{2} \\ \tilde{y}^{k+1} = y^{k+1} + \frac{t_k - 1}{t_{k+1}}(y^{k+1} - y^k) \end{cases}. \tag{2.33}$$

The next theorem demonstrates that algorithm (2.33) offers an optimal convergence rate of $\mathscr{O}(1/k^2)$.

Theorem 2.3 *Let $\mathscr{Y} \times \mathscr{X}$ denote the solution set of problem (2.14). If $\{y^k\}$ is generated by algorithm (2.33), and $\{x^k\}$ is the sequence with $x^k := \text{prox}_{\beta f \circ \mathscr{A}}(D^\top y^k)$, then for any $k \geq 1$,*

$$G(y^k, x^k) - G(y^\star, x^\star) \leq \frac{2\beta \|y^0 - y^\star\|_2^2}{(k+1)^2}, \quad \textit{for all } (y^\star, x^\star) \in \mathscr{Y} \times \mathscr{X}. \tag{2.34}$$

Proof Since E is differentiable and its gradient is Lipschitz continuous with Lipschitz constant $\frac{1}{\beta}$, by Theorem 4.4 in [1], we get that

$$E(y^k) - E(y^\star) \leq \frac{2\|y^0 - y^\star\|_2^2}{\beta(k+1)^2}, \quad \text{for all } y^\star \in \mathscr{Y}. \tag{2.35}$$

To complete the proof, we need only show that $G(y^k, x^k) = E(y^k)$ and $G(y^\star, x^\star) = E(y^\star)$. Since $(y^\star, x^\star)$ is a solution of problem (2.14), by the proof of Theorem 2.2, we get that

$$x^\star = \text{prox}_{\beta f \circ \mathscr{A}}(D^\top y^\star).$$

This relation together with (2.24) imply that $G(y^\star, x^\star) = E(y^\star)$. Likewise, one can conclude that $G(y^k, x^k) = E(y^k)$ by the relation $x^k = \text{prox}_{\beta f \circ \mathscr{A}}(D^\top y^k)$ and (2.24). This completes the proof.

This theorem justifies the numerical examples provided in [24], which demonstrate an optimal convergence rate of $\mathscr{O}(1/k^2)$.

Numerical Experiments

In this section, we present a numerical example to demonstrate the convergence and acceleration effects of the proposed algorithms. In the example, we consider the image inpainting problem in the wavelet domain which is described by model (2.2).

The wavelet inpainting problem is to restore an image from its incomplete and/or inaccurate wavelet coefficients [5] and the corresponding matrix A has the form of $A = PW$, where W represents a wavelet transform matrix and P is a selection matrix. For wavelet inpainting, both steps of algorithm (2.21) have the closed form solution which can be found in our previous work [24]. We use the tight framelet systems constructed by the discrete cosine transform to generate the matrix D and choose the image "Barbara" with the size of 256×256 as the testing image. In the experiment, our goal is to approximate the original image x from its 80% Haar wavelet coefficients through minimizing the optimization model (2.6) in which the parameter β is 10.

Even though we propose in section "Convergence Analysis of the Proposed Algorithms" to use the FISTA in algorithm (2.21) once the nonzero support of the variable y keeps unchanged, our numerical observation shows that using FISTA at the beginning of algorithm (2.21) can also accelerate the algorithm to find a stable nonzero support of y. Figure 2.1a plots the difference of the ℓ_0 norm between two successive variables y^k and y^{k+1} obtained from algorithm (2.21) (the dashed curve) and its FISTA version (the solid curve). We can see that both algorithms can find the stable nonzero support of the sparse variable and using FISTA is much faster than algorithm (2.21) to obtain the support. The values of the objective function of model (2.6) against the number of iterations of the algorithms are displayed in Fig. 2.1b. Again, we use the dashed curve and the solid curve to represent the values computed by Algorithm (2.29) and the FISTA version. respectively. We can see the dramatic acceleration effects of using FISTA.

Conclusion

This paper provides a rigorous mathematical understanding of the fixed-point proximity algorithm for solving non-convex optimization problems involved the ℓ_0 norm. Specifically, we propose a sparse recovery model using an approximation of the ℓ_0 norm of the transformed coefficients of the underlying signal in a redundant tight framelet system. We characterize the minimizers of the proposed model by the fixed-point inclusion expressed in terms of the proximity operator of the functions involved in its objective function. We further develop a two stage algorithm based on the characterization. We prove that the second stage of the proposed algorithm can be accelerated by using the FISTA technique to reach the optimal rate convergence of $\mathcal{O}(1/k^2)$. Our numerical example confirms the convergence and acceleration convergence of the proposed algorithms.

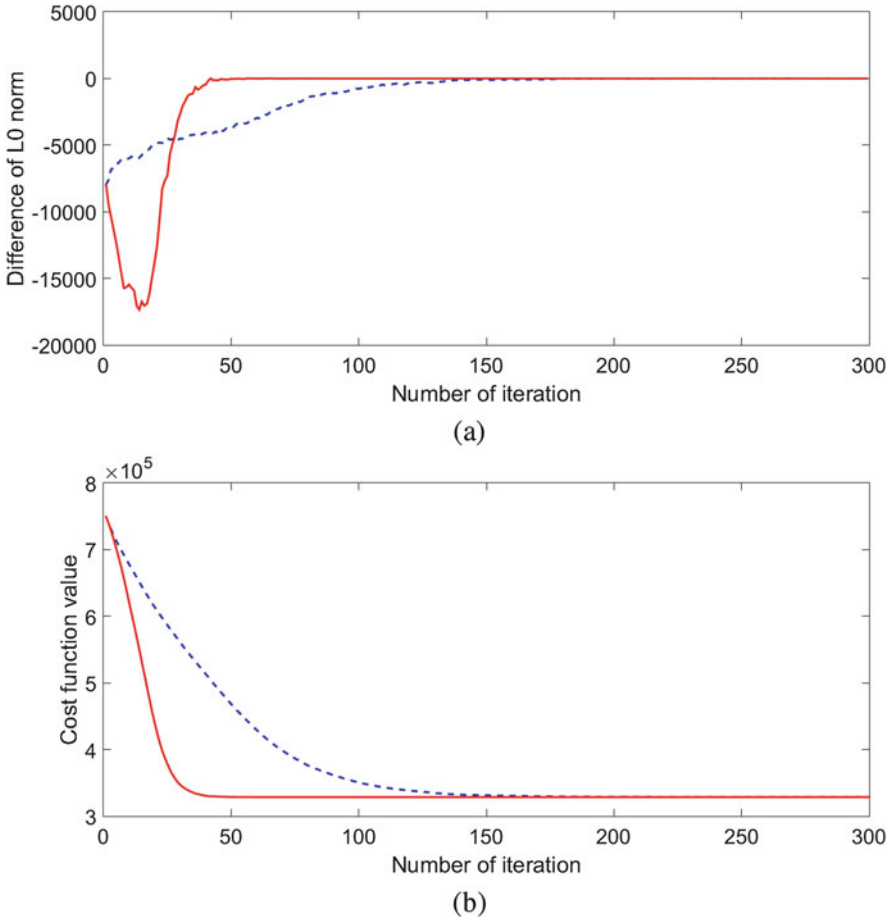


Fig. 2.1 The effects of using FISTA on the convergence and acceleration effects of the algorithms (blue dashed: Algorithm (2.21), red solid: using FISTA in (2.21))

Acknowledgements The author Xueying Zeng is supported by the Natural Science Foundation of China (No. 11701538, 11771408) and the Fundamental Research Funds for the Central Universities (No. 201562012). Both Lixin Shen and Yuesheng Xu were supported in part by the US National Science Foundation under Grant DMS-1522332. The author Yuesheng Xu is supported in part by the Special Project on High-performance Computing under the National Key R&D Program (No. 2016YFB0200602), and by the Natural Science Foundation of China under grants 11471013 and 11771464.

References

1. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. **2**(1), 183–202 (2009)
2. E.J. Candès, T. Tao, The power of convex relaxation: near-optimal matrix completion. IEEE Trans. Inf. Theory **56**(5), 2053–2080 (2010)
3. E. Candes, M. Wakin, S. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl. **14**, 877–905 (2008)
4. A. Chambolle, T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vision **40**(1), 120–145 (2011)
5. T.F. Chan, J. Shen, H.-M. Zhou, Total variation wavelet inpainting. J. Math. Imaging Vision **25**(1), 107–125 (2006)
6. F. Chen, L. Shen, Y. Xu, X. Zeng, The Moreau envelope approach for the L1/TV image denoising model. Inverse Prob. Imaging **8**(1), 53–77 (2014)
7. P.L. Combettes, V. Wajs, Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. **4**, 1168–1200 (2005)
8. I. Daubechies, R. DeVore, M. Fornasier, C. Guntuk, Iteratively reweighted least squares minimization for sparse recovery. Commun. Pure Appl. Math. **53**, 1–38 (2010)
9. D.L. Donoho, For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. Commun. Pure Appl. Math. **59**(6), 797–829 (2006)
10. J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**(456), 1348–1360 (2001)
11. M. Fazel, H. Hindi, S.P. Boyd, Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices, in *Proceedings of the 2003 American Control Conference, 2003*, vol. 3 (IEEE, Piscataway, 2003), pp. 2156–2162
12. S. Foucart, M.-J. Lai, Sparsest solutions of underdetermined linear systems via ℓ q-minimization for $0 < q \leq 1$. Appl. Comput. Harmon. Anal. **26**(3), 395–407 (2009)
13. T. Goldstein, S. Osher, The split bregman method for l1-regularized problems. SIAM J. Imaging Sci. **2**(2), 323–343 (2009)
14. M.-J. Lai, Y. Xu, W. Yin, Improved iteratively reweighted least squares for unconstrained smoothed \ell_q minimization. SIAM J. Numer. Anal. **51**(2), 927–957 (2013)
15. Q. Li, L. Shen, Y. Xu, N. Zhang, Multi-step fixed-point proximity algorithms for solving a class of optimization problems arising from image processing. Adv. Comput. Math. **41**(2), 387–422 (2015)
16. M. Malek-Mohammadi, M. Babaie-Zadeh, A. Amini, C. Jutten, Recovery of low-rank matrices under affine constraints via a smoothed rank function. IEEE Trans. Signal Process. **62**(4), 981–992 (2014)
17. O. Mangasarian, Minimum-support solutions of polyhedral concave programs*. Optimization **45**(1–4), 149–162 (1999)
18. C.A. Micchelli, L. Shen, Y. Xu, Proximity algorithms for image models: denoising. Inverse Prob. **27**(045009), 30 pp. (2011)
19. C.A. Micchelli, L. Shen, Y. Xu, X. Zeng, Proximity algorithms for the l_1/tv image denoising model. Adv. Comput. Math. **38**(2), 401–426 (2013)
20. M. Nikolova, Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares. Multiscale Model. Simul. **4**(3), 960–991 (2005)
21. B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
22. R.T. Rockafellar, R.J.-B. Wets, *Variational Analysis*. Grundlehren der mathematischen Wissenschaften, vol. 317 (Springer, New York, 1998)
23. L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)

24. L. Shen, Y. Xu, X. Zeng, Wavelet inpainting with the ℓ_0 sparse regularization. Appl. Comput. Harmon. Anal. **41**(1), 26–53 (2016)
25. H. Zou, The adaptive lasso and its oracle properties. J. Am. Stat. Assoc. **101**(476), 1418–1429 (2006)

Chapter 3
Sparse-Data Based 3D Surface Reconstruction for Cartoon and Map

Bin Wu, Talal Rahman, and Xue-Cheng Tai

Abstract A model combining the first-order and the second-order variational regularizations for the purpose of 3D surface reconstruction based on 2D sparse data is proposed. The model includes a hybrid fidelity constraint which allows the initial conditions to be switched flexibly between vectors and elevations. A numerical algorithm based on the augmented Lagrangian method is also proposed. The numerical experiments are presented, showing its excellent performance both in designing cartoon characters, as well as in recovering oriented three dimensional maps from contours or points with elevation information.

Introduction

Image processing has a strong influence and impact on our world, finding applications in almost all areas from nanophysics to astrophysics, from biology to social sciences, from robotics to smart phone applications, etc. 3D surface reconstruction from sparse data is both a challenging and an interesting image processing task.

One area of application of the surface reconstruction has been the sketch based 3D design, which has attracted much attention, cf. [1–5], because it is intuitive and effective, particularly in applications like cartoon and game design. To a sketch based method, the only known informations are information given on sparse lines, for instance in the form of contours [2], without specifying the heights, or in the form of complex sketches with elevation [3], or structured annotations [6]. However, the methods proposed in those papers are limited in their capabilities in

B. Wu (✉) · T. Rahman
Department of Computing, Mathematics and Physics, Western Norway University of Applied Sciences, Bergen, Norway
e-mail: Bin.Wu@hvl.no; Talal.Rahmann@hvl.no

X.-C. Tai
Department of Mathematics, University of Bergen, Bergen, Norway
e-mail: Xue-Cheng.Tai@uib.no

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_3

reconstructing structures with crease. The crease can be added artificially [7]. However, a simple and automatic method is still necessary when the task becomes large, complex and computationally intensive. Recently, to this end, there is a method been proposed [8, 9] which interpolates the normal vectors under curl-free constraint and then reconstructs the 3D surface based on the obtained vector field. The method [8, 9] is based on the previous work on surface reconstruction from surface gradients [10–17] and inspired by TV-Stokes method [18–22] where actually the curl-free constraint comes from. The main difference of this method [8, 9] compared to the other two-step methods [15, 23] is that, instead of the Laplace operator, an TV regularizer is employed which is better in edge preserving. In addition, a more physical constraint, the curl-free constraint is introduced by the method. The numerical results show an excellent performance in preserving edges and crease structures.

Another area of application has been the 3D surface reconstruction based only on height values (contours) or both height values and vectors. The height values are needed because the reconstructed surfaces for such applications are expected to be as precise as possible to the ground truth, e.g., the digital elevation maps and data compression. One way is to use explicit parameterization of given contours with subsequent pointwise matching and interpolation [24–26]. For such models, the parametrization may be difficult and expensive to compute, and the loss of continuity of slope across contours is a challenge. Another way is to treat the expected surface as a function over the considered domain. A renowned model is the absolutely minimizing Lipschitz extension (AMLE) interpolation model, see [27, 28], based on the PDE theory. The AMLE has a drawback in interpolating slopes. To overcome, one can rely on high-order differential operators or regularizations [29–33]. The method addressed in [33] introduces a third order anisotropic regularization together with a way to find an auxiliary vector field. The method results in clear surfaces with anisotropic features.

It is however desirable to recover the 3D surface with enough precision at the same time to be able to adjust the shape of the reconstructed surface by tuning vectors. For instance in case of data compression, it may be helpful to store vectors (relative positions) along with sparse elevations instead of single the sparse elevations for correct representations. The aim of this paper is to propose a versatile model incorporating both height and vector information in one place. We thus propose a one-step model with a combination of first-order and second-order variational regularizations under a hybrid fidelity constraint consisting of both elevation and normal vectors. The main contributions of our research can be summarized as follows:

- The model allows for adjusting normal vectors intuitively and a more precise representation of the elevations. It preserves both structures and details.
- A fast and efficient numerical algorithm based on the augmented Lagrangian method [34–36] is proposed which can be used for 3D surface reconstruction of cartoon and digital map based on very sparse 2D input data.

The paper is organized as follows. In section "Proposed Model", we propose our model with a first-order and a second-order regularizations and a hybrid constraint.

In section "Augmented Lagrangian Method", we present numerical method based on augmented Lagrangian. Numerical experiments on cartoon design and three dimensional map reconstructions are presented in section "Numerical Results". Finally, in section "Conclusion", we give our conclusion.

Proposed Model

We first explain the model presented in [8, 9] before we propose ours. We define the surface as the graph of I given by the points $(x, y, I(x, y)) \subset \mathbb{R}^3$ in the space, where I is a function of the coordinates x and y over a two dimensional domain $\Omega \subset \mathbb{R}^2$. The normal vector to the surface or the graph is then given by $(-\partial_x I, -\partial_y I, 1)$. Projecting it to the $xy-$plane, we get the 2D normal vectors as $(-\partial_x I, -\partial_y I)$. Because I is a scalar-valued function, the curl of the gradient of I must be zero. Based on this, a curl-free model has been proposed in [8, 9]. They first interpolated the normal vector $\mathbf{n} := (\partial_x I, \partial_y I)$ by solving the following constrained minimization problem

$$\min_{\mathbf{n}} \left\{ \int_{\Omega} (1-g)|\nabla \mathbf{n}|_F + g|\nabla \mathbf{n}|_F^2 + \eta \int_{\Gamma} |\mathbf{n} - \mathbf{n}^*| \right\}, \tag{3.1}$$

subject to the curl free condition

$$\nabla \times \mathbf{n} = 0,$$

where $\mathbf{n}^*$ is the known normal vector along some given sparse lines or strokes Γ, g is the parameter for a convex combination of the TV and the H^1 norm, and η is the parameter to balance between the regularization terms and the fidelity term. We note that $|\cdot|_F$ is used to denote the standard Frobenius norm [37]. The height map I is then reconstructed by solving the following minimization problem

$$\min_{I} \left\{ \int_{\Omega} (1-h)|\nabla I - \mathbf{n}| + h|\nabla I - \mathbf{n}|^2 + \xi \int_{\Sigma} |I - I_0| \right\}, \tag{3.2}$$

where $\mathbf{n}$ is the normal vector field obtained from the first minimization step, I_0 is the known elevation along some given sparse lines or strokes Σ, h is the parameter for a convex combination of TV and H^1 norms, ξ is the parameter to balance between the regularization terms and the fidelity term.

It is obvious that reconstructing a 3D surface would require both constraints, the one on the normal vector $\mathbf{n}$ and the one on the height I, corresponding to the fidelity terms of (3.1) and (3.2). However, since the model above is not coupled, it is hard to satisfy both constraints simultaneously, and therefore the resulting surface may deviate from the surface actually being sought.

We therefore propose the following one-step model including both the height and the normal vector constraint, that is the hybrid constraint.

$$\min_I \left\{ \int_\Omega g|\nabla(\nabla I)|_F + h|\nabla I| + \int_\Gamma \eta|\nabla I - \mathbf{n}^0| + \int_\Sigma \theta|I - I^0| \right\}, \tag{3.3}$$

where h and g are parameters for the first and the second variational regularizations, respectively. We note here that because our model is of second order, it naturally satisfies the curl free condition.

Augmented Lagrangian Method

For the numerical solution of the problem (3.4), we derive an augmented Lagrangian method, cf. [34]. Augmented Lagrangian method is preferred because it is in general fast and efficient; for its use in image processing, we refer to e.g. [35, 36].

In order to be able to define our entire minimization problem over the whole domain, we replace the two fidelity parameters η and θ with the following parameters,

$$\widehat{\eta} = \begin{cases} \eta, & \text{on } \Gamma \\ 0, & \text{in } \Omega\backslash\Gamma, \end{cases} \quad \text{and} \quad \widehat{\theta} = \begin{cases} \theta, & \text{on } \Sigma \\ 0, & \text{in } \Omega\backslash\Sigma. \end{cases}$$

We get our model (3.3) reformulated as follows,

$$\min_I \left\{ \int_\Omega g|\nabla(\nabla I)|_F + h|\nabla I| + \widehat{\eta}|\nabla I - \mathbf{n}^0| + \widehat{\theta}|I - I^0| \right\}. \tag{3.4}$$

We shall introduce some auxiliary variables and turn the above minimization problem into an equivalent constrained minimization problem. For the four L_1-norm terms in the above functional, introducing one new variable to each, we get four new variables $\mathbf{Q} := \nabla\mathbf{E}$, $\mathbf{P} := \nabla I$, $\mathbf{C} := \mathbf{P}$, and $S := I$, corresponding to $|\nabla(\nabla I)|_F$, $|\nabla I|$, $|\nabla I - \mathbf{n}^0|$, and $|I - I^0|$, respectively. In addition, for the term $|\nabla(\nabla I)|_F$, we introduce another variable $\mathbf{E} := \mathbf{P}$ in order to avoid dealing with high order terms. The unconstrained minimization problem (3.4) is then converted to an equivalent constrained optimization problem as:

$$\min_{\mathbf{Q},\mathbf{P},\mathbf{C},S} \left\{ \int_\Omega g|\mathbf{Q}|_F + h|\mathbf{P}| + \widehat{\eta}|\mathbf{C} - \mathbf{n}^0| + \widehat{\theta}|S - I^0| \right\}$$

such that

$$\mathbf{P} - \nabla I = 0; \quad \mathbf{E} - \mathbf{P} = 0; \quad \mathbf{Q} - \nabla\mathbf{E} = 0; \quad S - I = 0; \quad \text{and} \quad \mathbf{C} - \mathbf{P} = 0,$$

where $\mathbf{C}, \mathbf{E}, \mathbf{P} \in \mathbb{R}^2$ are 2-dimensional vectors, and $\mathbf{Q} \in \mathbb{R}^{2\times 2}$ is a 2-by-2 matrix. Using Lagrange multipliers and adding penalty terms for each condition, we get the following augmented Lagrangian functional

$$\begin{aligned}
&\mathscr{L}(\mathbf{Q}, \mathbf{P}, \mathbf{C}, S, I, \mathbf{E}; \Lambda_Q, \Lambda_P, \Lambda_C, \lambda_S, \Lambda_E) \\
&= \int_\Omega g|\mathbf{Q}|_F + h|\mathbf{P}| + \widehat{\eta}|\mathbf{C} - \mathbf{n}^0| + \widehat{\theta}|S - I^0| \\
&\quad + \Lambda_Q \cdot (\mathbf{Q} - \nabla\mathbf{E}) + \frac{c_Q}{2}|\mathbf{Q} - \nabla\mathbf{E}|_F^2 \\
&\quad + \Lambda_P \cdot (\mathbf{P} - \nabla I) + \frac{c_P}{2}|\mathbf{P} - \nabla I|^2 \\
&\quad + \Lambda_C \cdot (\mathbf{C} - \mathbf{P}) + \frac{c_C}{2}|\mathbf{C} - \mathbf{P}|^2 \\
&\quad + \lambda_S \cdot (S - I) + \frac{c_S}{2}|S - I|^2 \\
&\quad + \Lambda_E \cdot (\mathbf{E} - \mathbf{P}) + \frac{c_E}{2}|\mathbf{E} - \mathbf{P}|^2,
\end{aligned}$$

where Λ_Q, Λ_P, Λ_C, λ_S and Λ_E are Lagrange multipliers, c_Q, c_P, c_C, c_S and c_E are positive penalty parameters. That is, the augmented Lagrangian method is to seek a saddle point of the following problem:

$$\min_{\mathbf{Q},\mathbf{P},\mathbf{C},S,I,\mathbf{E}} \max_{\Lambda_Q,\Lambda_P,\Lambda_C,\lambda_S,\Lambda_E} \mathscr{L}(\mathbf{Q}, \mathbf{P}, \mathbf{C}, S, I, \mathbf{E}; \Lambda_Q, \Lambda_P, \Lambda_C, \lambda_S, \Lambda_E). \tag{3.5}$$

For the solution we solve its associated system of optimality conditions with an iterative procedure, see Algorithms 3.1 and 3.2. For the sake of convenience, we use $\Lambda := (\lambda_S, \Lambda_P, \Lambda_C, \Lambda_Q, \Lambda_E)$ to denote the Lagrange multipliers.

Algorithm 3.1 The augmented Lagrangian for (3.5)

Set $k = 0$ Initialize $\mathbf{Q}^0$, $\mathbf{P}^0$, $\mathbf{C}^0$, S^0, I^0, $\mathbf{E}^0$ and Λ^0 **while** *not converged* **do**

Set $k = k + 1$ Given Λ^{k-1}, solve the minimization problem:

$$(\mathbf{Q}^k, \mathbf{P}^k, \mathbf{C}^k, S^k, I^k, \mathbf{E}^k) = \arg\min_{\mathbf{Q},\mathbf{P},\mathbf{C},S,I,\mathbf{E}} \mathscr{L}(\mathbf{Q}, \mathbf{P}, \mathbf{C}, S, I, \mathbf{E}; \Lambda^{k-1}); \tag{3.6}$$

Update the Lagrange multipliers:

$$\begin{aligned}
&\lambda_S^k = \lambda_S^{k-1} + c_S(S^k - I^k); \quad && \Lambda_P^k = \Lambda_P^{k-1} + c_P(\mathbf{P}^k - \nabla I^k); \\
&\Lambda_C^k = \Lambda_C^{k-1} + c_C(\mathbf{C}^k - \mathbf{P}^k); && \Lambda_Q^k = \Lambda_Q^{k-1} + c_Q(\mathbf{Q}^k - \nabla(\mathbf{E}^k)); \\
&\Lambda_E^k = \Lambda_E^{k-1} + c_E(\mathbf{E}^k - \mathbf{P}^k);
\end{aligned}$$

end

Algorithm 3.2 Alternating minimization for (3.6)

Set $l = 0$ Initialize $\mathbf{Q}^{k,0} = \mathbf{Q}^{k-1}$; $\mathbf{P}^{k,0} = \mathbf{P}^{k-1}$; $\mathbf{C}^{k,0} = \mathbf{C}^{k-1}$; $S^{k,0} = S^{k-1}$; $I^{k,0} = I^{k-1}$; $\mathbf{E}^{k,0} = \mathbf{E}^{k-1}$;

while *not converged and* $l < L$ **do**

Solve the sub-minimization problems:

$$\mathbf{Q}^{k,l+1} = \arg\min_{\mathbf{Q}} \mathscr{L}(\mathbf{Q}, \mathbf{P}^{k,l}, \mathbf{C}^{k,l}, S^{k,l}, I^{k,l}, \mathbf{E}^{k,l}; \Lambda^{k-1});$$

$$\mathbf{P}^{k,l+1} = \arg\min_{\mathbf{P}} \mathscr{L}(\mathbf{Q}^{k,l+1}, \mathbf{P}, \mathbf{C}^{k,l}, S^{k,l}, I^{k,l}, \mathbf{E}^{k,l}; \Lambda^{k-1});$$

$$\mathbf{C}^{k,l+1} = \arg\min_{\mathbf{C}} \mathscr{L}(\mathbf{Q}^{k,l+1}, \mathbf{P}^{k,l+1}, \mathbf{C}, S^{k,l}, I^{k,l}, \mathbf{E}^{k,l}; \Lambda^{k-1});$$

$$S^{k,l+1} = \arg\min_{S} \mathscr{L}(\mathbf{Q}^{k,l+1}, \mathbf{P}^{k,l+1}, \mathbf{C}^{k,l+1}, S, I^{k,l}, \mathbf{E}^{k,l}; \Lambda^{k-1});$$

$$I^{k,l+1} = \arg\min_{I} \mathscr{L}(\mathbf{Q}^{k,l+1}, \mathbf{P}^{k,l+1}, \mathbf{C}^{k,l+1}, S^{k,l+1}, I, \mathbf{E}^{k,l}; \Lambda^{k-1});$$

$$\mathbf{E}^{k,l+1} = \arg\min_{\mathbf{E}} \mathscr{L}(\mathbf{Q}^{k,l+1}, \mathbf{P}^{k,l+1}, \mathbf{C}^{k,l+1}, S^{k,l+1}, I^{k,l+1}, \mathbf{E}; \Lambda^{k-1});$$

Set $l = l + 1$

end

Set $(\mathbf{Q}^k, \mathbf{P}^k, \mathbf{C}^k, S^k, I^k, \mathbf{E}^k) = (\mathbf{Q}^{k,L}, \mathbf{P}^{k,L}, \mathbf{C}^{k,L}, S^{k,L}, I^{k,L}, \mathbf{E}^{k,L})$.

Because the variables $\mathbf{Q}$, $\mathbf{P}$, $\mathbf{C}$, S, I and $\mathbf{E}$ in $\mathscr{L}(\mathbf{Q}, \mathbf{P}, \mathbf{C}, S, I, \mathbf{E}; \Lambda^{k-1})$ are coupled together in the minimization problem (3.6), it is difficult to solve them simultaneously. We split the minimization problem into six sub minimization problems, and solve them alternatively to convergence, see Algorithm 3.2.

The six sub-minimization problems can be formulated in a more specific and clearly way as in the following:

- The $\mathbf{Q}$-subproblem needs to solve:

$$\mathbf{Q}^\star = \arg\min_{\mathbf{Q}} \int_\Omega g|\mathbf{Q}|_F + \Lambda_Q \cdot \mathbf{Q} + \frac{c_Q}{2}|\mathbf{Q} - \nabla\mathbf{E}|_F^2. \tag{3.7}$$

- With $\tilde{\Lambda} := \Lambda_P - \Lambda_E - \Lambda_C.$, the $\mathbf{P}$-subproblem needs to solve:

$$\mathbf{P}^\star = \arg\min_{\mathbf{P}} \int_\Omega h|\mathbf{P}| + \tilde{\Lambda} \cdot \mathbf{P} + \frac{c_P}{2}|\mathbf{P} - \nabla I|^2 + \frac{c_E}{2}|\mathbf{E} - \mathbf{P}|^2 + \frac{c_C}{2}|\mathbf{C} - \mathbf{P}|^2, \tag{3.8}$$

- The $\mathbf{C}$-subproblem needs to solve:

$$\mathbf{C}^\star = \arg\min_{\mathbf{C}} \int_\Omega \Lambda_C \cdot \mathbf{C} + \frac{c_C}{2}|\mathbf{C} - \mathbf{P}|^2 + \widehat{\eta}|\mathbf{C} - \mathbf{n}^0|. \tag{3.9}$$

- The S-subproblem needs to solve:

$$S^\star = \arg\min_S \int_\Omega \lambda_S \cdot S + \frac{c_S}{2}|S - I|^2 + \widehat{\theta}|S - I^0|. \tag{3.10}$$

- The I-subproblem needs to solve:

$$I^\star = \arg\min_I \int_\Omega -\Lambda_P \cdot \nabla I + \frac{c_P}{2}|\mathbf{P} - \nabla I|^2 - \lambda_S \cdot I + \frac{c_S}{2}|S - I|^2. \tag{3.11}$$

- The $\mathbf{E}$-subproblem needs to solve:

$$\mathbf{E}^\star = \arg\min_{\mathbf{E}} \int_\Omega \Lambda_E \cdot \mathbf{E} + \frac{c_E}{2}|\mathbf{E} - \mathbf{P}|^2 - \Lambda_q \cdot \nabla \mathbf{E} + \frac{c_Q}{2}|\mathbf{Q} - \nabla \mathbf{E}|_F^2. \tag{3.12}$$

For the first four sub-minimization problems, we can find closed form solutions. Each problem has one L_1-norm term, and either one or more than one quadratic terms in its objective functional. Such problems can be solved using either a subgradient method, cf. [38], or a geometric method, cf. [36]. However, we will use a different approach to get the close form solutions in this work. We propose a simpler approach which is based on the optimality condition of the minimization functionals, i.e. the Euler-Lagrange equations. More details on this will be given below, see also Definition 3.1. For the last two sub-minimization problems, we solve them by the discrete cosine transform, see Remark 3.1.

Definition 3.1 If A and B are two matrices such that $A = \lambda B$ for some non-negative real number λ, then we say that A is compatible with B. It is easy to see that $A/|A|_F = B/|B|_F$.

In the following, we elaborate more on the details in getting close form solutions or design fast solvers for the subproblems.

Solving the Q-Subproblem (3.7)

The optimality condition, that is the Euler-Lagrange equation, for the Q-subproblem (3.7) is as follows

$$\frac{g}{c_Q}\frac{\mathbf{Q}^\star}{|\mathbf{Q}^\star|_F} + \mathbf{Q}^\star = \nabla \mathbf{E} - \frac{\Lambda_Q}{c_Q}.$$

Since g and c_Q are both positive numbers, the matrices $\mathbf{Q}^\star$ and $(\nabla \mathbf{E} - \frac{\Lambda_Q}{c_Q})$ are both compatible in the sense of Definition 3.1, according to which, we can replace

$\mathbf{Q}^\star/|\mathbf{Q}^\star|_F$ with $(\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q})/|\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q}|_F$. Now moving it to the right hand side, we get

$$\mathbf{Q}^\star = \left(1 - \frac{g}{c_Q|\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q}|_F}\right)\left(\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q}\right).$$

Again since $\mathbf{Q}^\star$ and $\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q}$ are compatible, the coefficient on the right hand side, that is $\left(1 - \frac{g}{c_Q|\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q}|_F}\right)$, must be non-negative, and hence

$$\mathbf{Q}^\star = \max\left\{0, 1 - \frac{g}{c_Q|\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q}|_F}\right\}\left(\nabla\mathbf{E} - \frac{\Lambda_Q}{c_Q}\right).$$

With the derivation given above, we have shown an easy way to get a close form solution for the subproblem. We shall use similar techniques to get close form solutions for some of the other subproblems.

Solving the P-Subproblem (3.8)

The corresponding Euler-Lagrange equation for the P-subproblem (3.8) is the following,

$$\frac{h}{c_P + c_E + c_C}\frac{\mathbf{P}^\star}{|\mathbf{P}^\star|} + \mathbf{P}^\star = \frac{c_P\nabla I + c_E\mathbf{E} + c_C\mathbf{C}}{c_P + c_E + c_C} - \frac{\tilde{\Lambda}}{c_P + c_E + c_C}.$$

Use $\mathbf{X}$ to denote $\frac{c_P\nabla I + c_E\mathbf{E} + c_C\mathbf{C}}{c_P + c_E + c_C} - \frac{\tilde{\Lambda}}{c_P + c_E + c_C}$. In the same way as before, since h, c_P, c_E and c_C are positive numbers, both vectors $\mathbf{P}^\star$ and $\mathbf{X}$ are compatible (cf. Definition 3.1). Accordingly, we replace $\mathbf{P}^\star/|\mathbf{P}^\star|$ with $\mathbf{X}/|\mathbf{X}|$, and move it to the right hand side, to get

$$\mathbf{P}^\star = \left(1 - \frac{h}{(c_P + c_E + c_C)|\mathbf{X}|}\right)\mathbf{X}.$$

Again since $\mathbf{P}^\star$ and $\mathbf{X}$ are compatible, the coefficient $\left(1 - \frac{h}{(c_P + c_E + c_C)|\mathbf{X}|}\right)$ must be non-negative. Hence

$$\mathbf{P}^\star = \max\left\{0, 1 - \frac{h}{(c_P + c_E + c_C)|\mathbf{X}|}\right\}\mathbf{X}.$$

Solving the C-Subproblem (3.9)

The corresponding Euler-Lagrange equation is the following,

$$\mathbf{C}^\star - \mathbf{n}^0 + \frac{\widehat{\eta}}{c_C}\frac{\mathbf{C}^\star - \mathbf{n}^0}{|\mathbf{C}^\star - \mathbf{n}^0|} = \mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}.$$

Since $\widehat{\eta}$ and c_C are both positive numbers, it follows that both vectors $\mathbf{C}^\star - \mathbf{n}^0$ and $\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}$ are compatible (cf. Definition 3.1). Accordingly, we replace $(\mathbf{C}^\star - \mathbf{n}^0)/|\mathbf{C}^\star - \mathbf{n}^0|$ with $(\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C})/|\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}|$, and move it to the right hand side, to obtain

$$\mathbf{C}^\star - \mathbf{n}^0 = \left(1 - \frac{\widehat{\eta}}{c_C|\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}|}\right)(\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}).$$

Again since $\mathbf{C}^\star - \mathbf{n}^0$ and $\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}$ are compatible, the coefficient $\left(1 - \frac{\widehat{\eta}}{c_C|\mathbf{P}-\mathbf{n}^0-\frac{\Lambda_C}{c_C}|}\right)$ must be non-negative. Hence

$$\mathbf{C}^\star = \mathbf{n}^0 + \max\left\{0, 1 - \frac{\widehat{\eta}}{c_C|\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}|}\right\}\left(\mathbf{P} - \mathbf{n}^0 - \frac{\Lambda_C}{c_C}\right).$$

Solving the S-Subproblem (3.10)

The Euler-Lagrange equation is the following,

$$S^\star - I^0 + \frac{\widehat{\theta}}{c_S}\frac{S^\star - I^0}{|S^\star - I^0|} = I - I^0 - \frac{\lambda_S}{c_S}.$$

Again using the fact that $\widehat{\theta}$ and c_S are both positive numbers, it follows that $S^\star - I^0$ and $I - I^0 - \frac{\lambda_S}{c_S}$ have the same sign. Replacing $(S^\star - I^0)/|S^\star - I^0|$ with $(I - I^0 - \frac{\lambda_S}{c_S})/|I - I^0 - \frac{\lambda_S}{c_S}|$, and moving it to the right hand side, we obtain

$$S^\star - I^0 = \left(1 - \frac{\widehat{\theta}}{c_S|I - I^0 - \frac{\lambda_S}{c_S}|}\right)\left(I - I^0 - \frac{\lambda_S}{c_S}\right).$$

Because $S^\star - I^0$ and $I - I^0 - \frac{\lambda_S}{c_S}$ have the same sign, the coefficient $\left(1 - \frac{\widehat{\theta}}{c_S|I-I^0-\frac{\lambda_S}{c_S}|}\right)$ must be non-negative. Therefore

$$S^\star = I^0 + \max\left\{0, 1 - \frac{\widehat{\theta}}{c_S|I - I^0 - \frac{\lambda_S}{c_S}|}\right\}\left(I - I^0 - \frac{\lambda_S}{c_S}\right).$$

Solving the I-Subproblem (3.11)

The Euler-Lagrange equation is the following inhomogeneous modified Helmholtz equation,

$$\nabla \cdot \nabla I^\star - \frac{c_S}{c_P} I^\star = \nabla \cdot \mathbf{P} + \frac{1}{c_P}\nabla \cdot \Lambda_P - \frac{c_S}{c_P} S - \frac{1}{c_P}\lambda_S,$$

with the Neumann boundary condition,

$$\nabla I^\star \cdot \nu = (\mathbf{P} + \frac{1}{c_P}\Lambda_P) \cdot \nu,$$

where the ν denotes the outward unit normal vector on the boundary of the domain. The Euler-Lagrange equation, with the boundary condition, is solved by the discrete cosine transform. Details are given in Remark 3.1.

Solving the E-Subproblem (3.12)

The corresponding Euler-Lagrange equation is the following,

$$\nabla \cdot \nabla \mathbf{E}^\star - \frac{c_E}{c_Q}\mathbf{E}^\star = \frac{1}{c_Q}\Lambda_E - \frac{c_E}{c_Q}\mathbf{P} + \frac{1}{c_Q}\nabla \cdot \Lambda_Q + \nabla \cdot \mathbf{Q},$$

which is a set of two inhomogeneous modified Helmholtz equations, one equation for each component of $\mathbf{E} = (E_1, E_2)$, and corresponding Neumann boundary conditions,

$$\nabla E_1 \cdot \nu = (\mathbf{Q}_1 + \frac{1}{c_Q}\Lambda_{Q1}) \cdot \nu,$$

$$\nabla E_2 \cdot \nu = (\mathbf{Q}_2 + \frac{1}{c_Q}\Lambda_{Q2}) \cdot \nu,$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are the row vectors of the matrix $\mathbf{Q}$, and Λ_{Q1} and Λ_{Q2} are corresponding Lagrange multipliers, respectively. ν is the outward unit normal on the boundary of the domain. Each equation is solved in the same way as in the I-subproblem, cf. Remark 3.1.

Remark 3.1 The last two sub-minimization problems, each reduces to solve a partial differential equation of the form

$$\triangle u(x, y) - \lambda u(x, y) = F(x, y),$$

with a Neumann boundary condition and λ a non-negative number, also known as the inhomogeneous modified Helmholtz equation. A fast solver based on discrete cosine transform similar for the Poisson equation, cf. [39, 40], is developed as the following treatment.

Using the singular value decomposition, the discrete Laplace operator

$$\begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}$$

takes the form of [40],

$$-C^\top \begin{bmatrix} 0 & \\ & \Sigma^2 \end{bmatrix} C,$$

where C is the $N \times N$ discrete cosine transform matrix and $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_{N-1})$ is the diagonal matrix with its diagonal entries representing the singular values $\sigma_k = 2 \sin \dfrac{\pi k}{2N}$ for $k = 1, 2, \cdots, N-1$. Now using this the discrete (matrix) formulation of the inhomogeneous modified Helmholtz equation then reads

$$-uC^\top \begin{bmatrix} 0 & \\ & \Sigma_x^2 \end{bmatrix} C - C^\top \begin{bmatrix} 0 & \\ & \Sigma_y^2 \end{bmatrix} Cu - \lambda u = F.$$

A further transformation using $\tilde{u} = CuC^\top$ and $\tilde{F} = CFC^\top$, results in

$$-\tilde{u} \begin{bmatrix} 0 & \\ & \Sigma_x^2 \end{bmatrix} - \begin{bmatrix} 0 & \\ & \Sigma_y^2 \end{bmatrix} \tilde{u} - \lambda \tilde{u} = \tilde{F}.$$

The solution of the above equation can be obtained by a direct entrywise division due to the linearity of the equation as well as the non-singularity of the coefficient

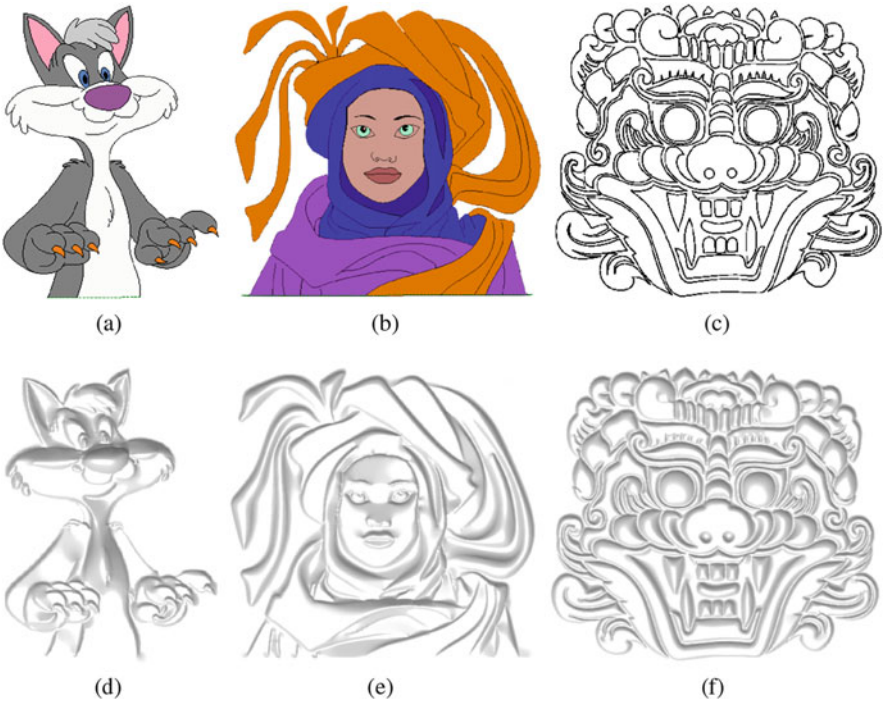


Fig. 3.1 Illustrating the cartoon case, where the input data are vectors along strokes (the top row) drawn by artists. The vectors are not shown here. The corresponding 3D cartoons generated by our algorithm, are shown in the bottom row. The parameters are $g = 0.5$, $h = 0$, $\theta = 0$, and $\eta = 5.0$

matrix, and is formulated as

$$\tilde{u} = \tilde{F}./M,$$

where ./ denotes the entrywise division and M is the $N \times N$ coefficient matrix defined as

$$M = -\begin{bmatrix} 0 & \sigma_{1,x}^2 & \cdots & \sigma_{N-2,x}^2 & \sigma_{N-1,x}^2 \\ \sigma_{1,y}^2 & \sigma_{1,x}^2 + \sigma_{1,y}^2 & \cdots & \sigma_{N-2,x}^2 + \sigma_{1,y}^2 & \sigma_{N-1,x}^2 + \sigma_{1,y}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{N-2,y}^2 & \sigma_{1,x}^2 + \sigma_{N-2,y}^2 & \cdots & \sigma_{N-2,x}^2 + \sigma_{N-2,y}^2 & \sigma_{N-1,x}^2 + \sigma_{N-2,y}^2 \\ \sigma_{N-1,y}^2 & \sigma_{1,x}^2 + \sigma_{N-1,y}^2 & \cdots & \sigma_{N-2,x}^2 + \sigma_{N-1,y}^2 & \sigma_{N-1,x}^2 + \sigma_{N-1,y}^2 \end{bmatrix} - \lambda,$$

u is defined on a squared $N \times N$ domain. The solution of the initial inhomogeneous modified Helmholtz equation is thus calculated as

$$u = C^\top((CFC^\top)./M)C$$

using discrete cosine transform.

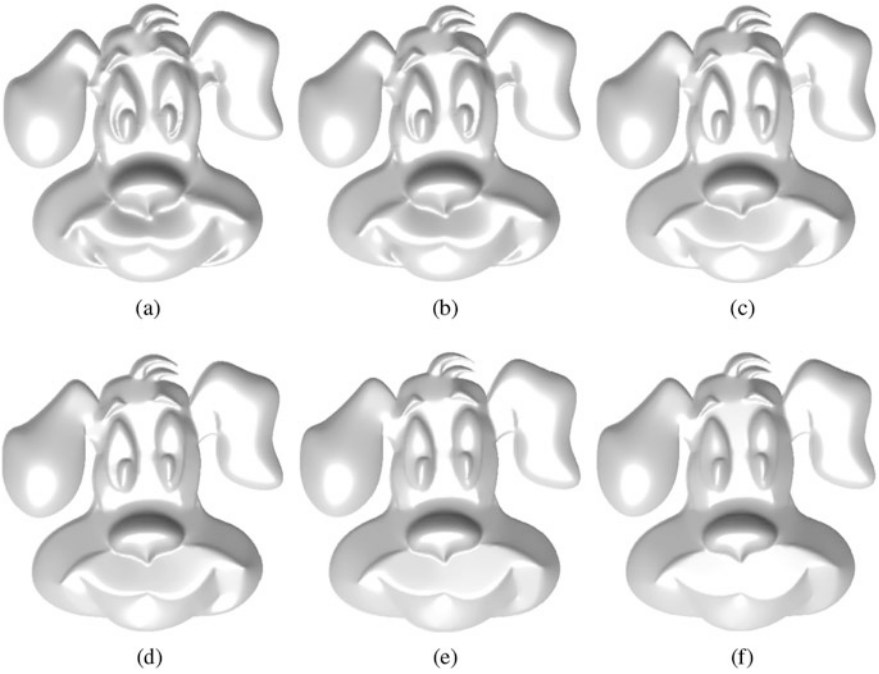


Fig. 3.2 Illustrating the effect of the second order regularization by varying the parameter g. In these tests, $h = 0$, $\theta = 0$, and $\eta = 5.0$. (**a**) g = 0.01, (**b**) g = 0.1, (**c**) g = 0.5, (**d**) g = 1, (**e**) g = 1.5, (**f**) g = 2

Numerical Results

For the numerical results, we consider the two different cases of surface reconstruction, namely, the 3D cartoon generation and the three dimensional map reconstruction, where in the first case we are given normal vectors along strokes while in the second case we are given both normal vectors and elevation data along contours and isolated points. The numerical tests are done using the augmented Lagrangian algorithm, Algorithms 3.1 and 3.2. Algorithm 3.1 is stopped when the total energy stabilized. For Algorithm 3.2, it was enough to use only one iteration.

In the cartoon case, we start with normal vectors along the strokes, which are given by artists. The results are shown in Fig. 3.1. Since we do not have the elevation data I_0, we set $\theta = 0$. In these experiments, we do not have flat surfaces, and hence we set $h = 0$. For flat surfaces h needs to be nonzero. We have used $g = 0.5$ and $\eta = 5.0$. As we can see from the Fig. 3.1, the algorithm is effective in preserving both structures and details.

In our next experiment with cartoon, we investigate the effect of the second order regularization by varying the g. The results are shown in Fig. 3.2, where the strokes and the normal vectors along the strokes are kept the same. θ is set equal to 0 in the

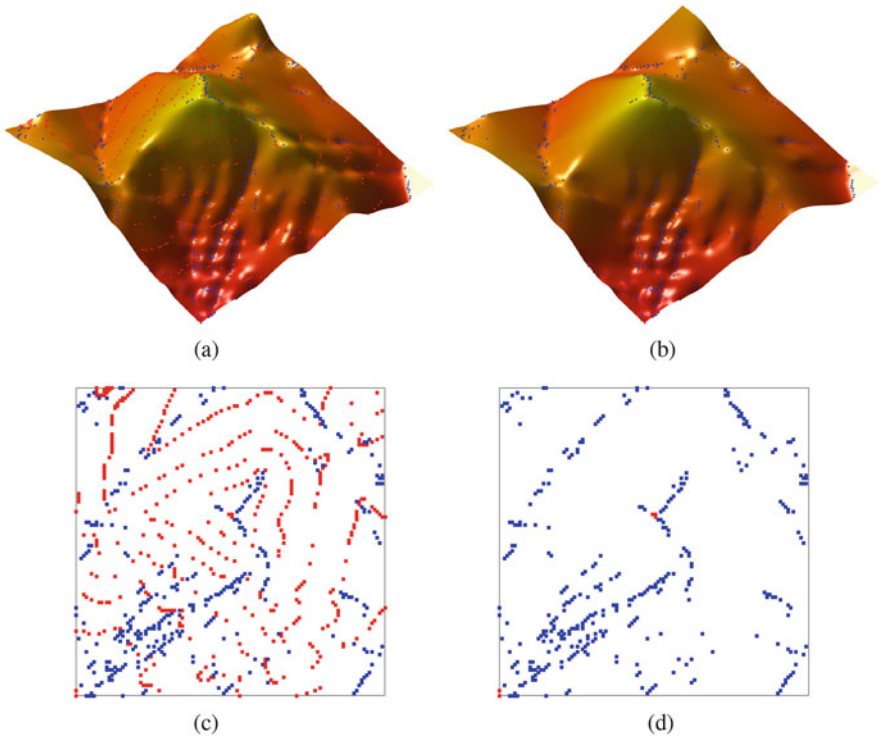


Fig. 3.3 3D surface reconstructions with two different sparsities of input data. Height values are given at red points, and normal vectors are given at blue points. (**a**) Reconstructed surface corresponding to sparsity of (**c**). (**b**) Reconstructed surface corresponding to sparsity of (**d**). (**c**) Input data sparsity: 5.18%. (**d**) Input data sparsity: 2.38%

experiment since the initial elevation is not known. h is set equal to 0 as we do not expect flat structures. The parameter g varies from 0.01 to 2.0 and the parameter η stays fixed at 5.0. As we can see in Fig. 3.2, the edges get sharper as g grows.

In our next experiment, we consider the 3D surface reconstruction of maps. The input data includes contours with height values, and isolated points with normal vectors. Figure 3.3 presents the results with two different sparsities of input data, respectively 5.18% and 2.38%. The given normal vectors are kept the same for both cases, and are represented by the blue points, as shown in Fig. 3.3c–d. The case in Fig. 3.3d has much less information on elevation than the case in Fig. 3.3c, represented by the red points. The parameters for both cases are $g = 0.1$, $h = 0$, $\theta = 10^5$ and $\eta = 10^6$. The results show that, if we have adequate vector information, even with less height data, our model preserves the main feature of the 3D maps perfectly.

In Fig. 3.4, we compare the effect of using vector constraint. Figure 3.4a shows the result of using the hybrid constraint while Fig. 3.4b shows the result using only the elevation data constraint. As we can see that without the vector constraint, in this

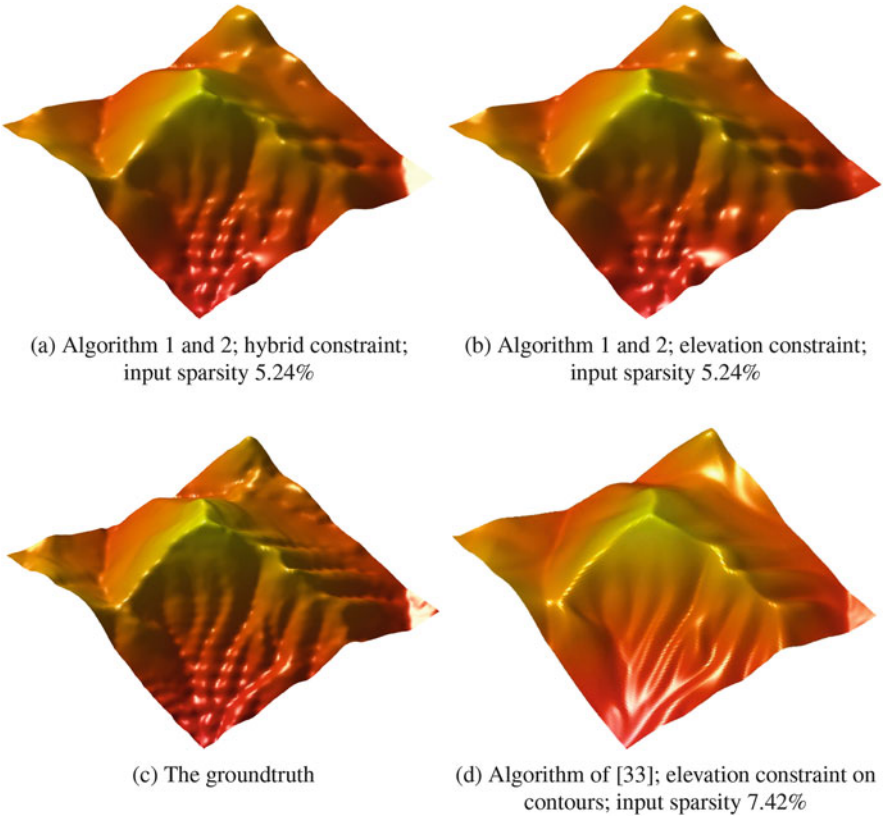


Fig. 3.4 Figures showing 3D reconstructions of map using different fidelity constraints, both elevation and vector constraint (hybrid) in (**a**), only elevation constraint in (**b**), using our algorithm, and elevation constraint on contours using the algorithm of [33] in (**d**)

test case, there is some loss of structure in the valley. The test cases in Fig. 3.4a, b have the same input points. The test case in Fig. 3.4b has only elevation data as input, while in the test case in Fig. 3.4a the elevation data is replaced with normal vectors for some points. Figure 3.4d the same reconstruction is made using the method [33] which is based on 3rd order anisotropic regularization. As we can see that our method manages to preserve the small structure comparatively better even with sparser data, because we have the flexibility to input additional information to our model like the normal vectors.

In the final experiment, cf. Fig. 3.5, the effect of the first order regularization is studied. As seen in the figure, the groundtruth contains a flat valley, cf. Fig. 3.5a, d. The parameters g, η and θ are kept the same in the whole experiment, whose values are $g = 0.1$, $\eta = 0$ and $\theta = 10^6$. In Fig. 3.5b, e, $h = 100$ while in Fig. 3.5c, f $h = 0$. As seen from the figure that ∇I term is needed to preserve the flat valley structure.

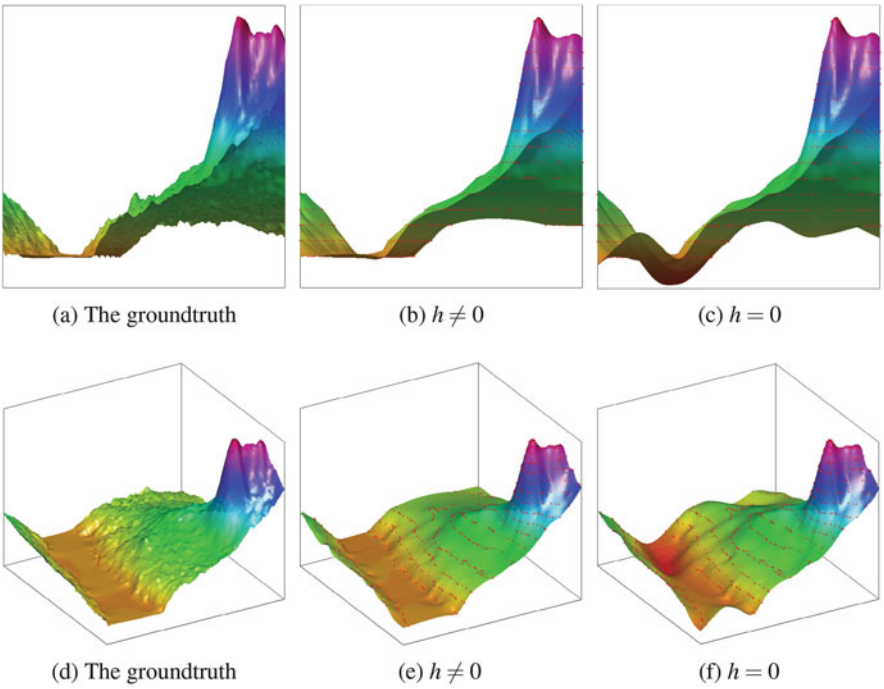


Fig. 3.5 Illustrating the effect of the first order regularization, 3D figures in the bottom row and 2D projection (yz-plane) in the top row. Inside the valley, we set the parameter h to 0 (switching off the first order regularization) in (**c**) and (**f**), and to 100 (switching it on) in (**b**) and (**e**). $h = 0$ in the rest of the domain. The sparsity of the input data is 5.98%. Here $g = 0.1$, $\eta = 0$, and $\theta = 10^6$ in all cases

Conclusion

We have proposed a model for 3D surface reconstruction based on 2D sparse hybrid data, that is involving both height values and normal vectors in the same model, allowing for flexible control of their fidelity. An effective algorithm based on the augmented Lagrangian has been developed, where we split the minimization problem into six sub minimization problems, each with either a closed form solution or a fast solver. The proposed model is well suited for both 3D cartoon design and digital 3D elevation maps. Because it allows for flexible use of both the height data and the vector information, which can be on sparse curves or points, it has the potential to be used in areas where precise reconstruction of surfaces, represented by rather sparse data, are needed, and rather quick, for instance in real time applications like the web-based 3D visualization of maps, 3D GPS navigation, and 3D online gaming.

Acknowledgements XC Tai acknowledges the support from Norwegian Research Council through ISP-Matematikk (Project no. 239033/F20). The authors also thank Dr. Jie Qiu for providing us example strokes.

References

1. R.C. Zeleznik, K.P. Herndon, J.F. Hughes, Sketch: an interface for sketching 3d scenes, in *The 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996), pp. 163–170
2. T. Igarashi, S. Matsuoka, H. Tanaka, Teddy: a sketching interface for 3d freeform design, in *The 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), pp. 409–416
3. O.A. Karpenko, J.F. Hughes, SmoothSketch: 3D free-form shapes from complex sketches, in *The 33th Annual Conference on Computer Graphics and Interactive Techniques* (2006), pp. 589–598
4. A. Nealen, T. Igarashi, O. Sorkine, M. Alexa, FiberMesh: designing freeform surfaces with 3D curves. ACM Trans. Graph. **26**(3), Article No. 41 (2007)
5. L. Olsen, F.F. Samavati, M.C. Sousa, J.A. Jorge, Sketch-based modeling: a survey. Comput. Graph. **33**(1), 88–103 (2009)
6. Y. Gingold, T. Igarashi, D. Zorin, Structured annotations for 2d-to-3d modeling. ACM Trans. Graph. **28**(5), Article No. 148 (2009)
7. L. Olsen, F.F. Samavati, M.C. Sousa, J.A. Jorge, Sketch-based mesh augmentation, in *The 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005)
8. J. Hahn, J. Qiu, E. Sugisaki, L. Jia, X.-C. Tai, H. Seah, Stroke-based surface reconstruction. CAM Report 12–18, UCLA (2012)
9. J. Hahn, J. Qiu, E. Sugisaki, L. Jia, X.-C. Tai, H. Seah, Stroke-based surface reconstruction. Numer. Math. Theory Meth. Appl. **6**(1), 297–324 (2013)
10. A. Agrawal, R. Raskar, R. Chellappa, What is the range of surface reconstructions from a gradient field?, in *Computer Vision C ECCV* (2006), pp. 578–591
11. R.T. Frankot, R. Chellappa, S. Member, A method for enforcing integrability in shape from shading algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **10**, 118–128 (1987)
12. N. Petrovic, I. Cohen, B.J. Frey, R. Koetter, T.S. Huang, Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1 (2001), p. 743
13. T. Simchony, R. Chellappa, M. Shao, Direct analytical methods for solving Poisson equations in computer vision problems. IEEE Trans. Pattern Anal. Mach. Intell. **12**, 435–446 (1990)
14. L. Zhang, G. Dugas-Phocion, J.-S. Samson, Single-view modeling of free-form scenes. J. Vis. Comput. Anim. **13**, 225–235 (2002)
15. T.-P. Wu, C.-K. Tang, M. Brown, H.-Y. Shum, Shapepalettes: interactive normal transfer visa sketching. ACM Trans. Graph. **26**(3), 07, Article No. 44
16. M. Prasad, A. Fitzgibbon, Single view reconstruction of curved surfaces, in *The 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2 (2006), pp. 1345–1354
17. H.-S. Ng, T.-P. Wu, C.-K. Tang, Surface-from-gradients without discrete integrability enforcement: a Gaussian kernel approach. IEEE Trans. Pattern Anal. Mach. Intell. **32**(11), 2085–2099 (2010)
18. T. Rahman, X.C. Tai, S. Osher, A TV-stokes denoising algorithm, in *Scale Space and Variational Methods in Computer Vision* (Springer, Berlin, 2007), pp. 473–483
19. C.A. Elo, A. Malyshev, T. Rahman, A dual formulation of the TV-stokes algorithm for image denoising, in *Scale Space and Variational Methods in Computer Vision* (Springer, Berlin, 2009), pp. 307–318

20. X.C. Tai, S. Borok, J. Hahn, Image denoising using TV-Stokes equation with an orientation-matching minimization, in *International Conference on Scale Space and Variational Methods in Computer Vision* (Springer, Berlin, 2009), pp. 490–501
21. J. Hahn, X.C. Tai, S. Borok, A.M. Bruckstein, Orientation-matching minimization for image denoising and inpainting. Int. J. Comput. Vis. **92**(3), 308–324 (2011)
22. W.G. Litvinov, T. Rahman, X.C. Tai, A modified TV-stokes model for image processing. SIAM Sci. Comput. **33**(4), 1574–1597 (2011)
23. S.F. Johnston, Lumo: illumination for CEL animation, in *The 2nd International Symposium on Non-photorealistic Animation and Rendering* (ACM, New York, 2002), pp. 45–52
24. D. Meyers, S. Skinner, K. Sloan, Surfaces from contours. Trans. Graph. **11**(3), 228–258 (1992)
25. S. Masnou, J. Morel, Level lines based disocclusion, in *5th IEEE International Conference on Image Processing*, Chicago, Oct 4–7 (1998), pp. 259–263
26. T. Meyer, Coastal elevation from sparse level curves. Summer project under the guidance of T. Wittman, A. Bertozzi, and A. Chen, UCLA (2011)
27. L. Alvarez, F. Guichard, P.L. Lions, J.M. Morel, Axioms and fundamental equations of image processing. Arch. Ration. Mech. **123**, 199–257 (1993)
28. V. Caselles, J.-M. Morel, C. Sbert, An axiomatic approach to image interpolation. Trans. Image Proc. **7**(3), 376–386 (1998)
29. R. Franke, Scattered data interpolation: test of some methods. Math. Comput. **38**, 181–200 (1982)
30. J. Meinguet, Approximation theory and spline functions, in *Surface Spline Interpolation: Basic Theory and Computational Aspects* (Holland, Dordrecht, 1984), pp. 124–142
31. J.C. Carr, W.R. Fright, R.K. Beatson, Surface interpolation with radial basis functions for medical imaging. Trans. Med. Imaging **16**(1), 96–107 (1997)
32. L. Mitas, H. Mitasova, *Spatial Interpolation* (Wiley, New York, 1999)
33. J. Lellmann, J.M. Morel, C.-B. Schönlieb, *Anisotropic Third-Order Regularization for Sparse Digital Elevation Models* (Springer, Berlin, 2013)
34. R. Glowinski, P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics* (Society for Industrial and Applied Mathematics, Philadelphia, 1989)
35. X.-C. Tai, C. Wu, Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model, in *SSVM*, ed. by X.-C. Tai, K. Mrken, M. Lysaker, K.-A. Lie. Lecture Notes in Computer Science, vol. 5567 (Springer, Berlin, 2009), pp. 502–513
36. C.L. Wu, J.Y. Zhang, X.C. Tai, Augmented Lagrangian method for total variation restoration with non-quadratic fidelity. Inverse Prob. Imaging **5**(1), 237–261 (2011)
37. C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, vol. 2 (SIAM, Philadelphia, 2000)
38. Y. Wang, J. Yang, W. Yin, Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imaging Sci. **1**(3), 248–272 (2008)
39. C. Van Loan, *Computational Frameworks for the Fast Fourier Transform* (Society for Industrial and Applied Mathematics, Philadelphia, 1992)
40. C.A. Elo, Image denoising algorithms based on the dual formulation of total variation, Master thesis. University of Bergen, 2009

Part II
Image Enhancement, Restoration and Registration

Chapter 4
Variational Methods for Gamut Mapping in Cinema and Television

Syed Waqas Zamir, Javier Vazquez-Corral, and Marcelo Bertalmío

Abstract The cinema and television industries are continuously working in the development of image features that provide a better visual experience to viewers, increasing spatial resolution, frame rate, contrast, and recently, with emerging display technologies, much more vivid colors. For this reason there is a pressing need to develop fast, automatic and reliable gamut mapping algorithms that can transform the colors of the original content, adapting it to the capabilities of the display or projector system in which it is going to be viewed while at the same time respecting the artistic intent of the creator. In this article we present a review of our work on variational methods for gamut mapping that comply with some basic global and local properties of the human visual system, producing state-of-the-art results that appear natural and are perceptually faithful to the original material.

Introduction

The cinema and television industries are continuously working in the development of image features that provide a better visual experience to viewers; these image attributes include large spatial resolution, high temporal resolution (frame rate), rich contrast, and vivid colors. Virtually all display devices work on a similar principle; they use three well chosen red, green and blue color primaries that can be mixed in proper proportions to create different colors. These colors can be visualized in a 3D space; however when describing colors it is a very common and convenient practice to decouple the luminance component from the chromatic components. So by ignoring luminance, the chromatic content can be represented on a 2D plane known as the CIE xy chromaticity diagram (shown in Fig. 4.1). In this figure the tongue-shaped region corresponds to the chromaticities of all the colors a standard

S. W. Zamir (✉) · J. Vazquez-Corral · M. Bertalmío
Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain
e-mail: waqas.zamir@upf.edu; javier.vazquez@upf.edu; marcelo.bertalmio@upf.edu

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_4

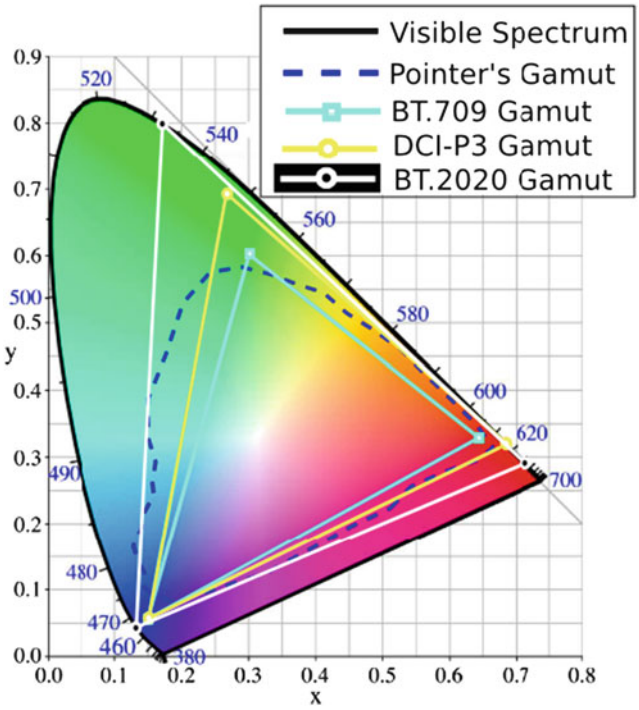


Fig. 4.1 Gamuts on CIE xy chromaticity diagram

observer can see. But for a display device, the xy coordinates of the three color primaries define its gamut, which is the range of colors it can reproduce. It is important to note that the three RGB primaries of a display form a triangle in the chromaticity diagram and the colors that this device generates will always lie within this triangle. Moreover, it is evident that there is no set of physically-realizable primaries that can create a gamut capable of covering the full visible spectrum (in fact no finite set of primaries can, due to the shape of the spectral locus). This implies that there are many colors that we are able to see but display devices are not capable of reproducing. Hence, a device gamut is a subset of the human vision gamut. With the goal of making a display device that can reproduce more of the colors that we can perceive, several multiple-primary displays have been proposed that make use of four [16], five [15, 73], and even six color primaries [64, 75]; however, the quest to make an optimal display still continues.

State-of-the-art digital movie cameras are capable of shooting content with a large range of colors. However, before a movie is released, its colors have to be "fitted" to the standard gamuts: DCI-P3 [68] used for digital cinema projections, or BT.709 [29] used for cable and broadcast TV, DVD, Blu-Ray and streaming. These standard color gamuts DCI-P3 and BT.709 (shown in Fig. 4.1) exist to ensure a consistent movie presentation across different digital cinema projectors and TVs,

respectively. This adaptation to a standard gamut implies altering the range of colors (and contrast) of the original captured content. The process of color reproduction is either carried-out within the camera in live TV broadcasts (or low-budget movie productions), or performed offline by colorists (expert technicians) in the cinema industry. In order to modify the color gamut, colorists at the post-production stage specify manually only a few colors using 3D look-up-tables (LUTs), while the rest are interpolated without taking into account their spatial or temporal distribution [8]. As a consequence, the reproduced movie may have false colors that were not originally present. To deal with this issue, intensive manual correction is usually necessary, commonly performed in a shot-by-shot, object-by-object basis. This process is difficult, time consuming and expensive, and therefore it makes an automated procedure called Gamut Mapping (GM) very desirable: GM transforms colors of a source material to a target gamut.

There are two types of GM: Gamut Reduction (GR) and Gamut Extension (GE). In the process of gamut reduction, colors are mapped from a larger source gamut to a smaller destination gamut. For example, footage mastered for cinema projection has to pass through a GR operation before it can be displayed on a TV [5, 34]. On the other hand, gamut extension involves the mapping of colors from a smaller source gamut to a larger destination gamut. For example, state-of-the-art digital cinema projectors have a wide color gamut but they often receive cinema footage that is encoded with a limited gamut as a precaution measure against regular (or poor) projectors; therefore, in order to realize the full color rendering potential of these new projectors, a GE procedure is needed [8]. The process of GE is gaining importance with the introduction of laser projectors [38, 67] and ultra-high definition (UHD) TVs. These new displays use pure (very saturated) color primaries which enable them to cover the very wide next generation UHDTV standard gamut known as BT.2020 [30], and therefore reproducing all the frequently occurring real surface colors as defined by Pointer's gamut [60].

Gamut mapping is not only important in the film or the broadcast industry but it is also an essential module in the image reproduction pipeline of printing technologies, where the end goal is to minimize the perceptual difference of the same image when it is viewed on a display device and when it is printed. Other application domains of gamut mapping include handheld devices (mobile, tablet computers), websites, photo-sharing online platforms, computer graphics, animation and video games.

At this point it is necessary to mention a key difference between the application of gamut reduction and gamut extension. Gamut reduction is required, not optional, when the colors of the input image fall outside the display's gamut; if not, the display will reproduce the image with artifacts and loss of spatial detail. On the contrary, gamut extension is not essential, rather it is considered as an enhancement operation [54]. For example, displaying a BT.709 footage (represented in a BT.2020 container) as it is on a wide-gamut BT.2020 supported display device will not cause any visual color distortion, but if we do not extend colors of the input footage to the gamut of display we will be missing the color rendering potential of the wide-gamut screen.

Related Work

There is a large number of works that exist in the literature on gamut mapping; while majority of these gamut mapping algorithms (GMAs) deal with the gamut reduction problem, only few of them perform gamut extension. We can divide GMAs into two broad categories: global GMAs and local GMAs. Global (also called non-local or non-adaptive) GMAs modify each color of an image independently, meaning that these methods completely ignore the spatial color distribution in the image. On the other hand, local GMAs modify pixel values by taking into account their neighborhoods. For an in-depth explanation of GMAs, we refer the interested reader to the comprehensive book [54].

Gamut Reduction Algorithms (GRAs)

The global gamut reduction algorithms can be classified into two sub-classes: clipping and compression. Gamut clipping is the simplest approach to perform gamut mapping where colors that lie inside the destination gamut are left unmodified while those colors that fall outside are projected onto the destination gamut boundary [33, 46, 47, 57, 65, 71]. Murch and Taylor [57] proposed the Hue Preserving Minimum ΔE (HPMINDE) algorithm, that clips an out-of-gamut (OOG) color to the closest color (in terms of ΔE error) on the destination gamut boundary along lines of constant hue. CIE recommends adding HPMINDE as a benchmark in the evaluation of GRAs. Masaoka et al. [47] presented a GRA that aims at mapping colors from a very wide gamut (BT.2020) to a small gamut (BT.709), while preventing excessive chroma loss; this involves dealing differently with colors of low luminance and high luminance, specifically the authors map bright colors without respecting the constant hue lines in order to avoid a blown-out appearance. All gamut clipping methods project the whole OOG color segment to a single point on the destination gamut boundary, and this may produce a gamut mapped image with a visible loss of texture and color gradients. To overcome this issue, gamut compression algorithms [23, 26, 31, 53, 66, 74] modify all the colors present in an input image. Gamut compression algorithms map a larger OOG color segment to a smaller in-gamut color segment and therefore they may cause a significant loss in saturation, especially when the difference between the source gamut and the target gamut is large.

Local GRAs are also known as ‘spatial’ methods. The spatial GRAs of [4, 51] and [81] perform gamut reduction in two stages: firstly the gamut of the input image is reduced using a global method, and secondly the high frequency image detail (texture) is added to the gamut-reduced image using a spatial filtering operation. Morovič [55] introduced a multilevel, full-color GRA that first decomposes the image into a number of spatial frequency bands. Secondly, at the lowest frequency band, the lightness compression is applied followed by the application of initial gamut mapping. Then, the next higher frequency band is added to the gamut mapped

image and again gamut mapping is applied to the resulting image. This step is repeated until the highest frequency band is reached. McCann [48, 49] proposed a Retinex-inspired framework that performs spatial comparisons to preserve the local gradients to obtain the final gamut mapped image. Alsam and Farup [1] proposed an iterative GRA that at iteration level zero behaves as a gamut clipping algorithm, whereas, by increasing the number of iterations, the solution approaches spatial gamut mapping. Nakauchi et al. [58] defined gamut mapping as an optimization problem where they use a perceptual metric to minimize the perceived differences between the original and the reproduced image in order to obtain the final reduced-gamut image, and many other algorithms [36, 40, 62, 63] followed a similar optimization idea. Local GRAs are adaptive and flexible but at the same time more complex and computationally far more expensive than global GRAs. Moreover, spatial GRAs are often based on many assumptions, and may produce halo artifacts.

Gamut Extension Algorithms (GEAs)

Unlike GRAs, there exists a small number of GEAs in the literature. One may take a GRA and use it in the reverse direction to obtain an image with an extended gamut [54]. However, the key struggle is to produce gamut extended images that are natural, pleasant and perceptually as similar as possible to the original images.

Hoshino [27] proposed the pioneering global GEA that first maps the lightness using a non-linear tone reproduction curve, and then the chroma is mapped along lines of constant lightness and hue. A revised version of the same GEA was introduced in [28] in order to produce extended-gamut images that are more pleasant and natural in appearance. Kang et al. [32] developed a GEA based on the numerical fitting of data that was obtained by allowing a group of observers to manually extend lightness and chroma in a linear manner. Anderson et al. [2] presented a semi-automated framework in which an expert first expands the color gamut of some key frames from which a LUT is learned. This LUT is then applied to the rest of frames to perform gamut extension. While all these aforementioned GEAs are image-independent methods, the authors of [13, 59] and [25] introduced global GEAs that first analyze the colors of the input image and classify them according to some criterion. Afterwards a mapping procedure is applied in order to treat these colors in different manner. The algorithm of [44] uses the CIELUV color space to expand the color gamut of the input image from an anchor point while keeping the hue constant. Kim et al. [35] proposed a GEA with three types of mapping directions: chroma mapping, mapping along lines from the origin, and adaptive mapping that is a compromise between the first two strategies. Laird et al. [39] proposed and evaluated five GEAs that are explained in more detail later in this chapter where we will compare our GEAs with them.

There are a few other GEAs [50, 69, 70] that mainly aim at preserving the skin tones in the reproductions. In three different thesis works [12, 14, 42], authors use GRAs in the reverse direction to obtain images with an extended gamut.

To the best of our knowledge, apart from the GEAs that we present in this book chapter, there is only the following previous work that extends colors taking into account their local neighborhoods. Li et al. [41] presented a two-stage framework. In the first stage the color gamut of the input image is globally extended using a non-linear hue-varying extension function. And in the second stage an image-dependent chroma smoothing procedure is applied in order to avoid an over-enhancement of contrast and to retain in result the local information of the input image.

Reproduction Intent and Evaluation

Every GMA aims at reproducing content according to the specific application in which it is going to be employed. For instance, a GMA that is intended to be used in the cinema industry needs to reproduce images that are faithful to the vision of the content's creator. Whereas in the television industry, TV makers prefer distorting image attributes such as tones and colors in ways that they think consumers may find visually pleasant [61].

On one hand, GMAs with *accurate reproduction intent* aim at reproducing images that are perceptually faithful to the originals. On the other hand, GMAs with *pleasant reproduction intent* reproduce images that a viewer deems pleasant. The latter may imply departing from the original content as much as needed, and incorporating steps that could modify the aesthetics (contrast, sharpness, etc.) of images. In this work we discuss GMAs that comply with the accurate reproduction intent, and the results these algorithms produce are evaluated with the criterion of accuracy. For a GMA to be adopted by the movie industry, it is important to preserve the artistic intent of the content's creator in the reproduced image, and this is guaranteed when the accurate reproduction intent is chosen in the evaluation, which can be either subjective or objective.

Subjective Evaluation

In the case of subjective evaluation, subjects take part in psychophysical experiments where they have to choose or rate the reproductions based on a criterion (preference or accuracy). Psychophysical studies to evaluate GMAs follow closely the guidelines of [17] in which the conditions of the experimental setup are detailed: for example, viewing conditions, method of comparison, minimum number of observers and test images, maximum time duration per experiment session, methods for subject's color vision testing, etc. Some commonly used subjective methods are:

- **Pair Comparison:** Two different gamut-mapped versions of an original image are shown to observers in isolation or alongside the original image. Observers are then asked to select the gamut-mapped image which exhibits more of the

property (pleasantness, naturalness, or accuracy) being evaluated. There are several studies that make use of the pair comparison scheme to evaluate GMAs such as the works of [11, 18, 24, 52, 56], and [39].

- **Category Judgment:** In a category judgement experiment the observer is shown, one at a time, several images reproduced by different GMAs and asked to evaluate their pleasantness by assigning them descriptive names (such as excellent, good, fair, poor, bad) or just numbers from an integer scale (e.g., 1–10). This method is based on the law of categorical judgement [72]. Before starting the main experiment, it is recommended by CIE [17] to show observers for training purposes a pair of images, one of which serves as an example of the best quality image and the other one as a worst case example.
- **Rank Order:** Observers are asked to rank a given set of images according to a perceptual attribute.

Objective Evaluation

Subjective evaluation is time consuming, expensive and often unreliable if the number of observers is not sufficient. Therefore, an alternative is to use objective quality metrics that are capable of finding specific distortions in reproduced images. There exists a vast variety of image quality metrics [6, 7, 19, 43, 45, 63] in the literature that could in principle be used to quantify the results of GMAs. Hardeberg et al. [24] and Baranczuk et al. [6] presented psychophysical studies to identify the best performing objective measure for the GR problem. It is important to note that the ranking of color metrics for gamut reduction may not be consistent in the context of gamut extension if the metrics are not trained to predict well the distortions found in gamut extended images.

Gamut Mapping in RGB Based on Perceptually-Based Color and Contrast Enhancement

In [9] the authors propose a variational method for color and contrast enhancement consisting in minimizing the following energy functional:

$$E(I) = \frac{\alpha}{2} \sum_x \left(I(x) - \frac{1}{2} \right)^2 - \frac{\gamma}{2} \sum_x \sum_y w(x, y) |I(x) - I(y)| + \frac{\beta}{2} \sum_x (I(x) - I_0(x))^2 , \tag{4.1}$$

where α, β and γ are constant weights, I is a color channel (R, G or B) in the range $[0, 1]$, I_0 is the original image channel, $w(x, y)$ is a normalized Gaussian

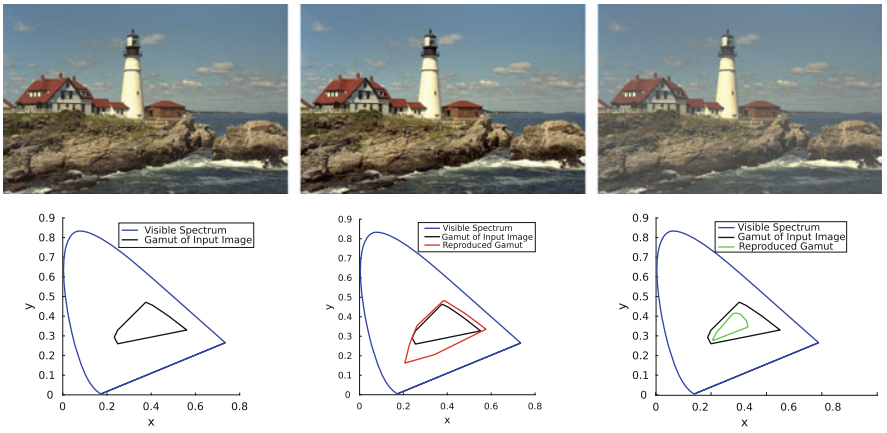


Fig. 4.2 Example images and their corresponding gamut plots. Left column: input image. Middle column: extended-gamut image. Right column: reduced-gamut image

kernel of standard deviation σ, and x and y are pixel locations. The constants α and β are always positive, so minimizing $E(I)$ penalizes the departure from the original image (third term of the functional) and from a mean value of $1/2$ (first term). If γ is positive, minimizing $E(I)$ amounts to increasing $\sum_x \sum_y w(x, y)|I(x) - I(y)|$, i.e. the local contrast. If, on the contrary, $\gamma < 0$, then the minimization of Eq. (4.1) *reduces*, not increases, the contrast, as pointed out in [10].

Figure 4.2, top row, shows example outputs that can be obtained with this approach. The left image is the input, the middle image is the result that minimizes Eq. (4.1) for some $\gamma > 0$, and the image on the right is the result that minimizes Eq. (4.1) for some $\gamma < 0$. Notice how, in the middle image, the contrast has been enhanced and the colors have become more saturated, while the opposite happens in the image on the right, where contrast has been reduced as well as the chromaticity of the pixel colors. This is corroborated in the bottom row, that plots the chromaticity diagrams for the pictures above.

This example highlights the potential for the variational method of [9] to be used for gamut mapping: with $\gamma > 0$ it is capable of producing gamut extension, and gamut reduction when $\gamma < 0$. That is the approach we followed in [76]: after replacing the value of $1/2$ in the first term of the functional with the global mean average value μ, the minimum of Eq. (4.1) can be obtained by iterating

$$I^{k+1}(x) = \frac{I^k(x) + \Delta t \left(\alpha\mu + \beta I_0(x) + \frac{\gamma}{2} R_{I^k}(x)\right)}{1 + \Delta t(\alpha + \beta)}, \tag{4.2}$$

where the initial condition is $I^{k=0}(x) = I_0(x)$, $R_{I^k}(x)$ indicates the contrast function

$$R_{I^k}(x) = \frac{\sum_y w(x, y) s \left(I^k(x) - I^k(y)\right)}{\sum_y w(x, y)}, \tag{4.3}$$

and the slope function $s(\cdot)$ is a regularized approximation of the sign function, which in [9] is chosen as a polynomial of degree 7.

GRA-RGB [76]: Gamut Reduction Algorithm on RGB

In detail, for gamut reduction we propose in [76] an iterative approach, where at each iteration we run Eq. (4.2) for some particular α, β, and γ until we reach the steady state. At this point we check all pixels and those that have values that are inside the target gamut are kept untouched for all subsequent iterations, i.e. these pixels will be part of the final output. Next we decrease the value of γ and proceed to the following iteration, repeating the process until all the out-of-gamut colors come inside the destination gamut. An example of this iterative procedure is shown in Fig. 4.3, where the vivid green color marks the pixels that are out-of-gamut at that iteration. Figure 4.4b shows how the gamut is iteratively reduced towards the target.

The standard deviation σ of the Gaussian kernel w has an impact on the final appearance: small σ values preserve colors but introduce artifacts, while larger σ values yield images that are free from artifacts but with less saturated colors. This is

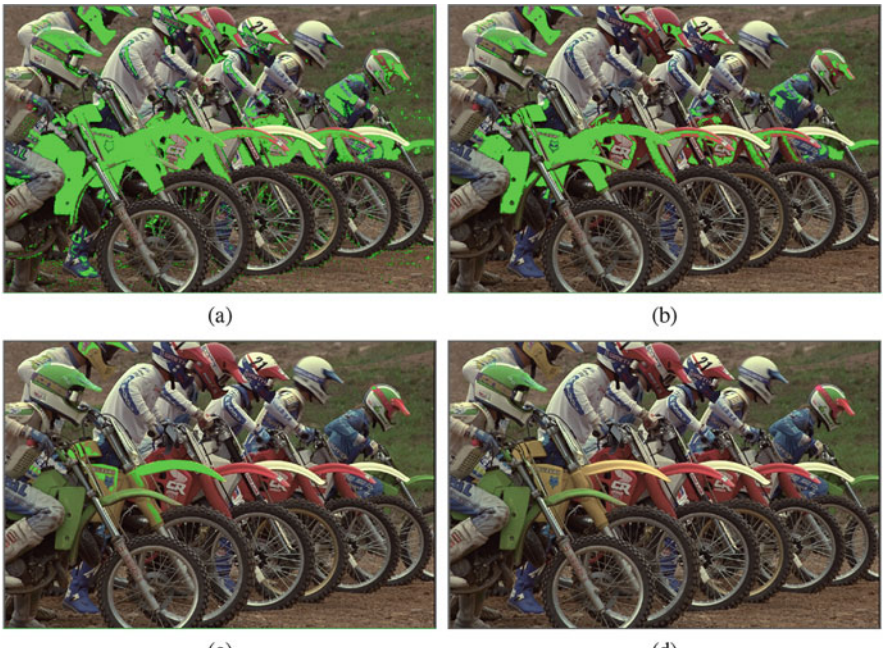
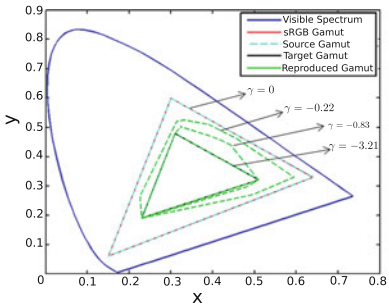


Fig. 4.3 Gradual mapping of colors. Out-of-gamut colors (in green) when (**a**) $\gamma = 0$, (**b**) $\gamma = -0.22$, (**c**) $\gamma = -0.83$, (**d**) $\gamma = -3.21$. As γ decreases the number of out-of-gamut pixels is reduced



(a)

(b)

Fig. 4.4 GR approach. (**a**) Gamuts on chromaticity diagram. (**b**) Top left: original image. Top right: $\gamma = -0.22$. Bottom left: $\gamma = -0.83$. Bottom right: $\gamma = -3.21$

why we compute four intermediate output results $\mathscr{I}_\sigma(x)$ with different values for σ, and for each pixel in the final output $\mathscr{I}_{final}(x)$ we select the value from that pixel location in the gamut mapped image $\mathscr{I}_\sigma(x)$ that has the minimum Lab ΔE distance with respect to the original image value $\mathscr{I}_{orig}(x)$

$$\mathscr{I}_{final}(x) = \arg\min_{\mathscr{I}_\sigma} \left(Lab(\mathscr{I}_\sigma(x)) - Lab(\mathscr{I}_{orig}(x))\right)^2, \quad \forall x, \sigma \in \{\sigma_1, \cdots, \sigma_4\}. \tag{4.4}$$

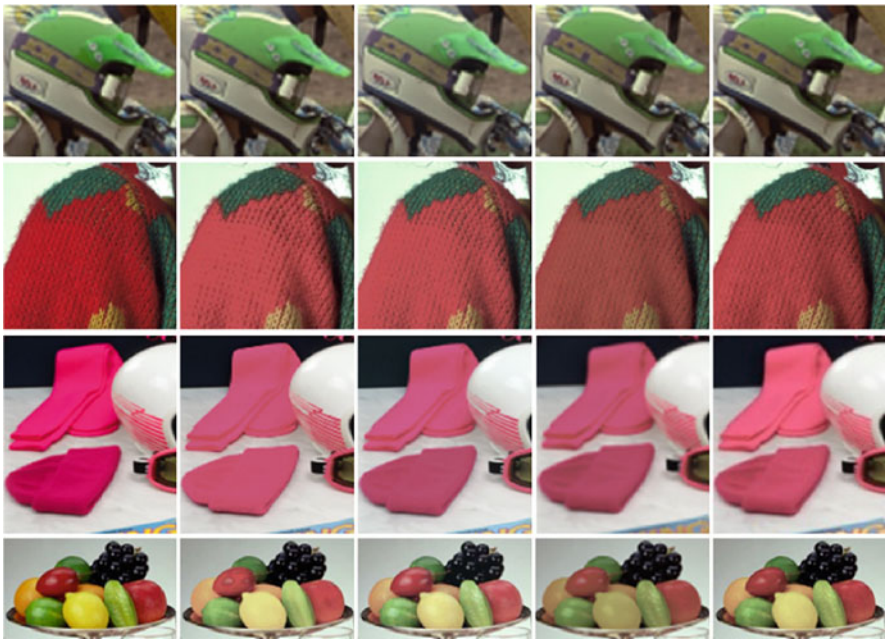


Fig. 4.5 Detail preservation using GRAs on still images. Column 1: original cropped regions. Column 2: output of HPMINDE [57]. Column 3: output of Lau et al. [40]. Column 4: output of Alsam et al. [1]. Column 5: output of our GRA-RGB

The proposed method is shown in [76] to outperform the state-of-the-art both visually (Fig. 4.5), with better color and detail preservation, and quantitatively, in terms of the CID perceptual metric presented in [43]. This is the case both for static images and for video sequences, which show no spatio-temporal artifacts.

GEA-RGB [76]: Gamut Extension Algorithm on RGB

For gamut extension the process is no longer iterative, but it consists of three stages. First we slightly shift to the right the histogram of each channel, so as to prevent pixel values from going to black in the subsequent contrast enhancement step. Next we run Eq. (4.2) with positive α, β, and γ until we reach the steady state. The value of γ is selected so that the processed image has a gamut slightly larger than the destination gamut. Finally, the out-of-gamut values are mapped back inside using the GRA-RGB method described in the previous section.

While in [76] we present experimental comparisons and visual and quantitative assessments of the results of our GEA, these tests were rather limited because they were not performed in realistic conditions: the target gamut was not an actual wide gamut like the DCI-P3 used in cinema, and the validation did not involve

psychophysical tests in the proper set-up (e.g. large screen, low ambient light like in a movie theater). We addressed these limitations in our following work, described next.

Gamut Extension in CIELAB Color Space

Our GEA-RGB [76] presented in the previous section, due to its inherent behavior of expanding colors by increasing the contrast of the image, produces results with over-enhanced contrast, which in turn makes a few colors go towards black (loss of saturation), as it is visible in Fig. 4.6b. It can be seen that the overall contrast of the reproduction is increased noticeably, making it depart from the original image. Also, the over-enhancement of contrast causes loss of color details as it is shown in the area highlighted by a bounding box in Fig. 4.6b. To overcome these problems, we

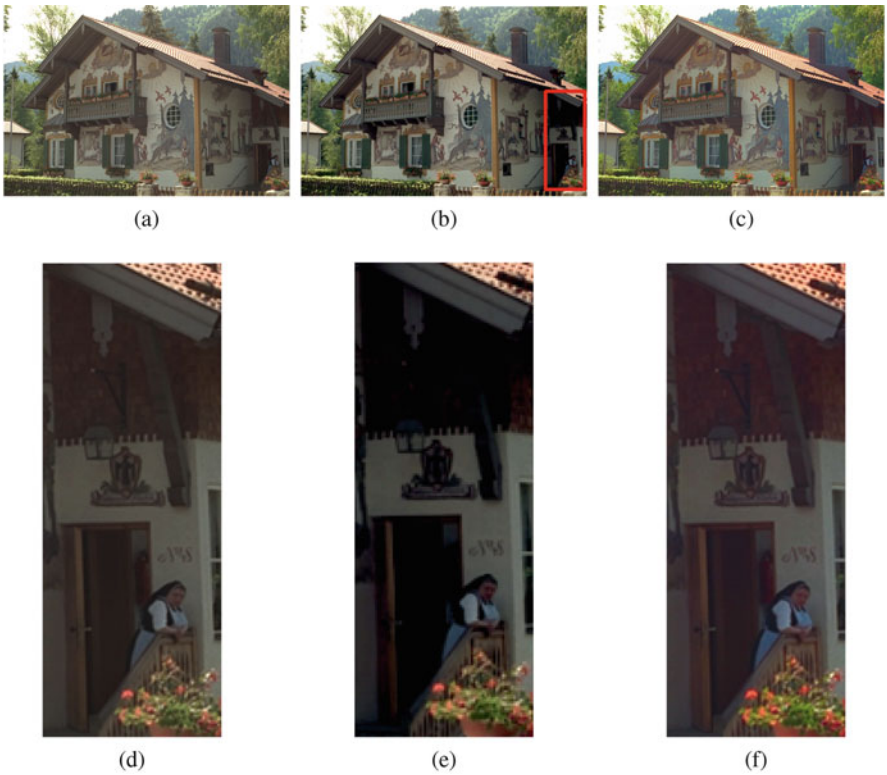


Fig. 4.6 Gamut extension example. (**a**) Input image. (**b**) Result of GEA-RGB [76]. (**c**) Result of GEA-LAB1 [77]. Bottom row (**d**)–(**f**): Zoomed-in view of the regions cropped from the top row. Original image is courtesy of [37]

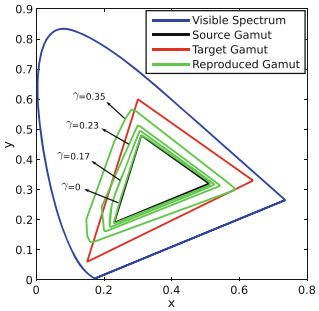
introduced in [77] and [79] two new GEAs based on [76]; these methods perform gamut extension using the same energy functional (Eq. (4.1)) and its corresponding evolution equation (Eq. (4.2)) as used by Zamir et al. [76], but under the CIELAB color space. This key modification eliminates not only the problems with saturation and contrast in the reproduced images, but also the need to perform any sort of preprocessing as it was the case with the GEA-RGB [76].

GEA-LAB1 [77]: Gamut Extension Algorithm

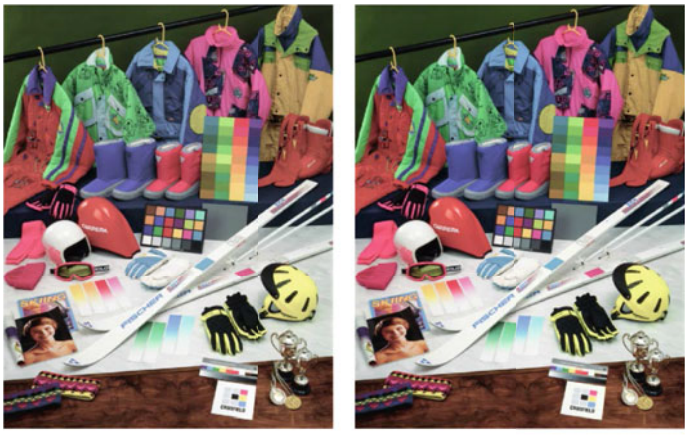
To perform gamut extension, our GEA-LAB1 [77] first converts the RGB input image to the CIELAB color space, and then only maximizes the contrast of the chromatic components 'a' and 'b' using Eq. (4.2), while keeping the lightness component constant. To show how the evolution equation (4.2) extends the color gamut, an example with several different gamuts (visible spectrum, source gamut, target gamut and reproduced gamut) on a chromaticity diagram is shown in Fig. 4.7a. It is important to note that for each set of values for α, β, and γ, the evolution equation (4.2) has a steady state. For example, it is shown in Fig. 4.7a that when $\beta = 1$, $\alpha = 0$, and $\gamma = 0$ we obtain the original image as the steady state of the evolution equation. Moreover, it can be seen in the same figure that as we increase γ the steady state of Eq. (4.2) has a gamut which is gradually larger. Figure 4.7a also shows that the colors of the source gamut can be expanded to the destination gamut just by using a large enough value for γ ($\gamma = 0.35$ in this case). And to select an adequate γ value, we keep increasing the γ value and running evolution equation (4.2) to steady state until the gamut of the input image exceeds the target gamut up to a certain threshold T. This threshold T defines a stopping criteria according to which if $T\%$ pixels of the original image move out of the target gamut we should stop performing extension. Additionally, the threshold T controls the amount of saturation: a large value of T provides a higher level of saturation, whereas a small value of T produces a less saturated output. For each γ value, the corresponding reproductions are shown in Fig. 4.7b–e. After this, the colors that were placed outside the target gamut in previous iterations are mapped back inside using our GRA-RGB [76]. However, since this method uses a fixed value of threshold T for all the images, the results can be off from the ground truth and may present hue shifts.

GEA-LAB2 [79]: Gamut Extension Algorithm Driven by Hue, Saturation and Chroma Constraints

To overcome the issues of GEA-LAB1 [77] we introduced in [79] another method that works iteratively with added constraints to perform gamut extension in terms of the contrast coefficient γ. The general structure of this algorithm (GEA-LAB2),



(a)



(b) (c)

(d) (e)

Fig. 4.7 Gamut extension approach. (**a**) Gamuts on chromaticity diagram. Gamut extension results: (**b**) input image ($\gamma = 0$), (**c**) gamut extended image with $\gamma = 0.17$, (**d**) $\gamma = 0.23$ and (**e**) $\gamma = 0.35$. As the γ value increases the gamut becomes larger; notice the increment in saturation of helmets, socks, ski suits and shoes. Original image is courtesy of [37]

that also operates on the 'a' and 'b' components in the CIELAB color space, is as follows.

First, we initialize our final image and set all its values to zero. Then, we run the evolution equation (4.2) for $\beta = 1$, $\alpha = 0$, and $\gamma = 0$ until we reach the steady state. For each pixel of the steady state solution we check if it accomplishes at the same time three different constraints described below: one in saturation, one in hue, and one in chroma. In case it does, we select the current value of the pixel as the value of this pixel in the final image. We keep repeating this process by increasing γ and setting $\alpha = \frac{\gamma}{20}$ until either the final image has been filled or until the gamut of the original image exceeds our destination gamut up to a threshold T. At this point, we stop the iterations and replenish the final image with the pixel values obtained in the last iteration, taking special care of those pixels that have exceeded the gamut (to which we apply a small reduction using our GRA-RGB [76]).

Let us now briefly describe our three constraints. In short, regarding the saturation, we aim at a result where no pixel is less saturated than in the original image; regarding the hue, unwanted changes on the hue (or tint) of the objects should be avoided; regarding the chroma, we should prevent the over-enhancement of natural colors such as sky, skin, or memory colors.

Another important part of our approach is the definition of a scaled destination gamut that allows our method to work under different target and source combinations. The process is as follows. Given the original image (I_0) whose gamut we want to extend, we convert the RGB values of this image into luminance values Y and chromaticity values x, y. Then, we define the first reference point p_{r1} as the mean among all the vertices of the gamut of I_0 on chromaticity values. We show the reference point p_{r1} and the convex hull of the chromaticities (CG_{chrom}) of I_0 in Fig. 4.8a. We then define a set of lines (L) that connect the reference point p_{r1} to any point in the border of CG_{chrom}. Finally, we generate new points (one from each line in L) using a scaling factor equal for all the lines. This scalar (η_1) is defined such as none of the new points falls outside the target gamut and at least one of them touches the boundary of the target gamut as shown in Fig. 4.8a. Similarly, we calculate another scaling factor η_2, but this time using the mean of all chromaticity values of the image as reference point p_{r2} and repeat the process. Once we have the scaling factors η_1 and η_2, we apply them on the xyY triplets that make the three-dimensional (3D) convex hull of the original image I_0 (shown in Fig. 4.8b) to obtain two 3D scaled gamuts. The final scaled destination gamut is defined as the intersection of both 3D scaled gamuts as illustrated in Fig. 4.8c. An example with all the relevant gamuts is shown Fig. 4.8d.

Qualitative Experiments and Results

Methodology

Let us start this section by pointing out that the final goal a GE method must accomplish is to respect as much as possible the intent of the person who created the

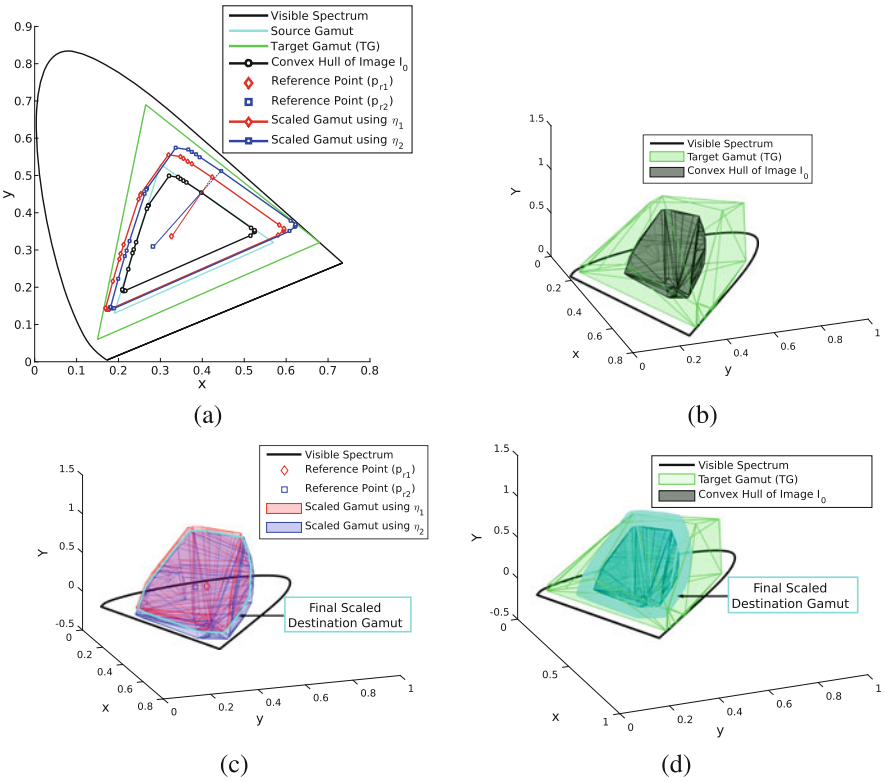


Fig. 4.8 Scaled destination gamut computation

material. This is the reason why we evaluate our method via a forced-choice pairwise comparison experiment that compares an original ground-truth image versus the results of two different methods: we inquire the observers for the result that is closer to the ground-truth (even if it is not the most pleasant for them).

A general framework for our experiment is shown in Fig. 4.9. The first task is to obtain both the wide-gamut ground truth images and the limited-gamut input images. For obtaining the wide-gamut test images we have used a camera that is capable of capturing images in RAW format which can then be associated with a particular wide-gamut color space (ProPhoto RGB) to obtain true color images. Along with the 21 images shown in Fig. 4.10, we use 9 other test images that come from professional feature films. Let us note that these original images may have colors that fall outside the gamut of the cinema projector we use for displaying the content; therefore, to create the ground truth, we map the colors of the test images to the gamut of the projector by using the state-of-the-art gamut reduction algorithm of [1]. Next, to create the limited-gamut input images, we apply the same state-of-the-art gamut reduction method of [1]. Once we have the input images ready, we

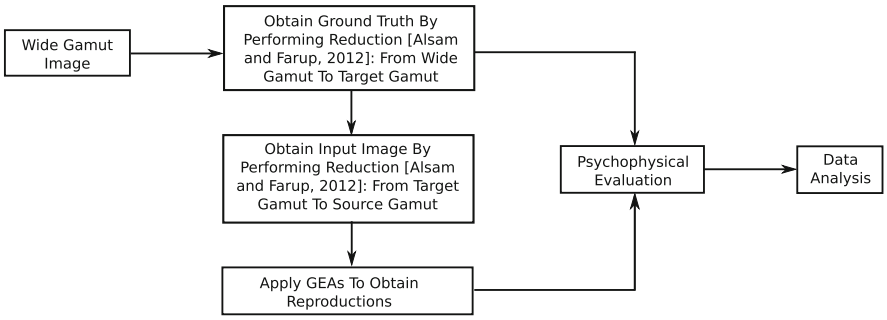


Fig. 4.9 A schematic of the evaluation for GEA-LAB2 [79]

Fig. 4.10 Some of the wide-gamut images used in our tests. Note that only the central part of the images is shown

apply to them our GEA-LAB2 and four competing GEAs, namely LCA, CE, HCM, SDS [39].

To test the robustness of our approach with respect to different combinations of source and target setups, we defined two setups for our experiments:

1. *Mapping from small gamut to DCI-P3 gamut:* laser displays with gamuts vastly extending current capabilities are becoming popular. This leaves us with a clear problem: in the near future, there will exist large differences between source standard gamuts and displays' gamuts. Therefore, to emulate this behavior we

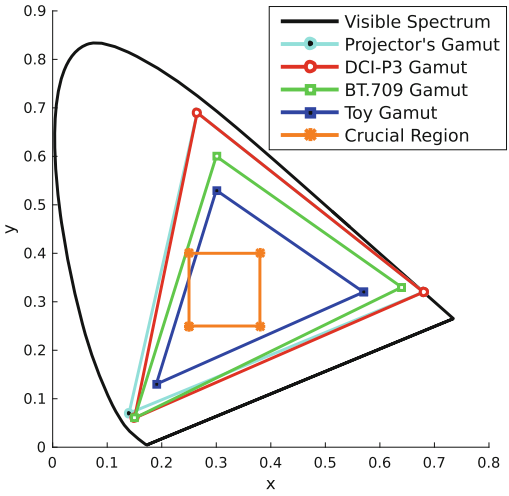


Fig. 4.11 Gamuts on chromaticity diagram

Table 4.1 Primaries of gamuts

	Red primaries		Green primaries		Blue primaries	
Gamuts	*x*	*y*	*x*	*y*	*x*	*y*
BT.2020	0.708	0.292	0.170	0.797	0.131	0.046
BT.709/sRGB	0.640	0.330	0.300	0.600	0.150	0.060
DCI-P3	0.680	0.320	0.265	0.690	0.150	0.060
Projector	0.680	0.320	0.265	0.690	0.140	0.070
Toy	0.570	0.320	0.300	0.530	0.190	0.130
Mock	0.510	0.320	0.310	0.480	0.230	0.190

map the source images from the small 'Toy' gamut (slightly smaller than the BT.709 gamut, see Fig. 4.11, and for its primaries see Table 4.1) to the large DCI-P3 gamut. Our 'Toy' gamut was selected so that the difference in gamuts for this setup is almost equal to the difference between BT.709 and BT.2020.

2. *Mapping from BT.709 to DCI-P3 gamut:* in this setup we mimic the practical situation where the source material is in the BT.709 gamut and we map the source colors to the colors of the DCI-P3 gamut.

The room for the experiment has a low-light ambiance of 1 lx and the illumination measured at the screen was around 750 lx. The glare-free screen used in our experiments was 3 m wide and 2 m high. We used 15 observers (10 male and 5 female) all with correct color vision and with ages between 27 and 44 years (average of 32 years). Observers were asked to sit approximately 5 m away from the screen.

As already stated before, we used a forced-choice pairwise comparison. The observers were simultaneously shown three images: ground-truth (in the center) and the results of two gamut extension methods (located left and right of the ground-truth image). The selection instructions given to the observers were: (a) if there are any sort of artifacts in one of the reproductions, choose the other, and (b) if both of

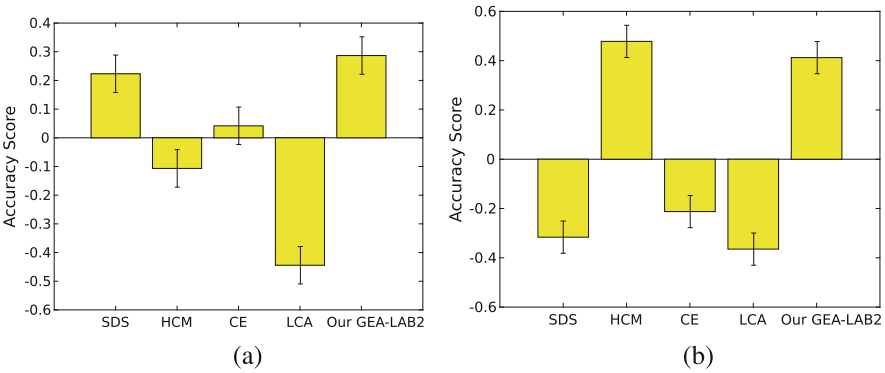


Fig. 4.12 Accuracy scores using 15 observers and 30 images. (**a**) Setup 1. (**b**) Setup 2

the reproductions have artifacts or are free from artifacts, choose the one which is perceptually more closer to the ground truth.

Moreover, to further validate the robustness of the different GEAs we asked nine experienced observers (who belong to the image processing community and participated in various psychophysical studies) to perform a second experiment. In this case, they were shown a pair of images side by side in the projection screen: the ground-truth image and the result for some particular method. In this case, observers were asked to look for artifacts and hue shifts in the reproductions as compared with the original material.

Results

In order to compute the accuracy scores from the raw psychophysical data we use the data analysis procedure presented in [53]. The analysis for the first and second setups are shown in Fig. 4.12a and b, respectively. For the first setup we can see that, when there is a large difference among the source-target gamut pair, our GEA-LAB2 produces images that are perceptually more faithful to the original as compared with the other competing algorithms. Regarding the second setup we can see that, when the difference between source and target gamut is smaller, the ranking order of the GEAs changes dramatically. In this case the HCM algorithm is ranked as the most accurate method, with our GEA showing comparable performance with it.

Results for this second experiment for setups 1 and 2 are shown in Fig. 4.13a, b. These results are computed as the average of cases where the observers noticed the visual distortions: artifacts or hue shifts. For the setup 1 subjects noticed artifacts in 25% of the reproductions obtained using the LCA algorithm and in 12% of the images in the case of the CE algorithm. The observers also confirmed that GEA-LAB2 produces images with very low error rate, around 2%. In terms of hue shifts, both the SDS and HCM algorithms show strong hue shifts. Regarding the setup 2 we can see that the SDS and HCM algorithms produce gamut extended images

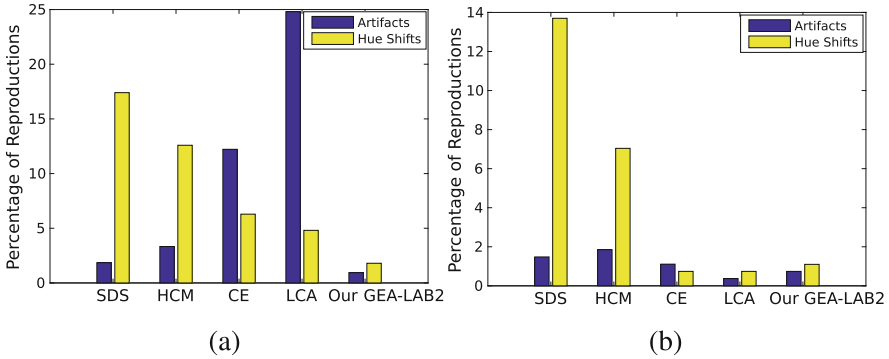


Fig. 4.13 Percentage of reproductions in which nine experienced observers noticed visual distortions. (**a**) Setup 1. (**b**) Setup 2

Fig. 4.14 Examples of artifacts. (**a**) Ground Truth. (**b**) Output of CE algorithm [39]. (**c**) Output of LCA algorithm [39]. (**d**) Output of our GEA-LAB2. (**e**)–(**h**) Zoomed-in view of the regions cropped from the top row. Note that these are wide-gamut images where out-of-sRGB pixels are masked green

with strong hue shifts for 13.6% and 7% of the input images, respectively. It can be seen in the same figure that none of the competing algorithms produces images with distinct visual artifacts for setup 2, in which there are small color differences between source and target gamut. Finally, Fig. 4.14 presents examples of artifacts found by the observers, while Fig. 4.15 presents examples of hue shifts. In both cases

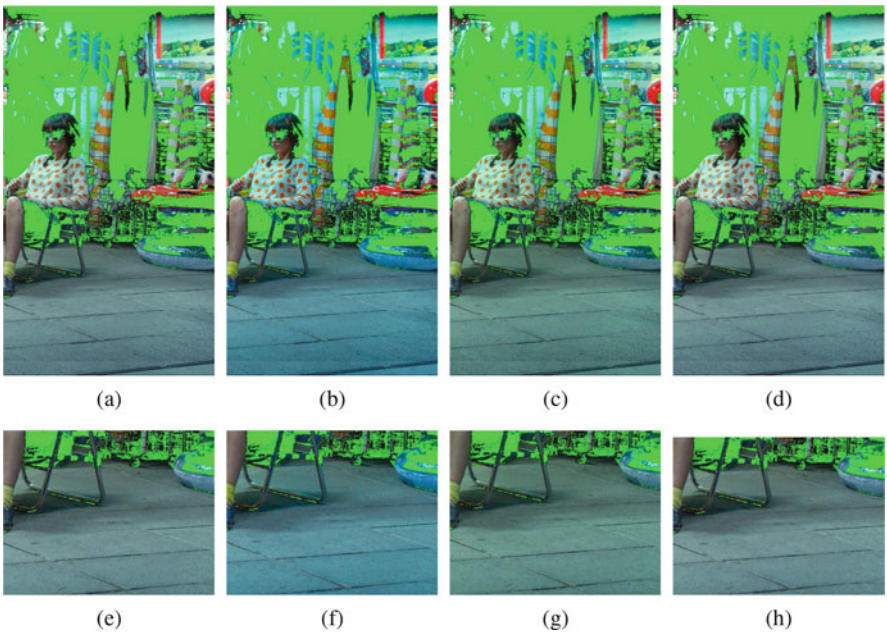


Fig. 4.15 Examples of hue shifts. (**a**) Ground Truth. (**b**) Output of SDS algorithm [39]. (**c**) Output of HCM algorithm [39]. (**d**) Output of our GEA-LAB2. (**e**)–(**h**) Zoomed-in view of the regions cropped from the top row. Note that these are wide-gamut images where out-of-sRGB pixels are masked green

we can see that our method is free of the problems presented by the competing ones. Let us note that in these figures we are only displaying the colors that are inside the sRGB gamut, and masking those that are outside.

Temporal Consistency Test

In order to examine the temporal consistency of the GEAs, we conducted a psychophysical study with nine experienced observers and two colorful image sequences with different levels of motion that had been extended using the different GEAs. Representative frames for both image sequences are presented in Fig. 4.16.

In this experiment, each observer was asked to inspect the following attributes: temporal color consistency (objects should retain the same hue, chroma and brightness), global flickering, local region flickering, and excessive noise. None of the observers noticed any temporal artifacts, which supports our choice to apply all competing GEAs on each frame independently. Finally, we want to stress that the quality of the input video is of high importance; if it contains any spatial artifacts due to compression or noise they may become prominent in the reproduced video.

Fig. 4.16 Representative frames of image sequences with toy gamut. (**a**) Image sequence 1. (**b**) Image sequence 2

Gamut Mapping Using Kernel Based Retinex (KBR) in HSV Color Space

GEA-KBR [78]: Gamut Extension Algorithm

The main limitations of the GEA-LAB2 [79] presented previously are, on one hand, its sensitivity to the correct choice of parameter values and, on the other hand, its significant computational cost: a non-optimized MATLAB implementation running on a machine with 8 cores 3.4-GHz CPU takes (on average) 4.5 min to process an image of resolution 656×1080 pixels.

Since the human visual system is very sensitive to changes in hue, it is recommended in the gamut mapping literature to leave the image hues unmodified, whenever possible [17, 54]. Therefore, we introduced in [78] a new GEA that works in the HSV color space and modifies only the saturation component while keeping hue and value constant. The proposed GEA is based on the kernel-based Retinex (KBR) method proposed by Bertalmío et al. in [10]. By using KBR we can pose the gamut extension problem as one of increasing saturation.

One fundamental mechanism of the KBR algorithm [10] is that it is capable of increasing contrast while being monotonically increasing, i.e. it can increase the contrast without decreasing the image values. Bertalmío et al. [10] propose the following formula to convert the intensity values I of an image into perceived lightness values L:

$$\begin{aligned} L(x) = &\sum_y w(x,y) f\left(\frac{I(x)}{I(y)}\right) \text{sign}^+(I(y) - I(x)) \\ &+ \sum_y w(x,y) \, \text{sign}^-(I(y) - I(x)), \end{aligned} \tag{4.5}$$

where $I(x)$ and $I(y)$ represent image values at pixel locations x and y, respectively, I is an image channel, f is a monotonically increasing function such that $f(r) \geq r$, $\forall r$, $w(x, y)$ is a normalized Gaussian kernel of standard deviation of σ, and the functions $\text{sign}^+(\cdot)$ and $\text{sign}^-(\cdot)$ are, respectively, defined as:

$$\text{sign}^+(a) = \begin{cases} 1, & \text{if } a > 0, \\ \frac{1}{2}, & \text{if } a = 0, \\ 0, & \text{if } a < 0, \end{cases} \tag{4.6}$$

$$\text{sign}^-(a) = 1 - \text{sign}^+(a). \tag{4.7}$$

In [10] it is shown that the KBR is not idempotent. Therefore, instead of manually choosing the number of times the KBR should be applied to obtain the best enhanced image, it is better to rewrite Eq. (4.5) in the form of a partial differential equation (PDE): $I_t(x) = L(x, t) - I(x, t)$, into which we can introduce an attachment to data term that leaves the image unmodified once it has departed too much from the original image. We use this framework in the GM context by applying this PDE to the Saturation channel only, while keeping the Hue and Value channels fixed: we know that KBR will increase the Saturation values, which implies that the gamut will be extended. Taking Eq. (4.5) for the Saturation channel S and adding also a chroma term $C = SV$ (from [21]) we get:

$$\begin{aligned} S_t(x, t) =& \gamma \sum_y w(x, y) \left[f\left(\frac{S(x, t)}{S(y, t)} \right) \text{sign}^+(S(y, t) \right. \\ & \left. - S(x, t)) + \text{sign}^-(S(y, t) - S(x, t)) \right] \\ & - S(x, t) - \beta(S(x, t) - S_0(x)) \\ & - \tau(S_0(x))(S(x, t)V_0(x) - S_0(x)V_0(x)), \end{aligned} \tag{4.8}$$

where $\beta > 0$ controls the strength of the original data attachment term S_0 and γ is a positive constant. V_0 is the value component of the original image. In gamut extension, if all colors of an image are extended in the same manner (and by an equal amount), the result may appear unnatural. Therefore, to treat objects of low saturation and high saturation differently, we make use of the saturation-dependent weighting function $\tau(\cdot)$. As it is shown in Fig. 4.17, higher weights $\tau(\cdot)$ are attached to the low saturated input colors (such as skin tones, neutral colors, etc.) as we should apply to them little to no extension, whereas low weights are given to the high saturated (artificial objects) colors in order to extend them normally. To compute the weights for each pixel, we use the following adapted version of the generalised logistic function (https://en.wikipedia.org/wiki/Generalised_logistic_function):

$$\tau(S_0(x)) = \tau_{max} \left(1 - \frac{1}{1 + 0.55e^{-1.74S_0(x)^2}} \right) \tag{4.9}$$

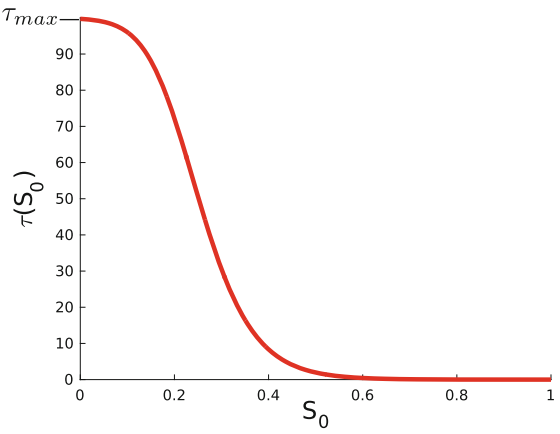


Fig. 4.17 Logistic function

where τ_{max} is a positive constant. The values used in Eq. (4.9) have been chosen based on tests we performed on several images with different color characteristics.

Let us now discretize the derivative and apply a forward-time numerical scheme on Eq. (4.8) as

$$\begin{aligned} S^{k+1}(x) =& \gamma \Delta t R_{S^k}(x) + S^k(x)[1 - \Delta t(1+\beta)] \\ &+ \Delta t[\beta S_0(x) + \tau(S_0(x))V_0(x)(S_0(x) - S^k(x))], \end{aligned} \tag{4.10}$$

where Δt is the time step and $k \in \mathbb{N}$ denotes the iteration number. The initial condition is $S^{k=0}(x) = S_0(x)$, and the function $R_{S^k}(x)$ indicates the contrast modification function:

$$\begin{aligned} R_{S^k}(x) = \sum_y w(x,y) \Bigg[f\left(\frac{S^k(x)}{S^k(y)}\right) \text{sign}^+(S^k(y) - S^k(x)) \\ + \text{sign}^-(S^k(y) - S^k(x)) \Bigg]. \end{aligned} \tag{4.11}$$

Following the work of [10] we can rewrite Eq. (4.10) as

$$S^{k+1}(x) = \frac{S^k(x) + \Delta t \left(S_0(x)(\beta + \tau(S_0(x))V_0(x)^2) + \frac{\gamma}{2} R_{S^k}(x)\right)}{1 + \Delta t(\beta + \tau(S_0(x))V_0(x)^2)} \tag{4.12}$$

Results of GEA-KBR

In this section we assess the image reproduction quality of our GEA-KBR and other algorithms [39] using the datasets of [3] and [22]. Given an RGB image, we first convert it into the HSV color space and apply gamut extension only on the saturation component using the evolution equation (4.12). The parameter values

Fig. 4.18 Results: mapping from Toy to sRGB gamut. Column 1: Input image. Column 2: HCM [39]. Column 3: SDS [39]. Column 4: Chroma extension [39]. Column 5: LCA [39]. Column 6: Our GEA-KBR. Original images are from [3] and [22]

that we use in Eq. (4.12) are $\beta = 1$, $\Delta t = 0.10$. The non-linear scaling function is $f(r) = A \log(r) + 1$, where $A = \frac{1}{\log(256)}$. The value for σ, the standard deviation for $w(x, y)$, is equal to one-third of the number of rows or columns of the input image (whichever is larger). The results that we present in this section are mapped from the 'Toy' gamut to the sRGB gamut (see primaries of gamuts in Table 4.1).

While emerging wide-gamut displays are capable of rendering vivid colors, it is extremely important to reproduce skin tones carefully so as to preserve the artistic intent of the content's creator. To compare how different GEAs reproduce flesh tones (always a key issue in movie postproduction), in Fig. 4.18 we present some results showing that our GEA-KBR extends skin tones in a controlled (limited) manner, but applies normal color extension to the artificial objects (see row 2 and row 4). Whereas in the same figure it can be seen that other methods such as same-drive signal (SDS) [39] and chroma extension [39] reproduce skin tones poorly and also have a problem of over-saturation. For better comparison, we show in Fig. 4.19 the zoomed-in view of regions cropped from Fig. 4.18. Since the lightness chroma adaptive (LCA) algorithm [39] works by modifying both lightness and chroma, it tends to produce images with artifacts, loss of saturation and loss of spatial detail. For instance, we can notice in Fig. 4.19 that the napkin in row 2 has some artifacts, and there is a loss of spatial details in the picture of the elephant in the last row. As shown in row 4 of Fig. 4.19, the chroma extension algorithm [39] reproduces high chromatic objects poorly as it has a problem of giving a strong chroma boost to colors.

It is reported in [20] that viewers prefer saturated colors, but this is only true when they are not aware of the original colors of objects: viewers tend to become very critical of color changes in those objects for which they have memory such as sky, grass, etc. Therefore, extra care should be taken while extending memory colors. In Fig. 4.18, row 1, it can be observed that the hybrid color mapping (HCM), and SDS algorithms produce images with an over-saturated grass region, whereas our GEA performs a controlled amount of extension and reproduces the same grass region accurately.

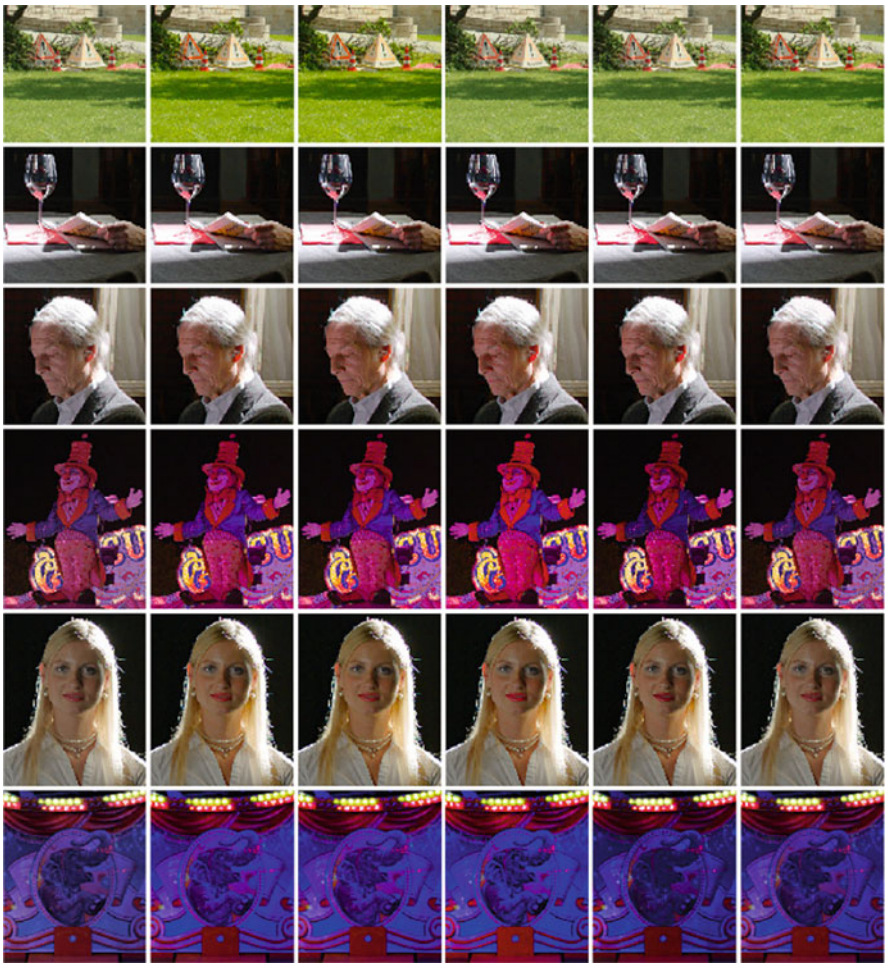


Fig. 4.19 Zoomed-in view of the regions cropped from Fig. 4.18. Column 1: Input image. Column 2: HCM [39]. Column 3: SDS [39]. Column 4: Chroma extension [39]. Column 5: LCA [39]. Column 6: Our GEA-KBR

Making the GEA-KBR Faster

The non-optimized MATLAB implementation of our GEA-KBR running on a desktop PC takes 11 s to process an image of resolution 656 × 1080 pixels, which means that the GEA-KBR is 25 times faster than the GEA-LAB2 (that we have presented in the previous section). However, by applying the following chain of operations (only on the saturation component) we can further reduce the computational cost to perform gamut extension using the GEA-KBR. All the stages

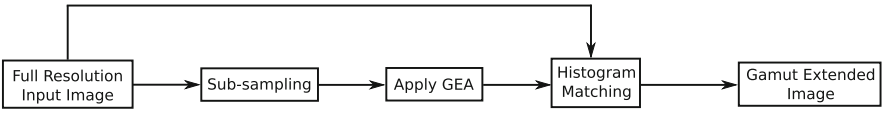


Fig. 4.20 A schematic to reduce the computational cost of the proposed framework

Fig. 4.21 Example of reducing computational time. (**a**) Input image. (**b**) Result of GEA-KBR applied on a full resolution image. (**c**) Result of GEA-KBR applied on a sub-sampled image. Original image is from [37]

of the procedure are shown in Fig. 4.20 and described as follows:

- **Sub-sampling:** the first step is to down-sample the full resolution image by some scaling factor. We recommend using the scaling factor of 0.40 as it provides a good trade-off between image quality and speed.
- **Apply GEA:** apply the proposed GEA-KBR on the sub-sampled image to obtain a reduced-size image with an extended gamut.
- **Histogram Matching:** finally perform histogram matching of the full resolution input image and the sub-sampled extended-gamut image in order to obtain the final full-resolution extended-gamut image.

Figure 4.21 shows that the aforementioned steps produce results having the same visual appearance as of applying the GEA-KBR directly on the full resolution image, but in just a fraction (25%) of the time.

GRA-KBR: Gamut Reduction Algorithm

This section is devoted to present a key modification in the framework of GEA-KBR that will allow us to perform gamut reduction. Let us note that the chroma term in

the PDE (Eq. (4.8)) was exclusively used to deal with the gamut extension problem in which we need to treat low-saturated and high-saturated colors differently. So by removing the effect of the chroma term from Eq. (4.8), i.e. by setting $\tau(\cdot)$ to zero, we obtain the following corresponding evolution equation

$$S^{k+1}(x) = \frac{S^k(x) + \Delta t \left(\beta S_0(x) + \frac{\gamma}{2} R_{S^k}(x)\right)}{1 + \beta \Delta t}, \tag{4.13}$$

and by using $\gamma < 0$ in this equation we can decrease the saturation of the input image and subsequently reduce the color gamut.

Results of GRA-KBR

Our GRA-KBR works iteratively and maps only the out-of-gamut colors to the smaller destination gamut, while leaving the in-gamut colors unmodified; this iterative process is the same as we have described for the GRA-RGB in section "GRA-RGB [76]: Gamut Reduction Algorithm on RGB".

To perform visual quality assessment we map colors of sRGB images to a challenging smaller 'Mock' gamut using our GRA-KBR and the GRAs of [1, 66], LCLIP [65] and HPMINDE [57]. The color primaries of the sRGB and Toy gamuts are given in Table 4.1. In the results presented in Fig. 4.22 it can be seen that our GRA-KBR works well in preserving hues and texture in the reduced-gamut images; and these reproductions are perceptually more faithful to the original images than those of other competing methods. For a better comparison we present close-ups in Fig. 4.23 where it is clearly noticeable in row 1 and row 4 that the results of HPMINDE have artifacts (loss of spatial detail) due to its inherent functionality of projecting two nearby out-of-gamut colors to far-away points on the destination gamut. The GRAs of Schweiger et al. [66] and LCLIP [65] may produce reduced-gamut images with excessive desaturation in bright regions, as shown in rows 1 and 3 of Fig. 4.23. The method of Alsam et al. [1] can over-compensate the contrast, as shown in the close-up of the yellow parrot. In the example of the second row of Fig. 4.23, all tested GRAs except the proposed one produce tonal discontinuities on the face of the woman.

Conclusion and Future Work

In this article we have presented a review of our work on variational methods for gamut mapping that comply with some basic global and local properties of the human visual system, producing results that appear natural and perceptually faithful to the original material. In Table 4.2 we list the characteristics and limitations of each of the presented methods. The following are some challenges in gamut mapping that still need to be addressed.

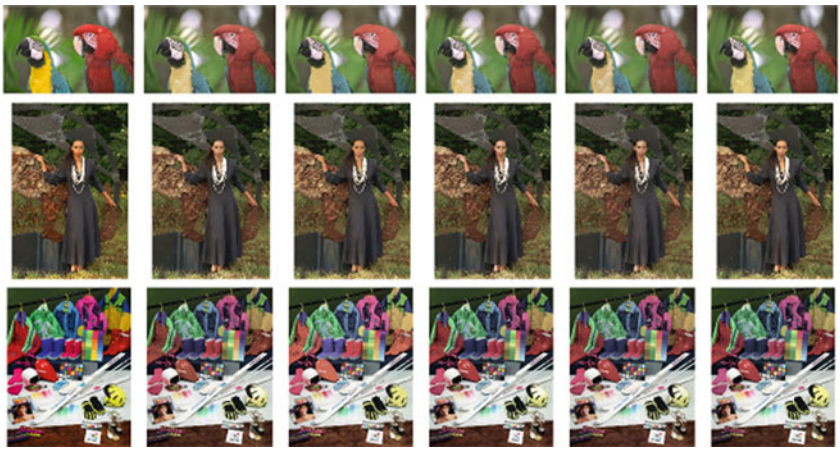


Fig. 4.22 Reproductions of GRAs. Column 1: input images. Column 2: LCLIP [65]. Column 3: HPMINDE [57]. Column 4: Schweiger et al. [66]. Column 5: Alsam et al. [1]. Column 6: Our GRA. The original image in the last row is from [17], while rest of the input images are from Kodak dataset [37]

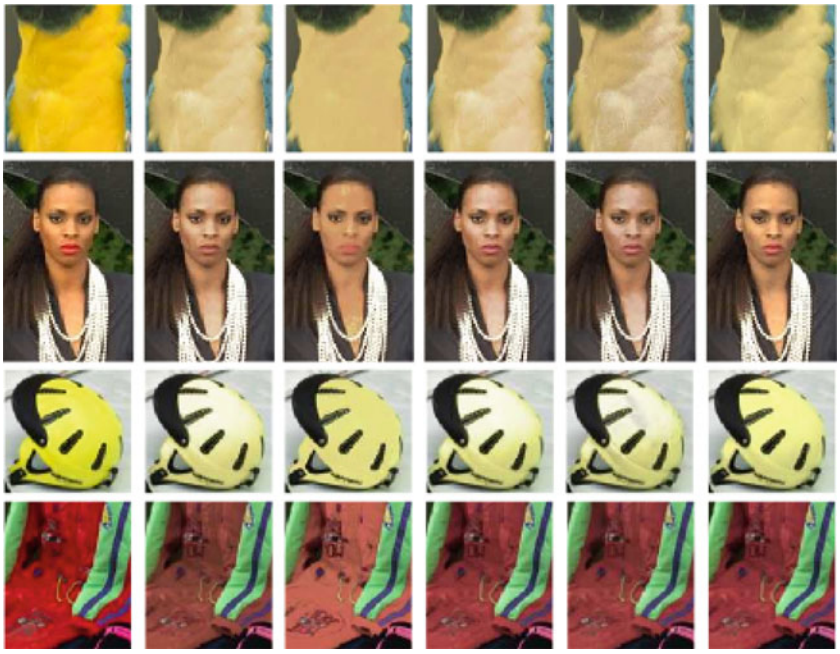


Fig. 4.23 Comparison of GRAs: croppings are from Fig. 4.22. Column 1: original cropped regions. Column 2: LCLIP [65]. Column 3: HPMINDE [57]. Column 4: Schweiger et al. [66]. Column 5: Alsam et al. [1]. Column 6: Our GRA-KBR

Table 4.2 Summary of the presented gamut mapping algorithms

GMAs	Characteristics	Limitations
GRA-RGB [76]	• Works in RGB color space • Apply on each color channel independently • Reduces gamut by reducing contrast • Iterative in nature	• May introduce spatial artifacts when difference between source and target gamut is large • Poor reproduction if the input image has low contrast • Computationally expensive
GEA-RGB [76]	• Works in RGB color space • Apply on each color channel independently • Extends gamut by increasing contrast • Non-iterative (one-shot) in nature	• May over-enhance contrast • May send a few colors towards black, leading to loss of saturation for high contrast input images • Computationally expensive
GEA-LAB1 [77]	• Works in CIELAB color space • Apply only on chromatic channels 'a' and 'b', while preserving lightness • Extends gamut by increasing contrast • Non-iterative (one-shot) in nature	• May introduce hue shifts • May send a few colors towards black, leading to loss of saturation for high contrast input images
GEA-LAB2 [79]	• Works in CIELAB color space • Apply only on chromatic channels 'a' and 'b', while preserving lightness • Extends gamut by increasing contrast • Iterative in nature • Added constraints to avoid hue shifts, loss of saturation and excessive chroma boost	• Wrong parameters selection may introduce false edges • Computationally expensive
GEA-KBR [78]	• Works in HSV color space • Apply only on saturation component, while preserving hue and value components • Extends gamut by increasing saturation • Uses logistic function to deal with low-saturated and high-saturated pixels differently • Non-iterative (one-shot) in nature	• Fixed contrast modification (γ) parameter for all input images, which may produce some results either under-enhanced or over-enhanced
GRA-KBR [80]	• Works in HSV color space • Apply only on saturation component, while preserving hue and value components • Reduces gamut by reducing saturation • Iterative in nature	• High computational cost

- Although current spatial GMAs are adaptive and flexible, at the same time they are more complex and computationally expensive than global GMAs. These local GMAs need to be fast so that they can yield color reproductions on the fly in live broadcasts. The development of fast spatial GMAs is also important if we want to implement them in the image processing pipeline of a camera.
- Our user study [79] showed that the current image quality metrics, when applied to the gamut extension problem, provide results that do not correlate well with users' choices. Therefore, there is the need to develop an error metric specifically for gamut extension. Without a suitable image quality metric, the gamut extension problem cannot be posed as an optimization procedure; moreover, we are forced to conduct psychophysical tests for evaluating GEAs but these subjective tests are cumbersome, time-consuming and expensive.

References

1. A. Alsam, I. Farup, Spatial colour gamut mapping by orthogonal projection of gradients onto constant hue lines, in *Proceedings of 8th International Symposium on Visual Computing* (2012), pp. 556–565
2. H. Anderson, E. Garcia, M. Gupta, Gamut expansion for video and image sets, in *International Conference on Image Analysis and Processing Workshops* (2007), pp. 188–191
3. S. Andriani, H. Brendel, T. Seybold, J. Goldstone, Beyond the Kodak image set: a new reference set of color image sequences, in *IEEE International Conference on Image Processing* (2013), pp. 2289–2293
4. R. Bala, R. Dequeiroz, R. Eschbach, W. Wu, Gamut mapping to preserve spatial luminance variations. J. Imaging Sci. Technol. **45**, 122–128 (2001)
5. D. Bankston, The color-space conundrum, part one. American Cinematographer (2005), p. 6
6. Z. Barańczuk, P. Zolliker, J. Giesen, Image quality measures for evaluating gamut mapping, in *Color and Imaging Conference* (2009), pp. 21–26
7. R.S. Berns, The mathematical development of CIE TC 1–29 proposed colour difference equation: CIELCH, in *Proceedings of the Seventh Congress of International Colour Association*, B, C19.1–19.4 (1993)
8. M. Bertalmío, *Image Processing for Cinema*, vol. 4 (CRC Press/Taylor & Francis, Boca Raton, 2014)
9. M. Bertalmío, V. Caselles, E. Provenzi, A. Rizzi, Perceptual color correction through variational techniques. IEEE Trans. Image Process. **16**(4), 1058–1072 (2007)
10. M. Bertalmío, V. Caselles, E. Provenzi, Issues about Retinex theory and contrast enhancement. Int. J. Comput. Vis. **83**(1), 101–119 (2009)
11. N. Bonnier, F. Schmitt, H. Brettel, S. Berche, Evaluation of spatial gamut mapping algorithms, in *Proceedings of IS&T/SID 14th Color Imaging Conference* (2006), pp. 56–61
12. G.J. Braun, A paradigm for color gamut mapping of pictorial images. Ph.D. thesis, Rochester Institute of Technology, Rochester, 1999
13. S.E. Casella, R.L. Heckaman, M.D. Fairchild, Mapping standard image content to wide-gamut displays, in *Color and Imaging Conference* (2008), pp. 106–111
14. X. Chen, Investigation of gamut extension algorithms. Master's thesis, University of Derby, Derby, 2002
15. H.-C. Cheng, I. Ben-David, S.-T. Wu, Five-primary-color LCDs. J. Disp. Technol. **6**(1), 3–7 (2010)

16. E. Chino, K. Tajiri, H. Kawakami, H. Ohira, K. Kamijo, H. Kaneko, S. Kato, Y. Ozawa, T. Kurumisawa, K. Inoue, K. Endo, H. Moriya, T. Aragaki, K. Murai, Development of wide-color-gamut mobile displays with four-primary-color LCDs. SID Symp. Dig. Tech. Pap. **37**(1), 1221–1224 (2006)
17. CIE, Guidelines for the evaluation of gamut mapping algorithms. Technical report, CIE 156 (2004)
18. F. Dugay, I. Farup, J.Y. Hardeberg, Perceptual evaluation of color gamut mapping algorithms. Color Res. Appl. **33**(6), 470–476 (2008)
19. A.M. Eskicioglu, P.S. Fisher, Image quality measures and their performance. IEEE Trans. Commun. **43**(12), 2959–2965 (1995)
20. E.A. Fedorovskaya, H. de Ridder, F.J.J. Blommaert, Chroma variations and perceived quality of color images of natural scenes. Color Res. Appl. **22**(2), 96–110 (1997)
21. A. Ford, A. Roberts, Colour space conversions. http://www.poynton.com/PDFs/coloureq.pdf (1998)
22. J. Froehlich, S. Grandinetti, B. Eberhardt, S. Walter, A. Schilling, H. Brendel, Creating cinematic wide gamut HDR-video for the evaluation of tone mapping operators and HDR-displays, in *Proceedings of IS&T/SPIE Electronic Imaging* (2014)
23. R.S. Gentile, E. Walowitt, J.P. Allebach, A comparison of techniques for color gamut mismatch compensation. J. Imaging Technol. **16**, 176–181 (1990)
24. J.Y. Hardeberg, E. Bando, M. Pedersen, Evaluating colour image difference metrics for gamut-mapped images. Color. Technol. **124**(4), 243–253 (2008)
25. R.L. Heckaman, J. Sullivan, Rendering digital cinema and broadcast TV content to wide gamut display media. SID Symp. Dig. Tech. Pap. **42**(1), 225–228 (2011)
26. P.G. Herzog, M. Müller, Gamut mapping using an analytical color gamut representation, in *Proceedings of Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts* (1997), pp. 117–128
27. T. Hoshino, A preferred color reproduction method for the HDTV digital still image system, in *Proceedings of IS&T Symposium on Electronic Photography* (1991), pp. 27–32
28. T. Hoshino, Color estimation method for expanding a color image for reproduction in a different color gamut, May 1994. US Patent 5,317,426
29. ITU-R Recommendation BT.709-5, Parameter values for the HDTV standards for production and international programme exchange (2002)
30. ITU-R Recommendation BT.2020, Parameter values for ultra high definition television systems for production and international programme exchange (2012)
31. A.J. Johnson, Perceptual requirements of digital picture processing. Paper Presented at IARAIGAI Symposium and Printed in Part in Printing World (1979)
32. B.H. Kang, J. Morovič, M.R. Luo, M.S. Cho, Gamut compression and extension algorithms based on observer experimental data. ETRI J. **25**(3), 156–170 (2003)
33. N. Katoh, M. Ito, Gamut mapping for computer generated images (ii), in *Proceedings of 4th IS&T/SID Color Imaging Conference* (1996), pp. 126–129
34. G. Kennel, *Color and Mastering for Digital Cinema: Digital Cinema Industry Handbook Series* (Taylor & Francis, New York, 2007)
35. M.C. Kim, Y.C. Shin, Y.R. Song, S.J. Lee, I.D. Kim, Wide gamut multi-primary display for HDTV, in *Proceedings of 2nd European Conference on color Graphics, Imaging and Vision* (2004), pp. 248–253
36. R. Kimmel, D. Shaked, M. Elad, I. Sobel, Space-dependent color gamut mapping: a variational approach. IEEE Trans. Image Process. **14**, 796–803 (2005)
37. Kodak, http://r0k.us/graphics/kodak/ (1993)
38. Y. Kusakabe, Y. Iwasaki, Y. Nishida, Wide-color-gamut super hi-vision projector, in *Proceedings ITE Annual Convention (in Japanese)* (2013)
39. J. Laird, R. Muijs, J. Kuang, Development and evaluation of gamut extension algorithms. Color Res. Appl. **34**(6), 443–451 (2009)
40. C. Lau, W. Heidrich, R. Mantiuk, Cluster-based color space optimizations, in *Proceedings of IEEE International Conference on Computer Vision, ICCV '11* (2011), pp. 1172–1179

41. Y. Li, G. Song, H. Li, A multilevel gamut extension method for wide gamut displays, in *Proceedings of International Conference on Electric Information and Control Engineering (ICEICE)* (2011), pp. 1035–1038
42. Y. Ling, Investigation of a gamut extension algorithm. Master's thesis, University of Derby, Derby, 2001
43. I. Lissner, J. Preiss, P. Urban, M.S. Lichtenauer, P. Zolliker, Image-difference prediction: from grayscale to color. IEEE Trans. Image Process. **22**(2), 435–446 (2013)
44. Y. Liu, G. Song, H. Li, A hue-preserving gamut expansion algorithm in CIELUV color space for wide gamut displays, in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP)* (2010), pp. 2401–2404
45. M.R. Luo, G. Cui, B. Rigg, The development of the CIE 2000 colour-difference formula: CIEDE2000. Color Res. Appl. **26**(5), 340–350 (2001)
46. G. Marcu, S. Abe, Gamut mapping for color simulation on CRT devices, in *Proceedings of Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts* (1996)
47. K. Masaoka, Y. Kusakabe, T. Yamashita, Y. Nishida, T. Ikeda, M. Sugawara, Algorithm design for gamut mapping from UHDTV to HDTV. J. Disp. Technol. **12**(7), 760–769 (2016)
48. J.J. McCann, Lessons learned from mondrians applied to real images and color gamuts, in *Proceedings of Color Imaging Conference* (1999), pp. 1–8
49. J.J. McCann, A spatial colour gamut calculation to optimize colour appearance, in *Colour Image Science: Exploiting Digital Media* (2002), pp. 213–233
50. X. Meng, G. Song, H. Li, A human skin-color-preserving extension algorithm for wide gamut displays, in *Proceedings of International Conference on Information Technology and Software Engineering*. Lecture Notes in Electrical Engineering (Springer, Berlin, 2013), pp. 705–713
51. J. Meyer, B. Barth, Color gamut matching for hard copy, in *Proceedings of SID Digest* (1989), pp. 86–89
52. E.D. Montag, M.D. Fairchild, Psychophysical evaluation of gamut mapping techniques using simple rendered images and artificial gamut boundaries. IEEE Trans. Image Process. **6**(7), 977–989 (1997)
53. J. Morovič, To develop a universal Gamut mapping algorithm. Ph.D. thesis, University of Derby, Derby, 1998
54. J. Morovič, *Color Gamut Mapping*, vol. 10 (Wiley, Chichester, 2008)
55. J. Morovič, Y. Wang, A multi-resolution, full-colour spatial gamut mapping algorithm, in *Proceedings of Color Imaging Conference* (2003), pp. 282–287
56. R. Muijs, J. Laird, J. Kuang, S. Swinkels, Subjective evaluation of gamut extension methods for wide-gamut displays, in *Proceedings of the 13th International Display Workshop* (2006), pp. 1429–1432
57. G.M. Murch, J.M. Taylor, Color in computer graphics: manipulating and matching color, in *Eurographics Seminar: Advances in Computer Graphics V* (1989), pp. 41–47
58. S. Nakauchi, S. Hatanaka, S. Usui, Color gamut mapping based on a perceptual image difference measure. Color Res. Appl. **24**(4), 280–291 (1999)
59. H. Pan, S. Daly, A gamut-mapping algorithm with separate skin and non-skin color preference controls for wide-color-gamut TV. SID Symp. Dig. Tech. Pap. **39**(1), 1363–1366 (2008)
60. M.R. Pointer, The gamut of real surface colours. Color Res. Appl. **5**(3), 145–155 (1980)
61. C. Poynton, Contrast, brightness, and the naming of things. Poynton's Vector 1 (2010)
62. J. Preiss, P. Urban, Image-difference measure optimized gamut mapping, in *Proceedings of IS&T/SID 20th Color Imaging Conference* (2012), pp. 230–235
63. J. Preiss, F. Fernandes, P. Urban, Color-image quality assessment: from prediction to optimization. IEEE Trans. Image Process. **23**(3), 1366–1378 (2014)
64. S. Roth, I. Ben-David, M. Ben-Chorin, D. Eliav, O. Ben-David, Wide gamut, high brightness multiple primaries single panel projection displays. SID Symp. Dig. Tech. Pap. **34**(1), 118–121 (2003)
65. J.J. Sara, The automated reproduction of pictures with nonreproducible colors. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, 1984

66. F. Schweiger, T. Borer, M. Pindoria, Luminance-preserving colour conversion, in *SMPTE Annual Technical Conference and Exhibition* (2016), pp. 1–9
67. B.D. Silverstein, A.F. Kurtz, J.R. Bietry, G.E. Nothhard, A laser-based digital cinema projector. SID Symp. Dig. Tech. Pap. **42**(1), 326–329 (2011)
68. SMPTE RP 431-2:2011, D-cinema quality – reference projector and environment (2011)
69. G. Song, H. Cao, H. Huang, Hue preserving multi-level expansion method based on saturation for wide gamut displays. J. Inf. Comput. Sci. **11**(2), 461–472 (2014)
70. G. Song, X. Meng, H. Li, Y. Han, Skin color region protect algorithm for color gamut extension. J. Inf. Comput. Sci. **11**(6), 1909–1916 (2014)
71. J.M. Taylor, G.M. Murch, P. McManus, Tektronix HVC: a uniform perceptual color system for display users, in *SID Symposium Digest of Technical Papers* (1989)
72. W.S. Torgerson, A law of categorical judgment, consumer behaviour, in *Consumer Behaviour* (New York University Press, New York, 1954), pp. 92–93
73. S. Ueki, K. Nakamura, Y. Yoshida, T. Mori, K. Tomizawa, Y. Narutaki, Y. Itoh, K. Okamoto, Five-primary-color 60-inch LCD with novel wide color gamut and wide viewing angle. SID Symp. Dig. Tech. Pap. **40**(1), 927–930 (2009)
74. UGRA, UGRA GAMCOM version 1.1: Program for the color gamut compression and for the comparison of calculated and measured values. Technical report, UGRA, St. Gallen, 17 July 1995
75. Y.-C. Yang, K. Song, S.G. Rho, N.-S. Rho, S.J. Hong, K.B. Deul, M. Hong, K. Chung, W.H. Choe, S. Lee, C.Y. Kim, S.-H. Lee, H.-R. Kim, Development of six primary-color LCD. SID Symp. Dig. Tech. Pap. **36**(1), 1210–1213 (2005)
76. S. W. Zamir, J. Vazquez-Corral, M. Bertalmío, Gamut mapping in cinematography through perceptually-based contrast modification. IEEE J. Sel. Top. Sign. Process. **8**(3), 490–503 (2014)
77. S.W. Zamir, J. Vazquez-Corral, M. Bertalmío, Gamut extension for cinema: psychophysical evaluation of the state of the art, and a new algorithm, in *Proceedings of IS&T/SPIE Electronic Imaging* (2015), pp. 1–12
78. S.W. Zamir, J. Vazquez-Corral, M. Bertalmío, Perceptually-based gamut extension algorithm for emerging wide color gamut display and projection technologies, in *SMPTE Annual Technical Conference and Exhibition* (2016), pp. 1–11
79. S.W. Zamir, J. Vazquez-Corral, M. Bertalmío, Gamut extension for cinema. IEEE Trans. Image Process. **26**(4), 1595–1606 (2017)
80. S.W. Zamir, J. Vazquez-Corral, M. Bertalmío, Gamut reduction through local saturation reduction, in *Color and Imaging Conference* (2017), pp. 214–218
81. P. Zolliker, K. Simon, Retaining local image information in gamut mapping algorithms. IEEE Trans. Image Process. **16**(3), 664–672 (2007)

Chapter 5
Functional Lifting for Variational Problems with Higher-Order Regularization

Benedikt Loewenhauser and Jan Lellmann

Abstract Variational approaches are an established paradigm in the field of image processing. The non-convexity of the functional can be addressed by functional lifting and convex relaxation techniques, which aim to solve a convex approximation of the original energy on a larger space. However, so far these approaches have been limited to first-order, gradient-based regularizers such as the total variation. In this work, we propose a way to extend functional lifting to a second-order regularizer derived from the Laplacian. We prove that it can be represented efficiently and thus allows numerical optimization. We experimentally demonstrate the usefulness on a synthetic convex denoising problem and on synthetic as well as real-world image registration problems.

Introduction and Related Work

In this work, we consider variational energy minimization problems of the form

$$\inf_{u:\Omega\to\Gamma} \int_{\Omega} \rho(x, u(x))\, dx + \lambda S(u), \tag{5.1}$$

for estimating some unknown data u defined on an open, bounded, connected—usually rectangular—image domain $\Omega \subseteq \mathbb{R}^d$ with values in $\Gamma \subseteq \mathbb{R}^n$. The data term in (5.1) is of integral form, with the integrand $\rho(x, u(x))$ typically depending

B. Loewenhauser
Institute of Mathematics and Image Computing (MIC), University of Lübeck, Lübeck, Germany

Technical University of Munich, Garching, Germany
e-mail: benedikt.loewenhauser@in.tum.de

J. Lellmann (✉)
Institute of Mathematics and Image Computing (MIC), University of Lübeck, Lübeck, Germany
e-mail: jan.lellmann@uni-luebeck.de

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_5

on some noisy, corrupted measurements. We are particularly interested in the case where ρ is *non-convex* in $u(x)$.

The regularizer S, weighted by a parameter $\lambda > 0$, encodes prior knowledge in order to account for randomness and is often used to resolve ambiguities and render the problem well-posed.

A classical convex example is the Rudin-Osher-Fatemi model with

$$\rho(x, u(x)) := \frac{1}{2}(u(x) - g(x))^2 \quad \text{and} \quad S(u) := \mathrm{TV}(u), \tag{5.2}$$

which can be used to remove noise from a given image $g : \Omega \to \Gamma$ while preserving discontinuities [35]. The *total variation* $\mathrm{TV}(u)$ is defined as the integral

$$\mathrm{TV}(u) := \int_\Omega d\|Du\|, \tag{5.3}$$

where the vector-valued Radon measure Du is used to represent the distributional derivative of u in order to allow for discontinuities [1, 41]. For (weakly) differentiable u, the total variation assumes the simpler form

$$\mathrm{TV}(u) = \int_\Omega \|\nabla u(x)\|_2 dx. \tag{5.4}$$

As we will be mainly focused on the discretized setting, we will restrict ourselves to the regular case and use the more suggestive notation (5.4).

In the ROF model, as ρ is convex, computing a global minimizer of (5.1) numerically is feasible even for large problems [4]. However, in many applications, one cannot assume convexity. As a prime example, consider the problem of *image registration [24],* also sometimes referred to as *large-displacement optical flow*: one starts with two images $R, T : \Omega \to \mathbb{R}$ and aims to find a *deformation*, also called *displacement*, $u : \Omega \to \mathbb{R}^d$ which is "sufficiently regular" and aligns R and T in the sense that

$$R(x) \approx T(x + u(x)) \tag{5.5}$$

for all $x \in \Omega$. A suitable energy is

$$\frac{1}{2}\int_\Omega (R(x) - T(x + u(x))^2 dx + \lambda S(u). \tag{5.6}$$

This data term is also referred to as sum-of-squares distance (SSD) [25].

Numerically minimizing (5.6) is a challenging problem: not only is the data term generally non-convex, the degree of non-convexity is also completely determined by the data R and T, which are generally noisy and result in an energy landscape with many local minima.

Typical methods for minimizing (5.6) therefore rely on local solvers such as gradient-based and (Quasi-)Newton methods; see [30] for an algorithmic overview. In the context of *optical flow* [16], the classical, and still most common, method is to linearize T around a current estimate, which renders the problem convex. These approaches suffer from the typical issue of local non-convex optimization: the algorithm can get stuck in local minima and requires a good initial estimate. Much work has been dedicated to finding such a starting point, such as "warping" and coarse-to-fine strategies [3].

Non-convexity also appears in much simpler settings, such as q-(pseudo-)norm denoising with energies of the form

$$\int_\Omega |u(x) - g(x)|^q dx + \lambda S(u), \tag{5.7}$$

with $q < 1$. This choice of q makes the method more robust against outliers in the data g, as the influence of outliers diminishes as $q \to 0$. Choosing $q < 1$ also encourages the sparsity of the argument more than convex variants with $q \geq 1$; this is a particularly useful feature in the context of sparse representation [11]. See also [29] for an extensive analysis of non-convex regularization. However, it again renders the data term non-smooth and non-convex. A recent development is to modify methods for non-smooth convex optimization to the non-convex setting [26, 27], however these are again local and convergence results are currently very limited.

Computational and algorithmic advances have recently made another strategy viable: Instead of solving the non-convex problem directly, one aims to approximate it by a—usually much larger—*convex* one, which can be solved to a global optimum. In order to approximate the original problem well, one relies on *functional lifting*, i.e., embedding the original problem into a much larger space: Instead of solving

$$\inf_{u:\Omega\to\Gamma} f(u), \tag{5.8}$$

one solves the *lifted problem*

$$\inf_{\bar{u}:\Omega\to\mathscr{P}(\Gamma)} \bar{f}(\bar{u}), \tag{5.9}$$

where $\mathscr{P}(\Gamma)$ is the set of *probability measures* over the range Γ, and $\bar{f}$ is a suitable extension of f on this larger function space in the following sense: with each $u : \Omega \to \Gamma$, one can associate a function $\bar{u} : \Omega \to \mathscr{P}(\Gamma)$ which is a Dirac measure at every point, $\bar{u}(x) := \delta_{u(x)}$, and require that $\bar{f}(\bar{u}) = f(u)$ for all u in the original space. On the other hand, if the solution of (5.9) is a Dirac measure $\delta_{u(x)}$ at every point and $\bar{f}$ does not introduce artificial minimizers, then u will be a solution of the original problem (5.8), as each element of the feasible set of (5.8) has a corresponding element in the feasible set of (5.9).

This leaves the question of how to define $\bar{f}$ on arguments $\bar{u}$ that are not Dirac measures, but rather mixtures or even diffuse measures. There is a series of publications discussing different strategies for deriving "good" liftings, starting with image segmentation [20, 32, 38, 39], general convex first-order regularization [33] as a functional-analytic formulation of the classical paper [17], recently advancing the framework to manifold-valued problems [21] and more accurate discretizations [18, 28, 40].

However, all these works assume that the regularizer depends only on the first-order derivative ∇u or its distributional counterpart. For natural images, such first-order regularization is often sub-optimal, as it penalizes linear parts and, in the case of TV, prefers -wise constant solutions.

For natural images, regularizers that use second- and higher-order derivatives have been found to be much more suitable [2, 7, 22, 23, 31, 36]. Therefore one would like to use these more advanced regularizers in the functional lifting framework. However, so far there has been little progress in this direction. The reason is that the space of probability measures $\mathscr{P}(\Gamma)$ is usually discretized as a discrete probability measure on ℓ chosen points in the range Γ. If one follows the same strategy as for lifting TV, one ends up with a large number of constraints on the dual variables, which is at least cubic with respect to ℓ. This requires to choose ℓ very small, which corresponds to a very rough discretization of the range Γ of u and brings the accuracy below acceptable thresholds.

Contributions

In this work, we propose a method for approximating energies of the form (5.1) using functional lifting and convex relaxation, where ρ is a general non-convex data term and the regularizer S incorporates *second-order* information:

- We investigate the non-smooth "Absolute Laplacian" regularizer, which incorporates second-order derivatives and coincides with TV^2 on one-dimensional domains (section "Lifting for Absolute Laplacian Regularization").
- After reviewing mathematical prerequisites (section "Notation and Mathematical Preliminaries") and the discretized version of the problem, we discuss where the usual strategy for computing a convex extension of the regularizer fails for more involved regularizers (section "Approximate Relaxation of the Absolute Laplacian").

  We prove that by introducing an approximation step, the number of required constraints can be reduced from cubic (ℓ^3) to linear (ℓ) growth in the one-dimensional case (Theorem 5.1). We propose an extension to the case $d \geq 2$, which—although currently without theoretical guarantees—has been very successful in all of our experiments.
- In order to show that a non-convex data term combined with higher-order regularization has practical benefits, we illustrate the method on a synthetical

q-pseudo-norm denoising example as in (5.7) with second-order regularization (section "Non-convex Denoising with Second-Order Regularity").
- We demonstrate the applicability to the non-convex problem of image registration as in (5.6) (section "Image Registration Using the Absolute Laplacian").

We conclude with an outlook and notes on further open questions (section "Conclusion and Outlook").

Lifting for Absolute Laplacian Regularization

In the following, we will consider a special case of second-order regularization: For $u = (u_1, \dots, u_n) : \Omega \to \mathbb{R}^n$, we define the *absolute Laplacian* regularizer

$$S_{AL}(u) := \int_\Omega \|\Delta u(x)\|_1 \, dx. \tag{5.10}$$

By convention the Laplacian $\Delta u := (\Delta u_1, \dots, \Delta u_n)^\top$ is vector-valued for $n > 1$.

Similar to the total variation, S_{AL} can be extended to functions with distributional Laplacians as well using a dual formulation; it can also be viewed as the set of functions with a gradient of bounded deformation [37]. Again we will focus on the discretized energy and therefore use the simplified notation (5.10).

The absolute Laplacian regularizer (5.10) has some drawbacks: most importantly, it is not isotropic in the sense that $S_{AL}(u) = S_{AL}(Ru)$ for some rotation matrix $R \in \mathbb{R}^{n\times n}$, and it has a large kernel that includes all harmonic functions. The latter issue was also discussed in detail in [14] for quadratic Laplacian regularization.

It is tempting to substitute a full Hessian regularization such as [9, 15, 23]

$$\int_\Omega \left(\sum_{i=1}^n \|\nabla^2 u_i(x)\|_2^2 \right)^{\frac{1}{2}} dx, \tag{5.11}$$

however this couples all components of u, which invalidates the argument used in the proof of Theorem 5.1 below. As of now, we have not found a way for efficiently computing a convex relaxation in the full Hessian-regularized case.

In contrast, the absolute Laplacian (5.10) decouples in the components u_i. Moreover, in the one-dimensional scalar case with $d = 1$ and $n = 1$, it is identical to the second-order total variation [9],

$$S_{AL}(u) = \int_\Omega |u''(x)| \, dx \tag{5.12}$$

or its distributional equivalent.

The absolute Laplacian is motivated by a regularizer that is—in a slightly loose interpretation of the term—known as "curvature" regularization in the medical

image registration community [10] and penalizes the squared Laplacians $\|\Delta u(x)\|_2^2$ instead of $\|\Delta u(x)\|_1$. However, as we will see in the next sections, the 1-homogeneous nature of S_{AL} is crucial in order to accurately lift the regularizer.

Notation and Mathematical Preliminaries

In the following, we detail the discretized lifting approach. We follow the notation in [28]. In order to discretize the probability measures $\mathscr{P}(\Gamma)$, we choose an n-dimensional regular grid of points $\{t_1, \dots, t_\ell\} \subseteq \Gamma$, which are referred to as *labels*. The number of labels in each dimension of the range Γ is denoted by l_k, $k = 1, \dots, n$, and the grid spacing h is assumed to be uniform and constant.

The space $\mathscr{P}(\Gamma)$ is discretized as the unit simplex in $\mathbb{R}^\ell$,

$$\Delta_\ell := \{\bar{p} \in \mathbb{R}^\ell | \bar{p} \geq 0, \ \sum_{i=1}^{\ell} \bar{p}_i = 1\}. \tag{5.13}$$

In a slight abuse of notation, we will from now on denote by $\bar{u}$ a function mapping into the set of *discretized* probability measures, i.e., $\bar{u} : \Omega \to \Delta_\ell$. The i-th unit vector $e_i \in \Delta_\ell$, $i \in \{1, \dots, \ell\}$, is associated with the Dirac measure δ_{t_i} at label t_i. Rather than associating a general vector $\bar{u}(x) \in \Delta_\ell$ with a weighted sum of Dirac measures as is commonly done, we assign to each vector a *single* Dirac measure $\delta_{u(x)}$, where $u(x) \in \Gamma$ is obtained by linear weighting of the labels:

$$u(x) = \sum_{i=1}^{\ell} \bar{u}_i(x) t_i. \tag{5.14}$$

Whenever (5.14) holds, we refer to $\bar{u}(x) \in \mathbb{R}^\ell$ as a *lifted representation* of $u(x) \in \Gamma$. A function $\bar{u} : \Omega \to \Delta_\ell$ is called a lifted representation of the function $u : \Omega \to \Gamma$ if (5.14) holds point-wise for all $x \in \Omega$.

Approximate Relaxation of the Absolute Laplacian

In order to illustrate the basic process of constructing an energy function for the lifted representation, first consider the data term in integral form:

$$F(u) := \int_\Omega \rho(x, u(x))\, dx. \tag{5.15}$$

We discretize $\mathscr{P}(\Gamma)$ as in the previous section, and seek a suitable convex extension of F,

$$\bar{F}(\bar{u}) := \int_\Omega \bar{\rho}(x, \bar{u}(x))\, dx, \tag{5.16}$$

to all $\bar{u} : \Omega \to \Delta_\ell$. A classical way [6] is to find the *largest convex* $\bar{\rho} : \Omega \times \Delta^\ell \to \mathbb{R}$ such that

$$\bar{\rho}(x, e_i) = \rho(x, t_i), \quad i = 1, \dots, \ell. \tag{5.17}$$

In order to do so for some fixed x, one first defines a function

$$\phi(p) := \begin{cases} \rho(x, t_i), & \text{if } p = e_i, \\ +\infty, & \text{otherwise,} \end{cases} \tag{5.18}$$

and sets $\bar{\rho}(x, p) := \phi^{**}(p)$, where ϕ^{**} is the Legendre-Fenchel biconjugate [34]. More precisely,

$$\phi^*(f) := \sup_p \{\langle p, f\rangle - \phi(p)\} = \max_{i \in \{1,\dots,\ell\}} \{\langle e_i, f\rangle - \rho(x, t^i)\}, \tag{5.19}$$

$$\phi^{**}(p) := \sup_f \{\langle p, f\rangle - \phi^*(f)\}. \tag{5.20}$$

As can be seen from (5.19), even for integrands ρ that depend only on a single value $u(x)$, the conjugate is generally composed of ℓ pieces. Using common first-order solvers, this incurs a cost of ℓ dual or auxiliary variables per point.

For the regularizer, this issue is much worse: Assume $\Omega \subseteq \mathbb{R}$, then the Laplacian of u at a point x is simply the second derivative and commonly discretized as

$$u''(x) \approx (u(x - \eta) - 2u(x) + u(x + \eta))/\eta^2, \tag{5.21}$$

which depends on *three* different values of u. A finite difference-based second-order regularizer will therefore be of the form

$$\int_\Omega \rho(u(x - \eta), u(x), u(x + \eta))\, dx, \tag{5.22}$$

which results in three running indices in (5.19) and thus ℓ^3 terms in the maximum. Even for a very moderate choice of $\ell = 10$, this results in 1000 additional variables per point, which is impractical.

In this section, we therefore consider an approximation of this process for the special case of the absolute Laplacian regularizer (5.10), which only requires *linear*

complexity. We derive the model for the one-dimensional case $d = 1$ and $n = 1$,

$$\int_\Omega |u''(x)|dx, \tag{5.23}$$

and subsequently discuss how to generalize it to n-dimensional image domains and vector-valued u.

The first step is to separate computation of the second derivatives from the lifting process, i.e., we also apply the derivative operator to the *lifted* representation $\bar{u}$ and seek a lifted regularizer

$$\int_\Omega \bar{\rho}(\bar{u}''(x))dx \approx \int_\Omega \bar{\rho}\left((\bar{u}(x+\eta) - 2\bar{u}(x) + \bar{u}(x-\eta))/\eta^2\right)dx, \tag{5.24}$$

where $x \pm \eta$ are the neighboring points of x. For simplicity, we assume $\eta = 1$.

We apply the same process as in (5.18) to $\rho(z) = |z|$ and set

$$\phi(p) = \begin{cases} |\mu| \cdot \left|t_{i_1} - 2t_{i_0} + t_{i_2}\right|, & \text{if } p = \mu \cdot (e_{i_1} - 2e_{i_0} + e_{i_2}), \\ +\infty, & \text{otherwise,} \end{cases} \tag{5.25}$$

where $1 \le i_0, i_1, i_2 \le \ell$. The free variable $\mu \in \mathbb{R}$ is not required, but ensures that ϕ is positively homogeneous. This implies that the conjugate ϕ is an indicator function of some set, which simplifies the later optimization. Taking the convex conjugate, we obtain

$$\phi^*(f) = \delta_{K_{1D}}(f) := \begin{cases} 0, & f \in K_{1D}, \\ +\infty, & \text{otherwise,} \end{cases} \tag{5.26}$$

with the set

$$K_{1D} := \bigcap_{1 \le i_0, i_1, i_2 \le \ell} \left\{ f \in \mathbb{R}^\ell : f_{i_1} - 2f_{i_0} + f_{i_2} \le h\,|i_1 - 2i_0 + i_2| \right\}. \tag{5.27}$$

This is a straightforward computation following from the definition of the convex conjugate and making use of the 1-homogeneity of ϕ, and using the assumption that the labels t_i are uniformly spaced with distance h. The above formulation consists of ℓ^3 constraints, which would render the problem numerically intractable except for very small ℓ.

A main contribution of this work is the following theorem, which shows that the number of constraints can be reduced to linear order.

Theorem 5.1 *The set K_{1D} in (5.27) with ℓ^3 linear constraints can be equivalently represented by ℓ linear constraints:*

$$K_{1D} = \left\{ f \in \mathbb{R}^\ell : f_2 - f_1 \leq h, \quad f_\ell - f_{\ell-1} \geq -h \right\} \cap \bigcap_{2 \leq i \leq \ell-1} \left\{ f \in \mathbb{R}^\ell : f_{i-1} - 2f_i + f_{i+1} \leq 0 \right\}. \tag{5.28}$$

Proof Denoting the right-hand side in (5.28) by K_{1D}^{red}, and using the definition of K_{1D} in (5.27), we have to show that $K_{1D} = K_{1D}^{red}$.

$K_{1D} \subseteq K_{1D}^{red}$

Assume $f \in K_{1D}$ as in (5.27), i.e., $f_{i_1} - 2f_{i_0} + f_{i_2} \leq h|i_1 - 2i_0 + i_2|$ holds for all triples $i_0, i_1, i_2 \in \{1, \ldots, \ell\}$. Choose $i_1 = i_2 = 2$ and $i_0 = 1$, then the first inequality in (5.28) follows. Analogously we obtain the second inequality $f_\ell - f_{\ell-1} \geq -h$ by setting $i_1 = i_2 = \ell - 1$ and $i_0 = \ell$. All other inequalities in (5.28) follow by setting $i_1 = i - 1, i_0 = i, i_2 = i + 1$, therefore $f \in K_{1D}^{red}$.

$K_{1D} \supseteq K_{1D}^{red}$

Suppose $f \in K_{1D}^{red}$, i.e., the inequalities in (5.28) hold. We define the vector $a \in \mathbb{R}^{\ell-1}$, $a_i := f_{i+1} - f_i$ as the difference between two consecutive components of f. Using this notation, we reformulate the constraints (5.28) in terms of a:

$$a_1 \leq h, \tag{5.29}$$

$$a_{\ell-1} \geq -h, \tag{5.30}$$

$$a_{i-1} \geq a_i, \ \forall i \in \{2, \ldots, \ell - 1\}. \tag{5.31}$$

Thus the components of a_i form a finite, monotonously non-increasing sequence that is absolutely bounded by h, i.e., $a \in S := \{x \in [-h, +h]^{\ell-1} : x_i \geq x_{i+1}\}$.

If $i_0 = i_1 = i_2$, the inequality in (5.27) holds trivially. Otherwise, if two of the indices agree, then the inequality in (5.27) takes the form

$$f_j - f_k \leq h|j - k|. \tag{5.32}$$

Assuming without loss of generality that $j > k$, this inequality follows from

$$f_j - f_k = a_k + \ldots + a_{j-1} \leq |a_k| + \ldots + |a_{j-1}| \leq h|j - k| \tag{5.33}$$

due to the observation that all a_i are bounded by $\pm h$.

We are left with the last case of distinct i_0, i_1, i_2. Without loss of generality assume $i_1 > i_2$, otherwise we swap the symbols.

As all inequalities are invariant with respect to the addition of a constant to f, it suffices to prove the claim for all f with f_1 fixed to some constant. Therefore we can assume $f_1 = 0$. Under this assumption, the linear map between vectors $f \in \mathbb{R}^\ell$

in K_{1D}^{red} and vectors $a \in \mathbb{R}^{\ell-1}$ satisfying (5.29)–(5.31) is bijective. As the vertices of the latter set consist of the vectors of the form $(h, \dots, h, -h, \dots, -h)$, from bijectivity we deduce that the vertices of the set $K_{1D}^{red} \cap \{f \,|\, f_1 = 0\}$ are the elements satisfying the *equality* $|f_{i+1} - f_i| = h$ and the inequality $f_{i-1} - 2f_i + f_{i+1} \leq 0$.

Showing that all f satisfying (5.28) are contained in the set in (5.27) is equivalent to showing

$$\max_{f \in K_{1D}^{red} \cap \{f \,|\, f_1 = 0\}} \{f_{i_1} - 2f_{i_0} + f_{i_2}\} \leq h|i_1 - 2i_0 + i_2|. \tag{5.34}$$

As the maximum problem is a linear program, it assumes its maximum on the set of *vertices* of $K_{1D}^{red} \cap \{f \,|\, f_1 = 0\}$. Therefore we only have to show that

$$f_{i_1} - 2f_{i_0} + f_{i_2} \leq h|i_1 - 2i_0 + i_2| \tag{5.35}$$

for all f in the finite set of vertices, i.e., satisfying $|f_{i+1} - f_i| = h$ and the inequality $f_{i+1} - 2f_i + f_{i-1} \leq 0$ (and still $f_1 = 0$). This can be argued case by case:

$i_0 < i_2 < i_1$
: As the left-hand side in (5.35) can be written as $(f_{i_1} - f_{i_0}) + (f_{i_2} - f_{i_0})$ and due the observation (5.33), the maximum is assumed on the vertex f satisfying $f_{i+1} = f_i + h$ for all i, with maximum value

$$f_{i_1} - 2f_{i_0} + f_{i_2} = h(i_1 - i_0) + h(i_2 - i_0) = h(i_1 - 2i_0 + i_2) = h|i_1 - 2i_0 + i_2|, \tag{5.36}$$

which shows that the inequality in (5.27) holds for this case.

$i_2 < i_0 < i_1$
: In this case the maximum is assumed if either $f_{i+1} = f_i + h$ or $f_{i+1} = f_i - h$ for all i, depending on which of $i_2 - i_0$ and $i_0 - i_1$ is larger. Therefore

$$f_{i_1} - 2f_{i_0} + f_{i_2} \leq \max\{\pm(h(i_0 - i_2) - h(i_1 - i_0))\} = h|i_1 - 2i_0 + i_2|. \tag{5.37}$$

$i_2 < i_1 < i_0$
: Again with the observation (5.33), we see that in this case the maximum is attained for $f_{i+1} = f_i - h$ for all i, in which case

$$f_{i_1} - 2f_{i_0} + f_{i_2} = -(f_{i_0} - f_{i_1}) - (f_{i_0} - f_{i_2}) = h(-i_2 + 2i_0 - i_1) = h|i_1 - 2i_0 + i_2|. \tag{5.38}$$

This shows that (5.35) holds for all vertices in the set $K_{1D}^{red} \cap \{f \,|\, f_1 = 0\}$, and therefore for all points, which concludes the proof of the remaining inclusion $K_{1D}^{red} \subseteq K_{1D}$. □

Interestingly, in the classical convex relaxation for the (first-order) total variation used in [19, 33], the dual constraint set is of the form

$$K_{\mathrm{TV},1D} = \bigcap_{1\le i\le \ell-1} \big\{ f \in \mathbb{R}^\ell : |f_i - f_{i+1}| \le h \big\}. \tag{5.39}$$

As the second intersection in (5.28) enforces $f_{i+1} - f_i \le f_i - f_{i-1}$, we obtain

$$K_{1D} = K_{\mathrm{TV},1D} \cap \bigcap_{2\le i\le \ell-1} \big\{ f \in \mathbb{R}^\ell : f_{i-1} - 2f_i + f_{i+1} \le 0 \big\}. \tag{5.40}$$

Thus, when moving from first- to second-order regularization in the proposed way, the only addition is an extra non-positivity constraint on the second derivative of the dual variable f.

So far we have only considered the case of a one-dimensional domain Ω. In order to generalize the construction in (5.25) to $d > 1$ dimensions, we replace the one-dimensional three-point stencil by the corresponding Laplacian stencil in higher dimensions:

$$\phi(p) = \begin{cases} |\mu| \cdot \left| \sum_{j=1}^d (i_{1,j} - 2i_0 + i_{2,j}) \right|, & \text{if } p = \mu \cdot \sum_{j=1}^d (e_{i_{1,j}} - 2e_{i_0} + e_{i_{2,j}}), \\ +\infty, & \text{otherwise,} \end{cases} \tag{5.41}$$

where $i_{1,j}$ and $i_{2,j}$ are the indices of the neighboring points of i_0 in the j-th spatial direction. The convex conjugate can be computed in a similar fashion as in the one-dimensional case:

$$\phi^*(f) = \delta_K(f) \tag{5.42}$$

with the set

$$K := \bigcap_{1\le i_0, i_{1,1}, i_{2,1}, \ldots \le \ell} \big\{ f \in \mathbb{R}^\ell : \sum_{j=1}^d (f_{i_{1,j}} - 2f_{i_0} + f_{i_{2,j}}) \le h \left| \sum_{k=1}^d (i_{1,j} - 2i_0 + i_{2,j}) \right| \big\}. \tag{5.43}$$

Taken all together, the lifted absolute Laplacian regularizer for scalar-valued images in a d-dimensional image domain becomes

$$\bar{S}_{AL,s}(\bar{u}) := \int_\Omega \sup_{f\in K} \langle \Delta \bar{u}(x),\, f \rangle \, dx. \tag{5.44}$$

In order to approximate the absolute Laplacian for lifted *vector*-valued functions $u = (u_1, \ldots, u_n)$, we apply (5.44) to the marginal distributions $\bar{u}^{(k)}(x) :=$

$\Pi_k \bar{u}(x) \in \Delta^{l_k}$ separately in each component $k \in \{1, \dots, n\}$, where

$$\Pi_k := \underbrace{(1, \dots, 1)}_{l_1 \cdot l_2 \cdot \dots \cdot l_{k-1} \text{ones}} \otimes \mathrm{Id}_{l_k} \otimes \underbrace{(1, \dots, 1)}_{l_{k+1} \cdot l_{k+2} \cdot \dots \cdot l_n \text{ones}} \in \mathbb{R}^{l_k \times \ell} \tag{5.45}$$

computes the k-th marginal distribution by summing the entries of $\bar{u}$ over all dimensions of the range with the exception of the k-th dimension. As the absolute Laplacian regularizer decouples in the components of u, it can be approximated by summing the one-dimensional regularizer of the marginalized label distribution over the label dimensions:

$$\bar{S}_{AL}(\bar{u}) := \sum_{i=1}^{n} \bar{S}_{AL,s}(\Pi_i \bar{u}) = \sum_{i=1}^{n} \int_{\Omega} \sup_{f^i \in K_{l_i}} \langle \Delta \Pi_i \bar{u}(x),\, f^i(x) \rangle \, dx. \tag{5.46}$$

Here $K_{l_i} \subseteq \mathbb{R}^{l_i}$ denotes a set of the form (5.43) in l_i-dimensional space, which accounts for the fact that there may be a different number of labels in each dimension of the range.

After discretizing the image domain $\Omega \subseteq \mathbb{R}^d$ on a d-dimensional Cartesian grid Ω', the full discretized problem can be formulated in saddle point form:

$$\inf_{\bar{u}:\Omega' \to \Delta_\ell} \sup_{f^i:\Omega' \to K_{l_i},\, i=1,\dots,n} \sum_{x \in \Omega'} \bar{\rho}(x, \bar{u}(x)) + \lambda \sum_{x \in \Omega'} \sum_{i=1}^{n} \langle \Delta \Pi_i \bar{u}(x),\, f^i(x) \rangle. \tag{5.47}$$

This problem can be readily solved using any available primal-dual method for non-smooth convex optimization.

We do not know of a result similar to Theorem 5.1 yet in order to reduce the number of constraints for the sets K_{l_i} in a similar way as for K_{1D}. Therefore, we take a pragmatic approach: we approximate each of the sets K_{l_i} by the set K_{1D} in the corresponding dimension, which amounts to an outer approximation of K_{l_i}. We can then apply Theorem 5.1 to solve the problem using the reduced number of constraints.

Experimental Results

We evaluate the proposed strategy for higher-order relaxation of non-convex problems on two applications. Firstly, we consider a non-convex denoising problem, using the MATLAB extension CVX [12, 13] to solve the primal formulation of the saddle-point problem (5.47) on an Intel Core i7-4500U CPU with 8 GB of RAM.

Secondly, we examine a real-world image registration problem, using a CUDA 7.5.17 implementation[1] of a first order primal-dual algorithm with diagonal preconditioning [5] which runs on an Nvidia GeForce GTX 680 GPU with an Intel Core i7 960 CPU and 24 Gb RAM. The implementation uses a more recent "sublabel-accurate" approach for lifting the data term in order to reduce the required resolution for the data term [18, 28].

Non-convex Denoising with Second-Order Regularity

In order to illustrate that non-convexity can be beneficial when combined with second-order regularization, we consider the simple one-dimensional denoising problem

$$\inf_{u:\Omega\to\mathbb{R}} \int_{\Omega} |u(x) - g(x)|^q \, dx + \lambda \int_{\Omega} |u''(x)| dx \tag{5.48}$$

with $\Omega \subseteq \mathbb{R}$. For $q = 1$, one obtains a simple *convex* $\mathrm{TV}^2 - L^1$ denoising model, while for $q < 1$, the energy is generally non-convex. We used the proposed method to approximate a global solution of (5.48).

The method was applied to a smooth input signal g distorted by heavy salt-and-pepper noise, with 80% of the values randomly set to 0 or 1. The locations of the outliers were unknown to the solver, and no additional preprocessing or outlier masking was performed.

As can be seen from Figs. 5.1 and 5.2, combining higher-order regularization with a non-convex data term allows to reconstruct the signal more faithfully. While both approaches prefer piecewise linear results as expected from the function-space formulation, in the convex approach with $q = 1$, input noise is carried over into the output before the structure is fully visible.

While convex methods relying on L^1 data terms are often—rightfully—referred to as "robust" methods in comparison to methods using smooth or quadratic data terms, the non-convex approach with $q = 0.1$ is even more robust against outliers and returns a decent reconstruction for a range of λ on this challenging problem. Run times were in the order of 0.3 s for a discretization of Ω using 120 grid points and $\ell = 63$ labels.

[1] See [28] and http://github.com/tum-vision/prost for the most recent version.

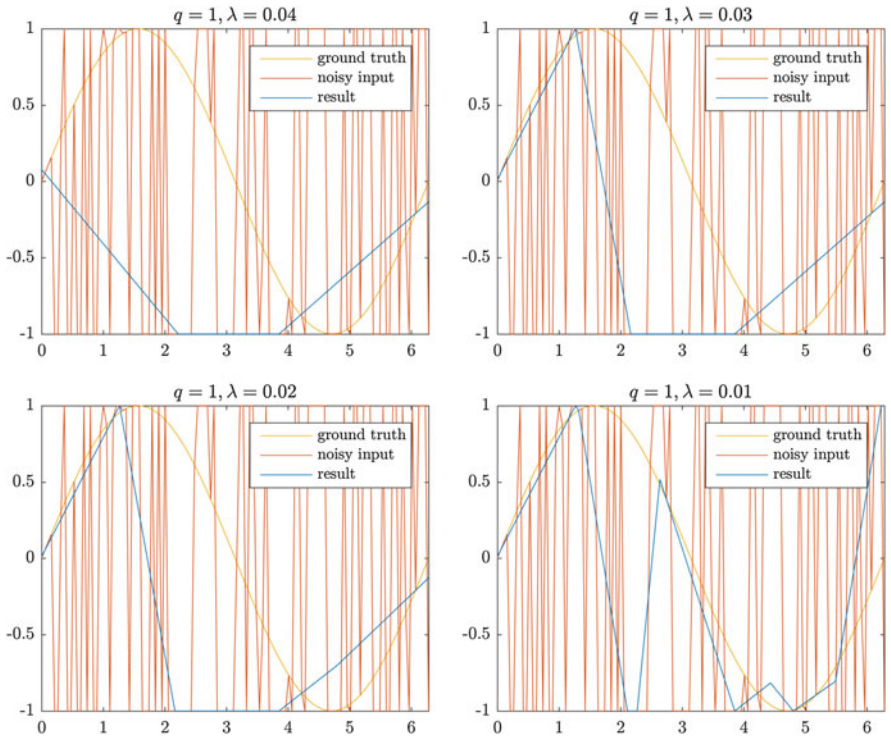


Fig. 5.1 Classical convex second-order ($TV^2 - L^1$) denoising of a smooth signal corrupted by 80% blind salt-and-pepper noise using varying regularization strength λ. The result is piecewise affine as expected from TV^2 regularization. Starting from large λ with heavy over-regularization and decreasing λ, noise is picked up early. There is no regimen where both noise is removed and the signal reconstructed faithfully

Image Registration Using the Absolute Laplacian

For a more challenging application, we apply the method to the image registration problem with SSD data term (5.6) and absolute Laplacian regularization.

Translation-Only Synthetic Image

We first apply the absolute Laplacian regularizer to a synthetic binary image registration problem. The input reference image R is a binary 64×64 image of two vertical boxes. The template image T is obtained by translating the input image by 12 pixels (Fig. 5.3, first row). Thus the ground truth is a uniform translation by 12 pixels and constitutes a global minimizer, as it has vanishing data term and the second-order regularizer does not penalize linear deformations. This configuration

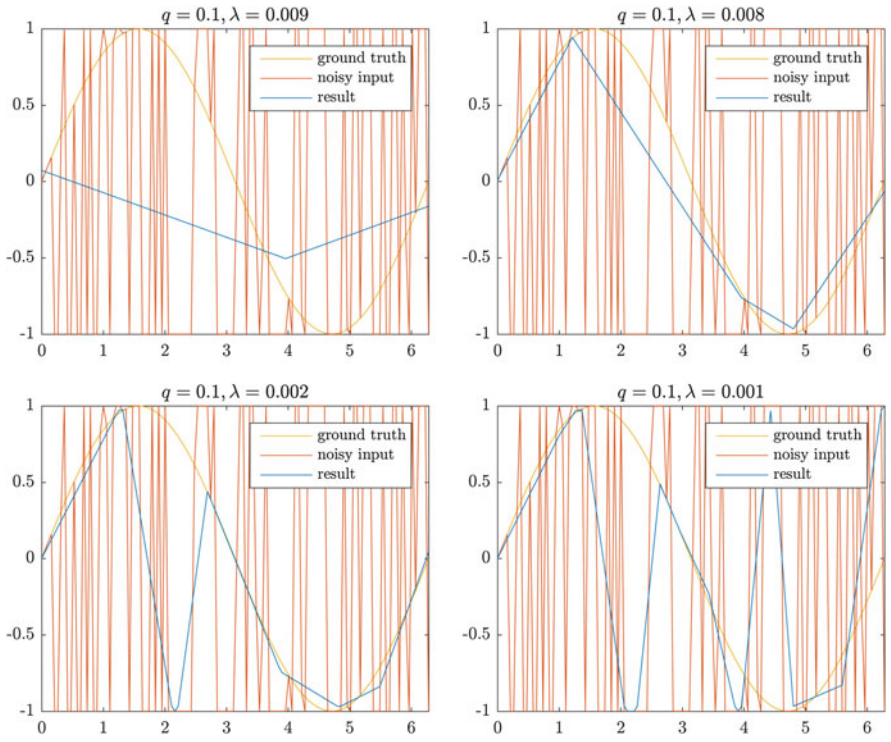


Fig. 5.2 *Non-convex* second-order ($\mathrm{TV}^2 - L^q$ with $q = 0.1$) denoising of the signal in Fig. 5.1 using the proposed convex lifting and approximation with varying regularization strength λ. The proposed approach allows to approximate *global* minimizers of such higher-order regularized non-convex models. The additional non-convexity achieves a better reconstruction of the signal (**top right**) than in the convex case (Fig. 5.1) before giving in to noise (**bottom left**)

is challenging for methods based on local optimization, as there is a strong local minimum. Furthermore, as the images contain large constant regions, the energy landscape has extensive flat regions with zero gradient.

We compare our approach to a traditional curvature-regularized model solved using a single-resolution local minimization method implemented in the MATLAB extension FAIR [24, 25]. The regularization strength was manually set to $\lambda = 10$, however a wide range of values for λ produced the same qualitative behavior. The traditional approach leads to a solution that is not globally optimal (Fig. 5.3, second row). Using our approach, we retrieve the globally optimal ground truth with $\ell = 9$ labels in the label space $\Gamma = [-12, 12]^2$ and a run time of 85 s, without having to resort to approaches such as coarse-to-fine or affine pre-registration for initialization (Fig. 5.3, bottom row).

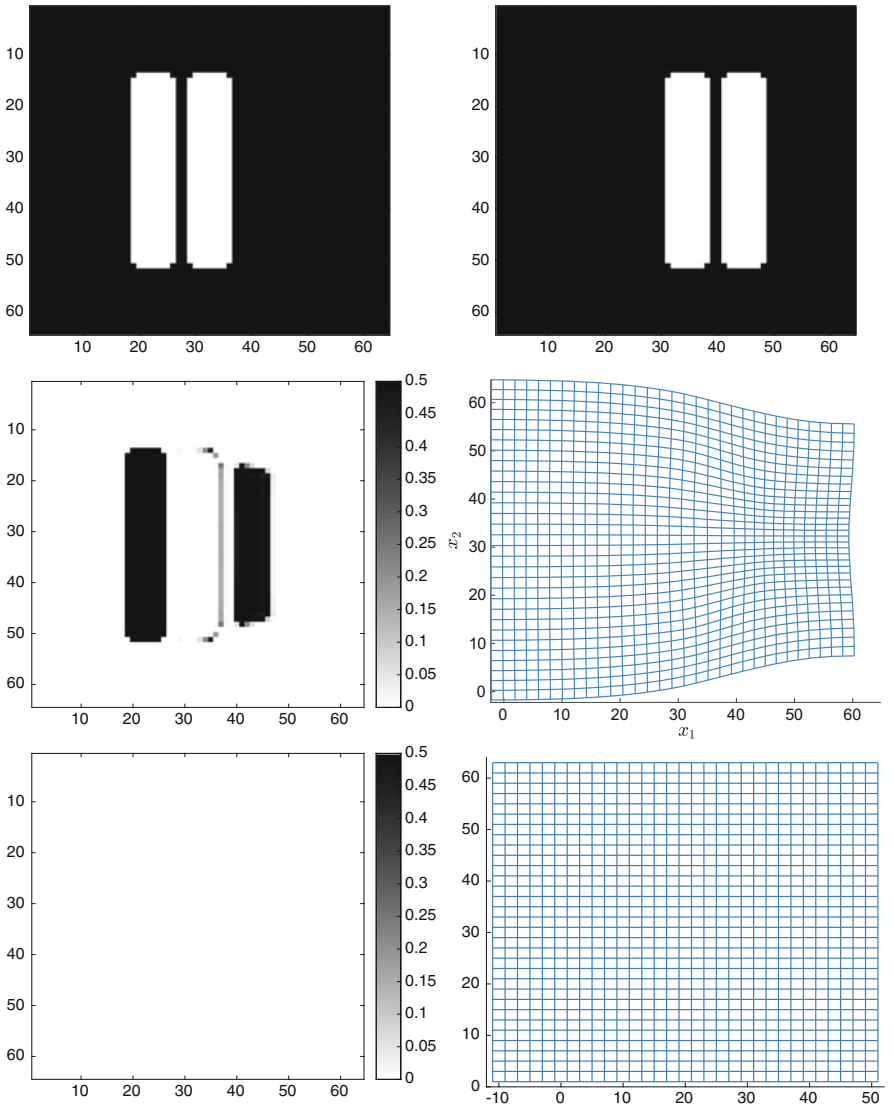


Fig. 5.3 Application of the proposed lifting for absolute Laplacian regularization to a synthetic image registration problem. A traditional curvature-regularized model solved using a local Gauss-Newton method serves as a baseline. The input reference image R **(top left)** and template image T **(top right)** differ by a ground truth translation of 12 pixels. The second and third row show the final difference images $\frac{1}{2}(R(x) - T(x + u(x)))^2$ **(left)** and obtained deformation u visualized as a deformation grid **(right)**. The classical local optimization method **(second row)** converges to a local solution which is not globally optimal and yields a non-constant deformation with a mean displacement of 2.3 pixels. Using the proposed functional lifting for absolute Laplacian regularization **(bottom row)**, the global optimum is retrieved accurately with an average displacement of 12.0002 pixels

Real-World Image Registration

As a real-world example, we employ the SSD energy with absolute Laplacian regularization to solve the image registration problem on a pair of X-ray images, and compare to the existing lifting approach [18] with total variation regularization. The regularization strength was manually set to $\lambda = 0.05$. Run times were 933 s for total variation, and 515 s for absolute Laplacian minimization.

As can be seen from the numerical results (Fig. 5.4), while the first-order total variation regularization achieves a very good data fit, it results in a physically implausible self-intersecting deformation grid (Fig. 5.4, second row). This behavior can be partly attributed to the well-known fact that total variation promotes piecewise constant solutions, also commonly referred to as stair-casing effect [8]. In the context of medical image registration, this is a highly undesired behavior, as jumps in the deformation map u correspond to infinite stretch or compression and often lead to self-intersections. In contrast, the proposed second-order regularizer (Fig. 5.4, bottom row) maintains a physically meaningful deformation, while still achieving an acceptable data fit.

Conclusion and Outlook

In this work, we have taken a first step towards extending the convex relaxation and functional lifting framework to second-order regularization. We showed how to solve the main issue of an exploding number of constraints for the absolute Laplacian regularization.

Experiments on a denoising problem showed that the combination of higher-order regularization and non-convex data terms can lead to better results than a convex model, and allows to recover highly corrupted data in a piecewise linear fashion. In the application of image registration, the absolute Laplacian faithfully retrieves simple translations and leads to a more realistic deformation grid than total variation regularization on a real-world problem.

While our relaxation allows to reduce the number of required constraints to linear complexity, it is an approximation, rather than a "tight" relaxation in the sense of an exact biconjugate, and the proof is still limited to one dimension. An open question is whether one can find a similar compact representation for the tight relaxation in more than one dimension.

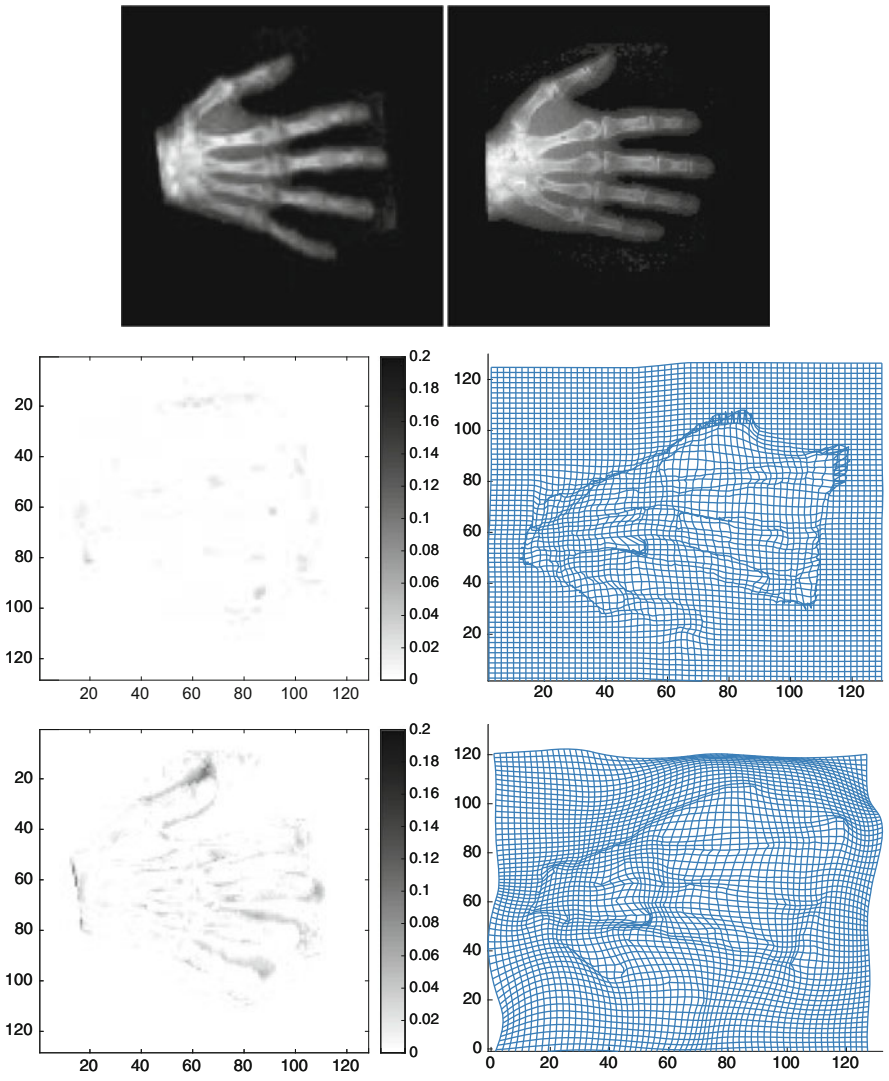


Fig. 5.4 Comparison between two global optimization methods for medical image registration: classical first-order total variation regularization [18] and the proposed second-order lifting approach. The input data consists of a pair of 128×128 grayscale X-ray images of two right hands **(top row)**. Both approaches are evaluated using $\ell = 10^2 = 100$ labels, $\Gamma = [-12, 12]^2$, $\Omega = [0, 128]^2$, and a regularization strength of $\lambda = 0.05$. The classical first-order total variation regularization generates piecewise constant deformations and a physically implausible self-intersecting deformation grid **(second row)**. The second-order regularizer avoids discontinuities and maintains a physically meaningful deformation grid **(bottom row)**

Finally, in this work we have constrained ourselves to the discretized setting. A functional-analytic discussion as well as an extension to the more recent manifold-valued and sublabel-accurate relaxations remain subject of future work.

Acknowledgements This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—394737018 "Functional Lifting 2.0: Efficient Convexifications for Imaging and Vision". We would like to thank Emanuel Laude and Thomas Möllenhoff for providing their library `prost`, which was used to solve the saddle-point formulation of the problems.

References

1. L. Ambrosio, N. Fusco, D. Pallara, *Functions of Bounded Variation and Free Discontinuity Problems* (Clarendon Press, Oxford, 2000)
2. K. Bredies, K. Kunisch, T. Pock, Total generalized variation. J. Imag. Sci. **3**(3), 294–526 (2010)
3. T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in *European Conference on Computer Vision* (Springer, Berlin, 2004), pp. 25–36
4. A. Chambolle, An algorithm for total variation minimization and applications. J. Math. Imag. Vis. **20**, 89–97 (2004)
5. A. Chambolle, T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imag. Vis. **40**(1), 120–145 (2011)
6. A. Chambolle, D. Cremers, T. Pock, A convex approach to minimal partitions. J. Imag. Sci. **5**(4), 1113–1158 (2012)
7. T. Chan, A. Marquina, P. Mulet, Higher-order total variation-based image restoration. J. Sci. Comput. **22**(2), 503–516 (2000)
8. T. Chan, S. Esedoḡlu, F. Park, A. Yip, Total variation image restoration: overview and recent developments, in *The Handbook of Mathematical Models in Computer Vision* (Springer, Berlin, 2005)
9. F. Demengel, Fonctions à Hessien borné. Ann. Inst. Fourier **34**, 155–190 (1985)
10. B. Fischer, J. Modersitzki, Curvature based image registration. J. Math. Imag. Vis. **18**(1), 81–85 (2003)
11. W.J. Fu, Penalized regressions: the bridge versus the lasso. J. Comput. Graph. Stat. **7**(3), 397–416 (1998)
12. M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in *Recent Advances in Learning and Control* (Springer, Berlin, 2008), pp. 95–110
13. M. Grant, S. Boyd, CVX: matlab software for disciplined convex programming (2014). http://cvxr.com/cvx
14. S. Henn, A full curvature based algorithm for image registration. J. Math. Imag. Vis. **24**(2), 195–208 (2006)
15. W. Hinterberger, O. Scherzer, Variational methods on the space of functions of bounded Hessian for convexification and denoising. Computing **76**(1), 109–133 (2006)
16. B.K.P. Horn, B.G. Schunck, Determining optical flow. Artif. Intell. **17**, 185–203 (1981)
17. H. Ishikawa, Exact optimization for Markov random fields with convex priors. Pattern. Anal. Mach. Intell. **25**(10), 1333–1336 (2003)
18. E. Laude, T. Möllenhoff, M. Moeller, J. Lellmann, D. Cremers, Sublabel-accurate convex relaxation of vectorial multilabel energies, in *European Conference on Computer Vision* (Springer, Berlin, 2016), pp. 614–627
19. J. Lellmann, C. Schnörr, Continuous multiclass labeling approaches and algorithms. SIAM J. Imag. Sci. **4**(4), 1049–1096 (2011)

20. J. Lellmann, J. Kappes, J. Yuan, F. Becker, C. Schnörr, Convex multi-class image labeling by simplex-constrained total variation, in *Scale Space and Variational Methods in Computer Vision*. Lecture Notes in Computer Science, vol. 5567 (2009), pp. 150–162
21. J. Lellmann, E. Strekalovskiy, S. Koetter, D. Cremers, Total variation regularization for functions with values in a manifold, in *International Conference on Computer Vision* (2013), pp. 2944–2951
22. M. Lysaker, X.C. Tai, Iterative image restoration combining total variation minimization and a second-order functional. Int. J. Comput. Vis. **66**(1), 5–18 (2006)
23. M. Lysaker, A. Lundervold, X.C. Tai, Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. IEEE Trans. Image Process. **12**(12), 1579–1590 (2003)
24. J. Modersitzki, *Numerical Methods for Image Registration* (Oxford University Press on Demand, Oxford, 2004)
25. J. Modersitzki, *FAIR: Flexible Algorithms for Image Registration* (SIAM, Philadelphia, 2009)
26. T. Möllenhoff, E. Strekalovskiy, M. Moeller, D. Cremers, The primal-dual hybrid gradient method for semiconvex splittings. SIAM J. Imag. Sci. **8**(2), 827–857 (2015)
27. T. Möllenhoff, E. Strekalovskiy, M. Möller, D. Cremers, Low rank priors for color image regularization, in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* (Springer, Berlin, 2015), pp. 126–140
28. T. Möllenhoff, E. Laude, M. Moeller, J. Lellmann, D. Cremers, Sublabel-accurate relaxation of nonconvex energies, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 3948–3956
29. M. Nikolova, Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares. Multiscale Model. Simul. **4**(3), 960–991 (2005)
30. J. Nocedal, S.J. Wright, *Numerical Optimization* (Springer, Berlin, 2006)
31. K. Papafitsoros, C.B. Schönlieb, A combined first and second order variational approach for image reconstruction. J. Math. Imag. Vision **48**(2), 308–338 (2014)
32. T. Pock, A. Chambolle, D. Cremers, H. Bischof, A convex relaxation approach for computing minimal partitions, in *Computer Vision and Pattern Recognition* (2009)
33. T. Pock, D. Cremers, H. Bischof, A. Chambolle, Global solutions of variational models with convex regularization. J. Imag. Sci. **3**(4), 1122–1145 (2010)
34. R.T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, 1970)
35. L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
36. S. Setzer, G. Steidl, T. Teuber, Infimal convolution regularizations with discrete l1-type functionals. Commun. Math. Sci **9**(3), 797–872 (2011)
37. P. Suquet, Existence et régularité des solutions des équations de la plasticité parfaite. C. R. Acad. Sci. Paris, Ser. A **286**, 1201–1204 (1978)
38. J. Yuan, E. Bae, X.C. Tai, Y. Boykov, A continuous max-flow approach to Potts model, in *European Conference on Computer Vision* (2010), pp. 379–392
39. C. Zach, D. Gallup, J.M. Frahm, M. Niethammer, Fast global labeling for real-time stereo using multiple plane sweeps, in *Vision, Modeling, and Visualization* (2008)
40. C. Zach, C. Häne, M. Pollefeys, What is optimized in convex relaxations for multilabel problems: connecting discrete and continuously inspired map inference. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 157–170 (2014)
41. W.P. Ziemer, *Weakly Differentiable Functions* (Springer, Berlin, 1989)

Chapter 6
On the Convex Model of Speckle Reduction

Faming Fang, Yingying Fang, and Tieyong Zeng

Abstract Speckle reduction is an important issue in image processing realm. In this paper, we propose a novel model for restoring degraded images with multiplicative noise which follows a Nakagami distribution. A general penalty term based on the statistical property of the speckle noise is used to guarantee the convexity of the denoising model. Moreover, to deal with the minimizing problem, a generalized Bermudez-Moreno algorithm is adopted and its convergence is analysed. The experimental results on some images subject to multiplicative noise as well as comparisons to other state-of-the-art methods are also presented. The results can verify that the new model is reasonable.

Introduction

Speckle has been widely known as one of the main drawbacks in synthetic aperture radar (SAR) images. It significantly degrades the visual appearance of images as a signal dependent noise and eventually leads to diminishing performance of other vision tasks based on the SAR images such as image interpretation and information extraction [24]. In fact, the speckle is a comprehensive result of attenuation and scattering, which means it is hard to control speckle from origin [5]. Due to

F. Fang
Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai, China

Department of Computer Science & Technology, East China Normal University, Shanghai, China
e-mail: fmfang@cs.ecnu.edu.cn

Y. Fang
Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong
e-mail: fangyingying@life.hkbu.edu.hk

T. Zeng (✉)
Department of Mathematics, The Chinese University of Hong Kong, Ma Liu Shui, Hong Kong
e-mail: zeng@math.cuhk.edu.hk

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_6

the inevitability of the speckle and the important applications of SAR images, despeckling therefore is an essential preliminary process of using SAR images and a number of despeckling methods have been proposed over the past decade to provide better performance of images.

The multi-look technique is a typical spatial method. It reduces noise by means of averaging the values from several independent observations. However, this simple process brings about a significant loss of resolution due to the overlook of the characteristics of SAR images. Therefore, filtering techniques have been employed in a great deal of research to overcome this deficiency, which perform better in preserving the details and edges of images while removing the noise meanwhile. Such filters including Lee filter in [17–19] and sigma filter in [19–21] were based on a minimum mean-square error (MMSE) approach, and were later refined to a more sophisticated maximum a posteriori (MAP) approach such as T-MAP filter in [23]. Anisotropic diffusion, another popular technique in the image processing community, has also been used to act as spatial filters, e.g., speckle reducing anisotropic diffusion (SRAD) [33] and detail preserving anisotropic diffusion (DPAD)[1]. However, these spatial filters have fallen out of favor in recent years mainly because of the limited performance in either preserving image details or moving off speckles.

Wavelet-based despeckling, as a new filter-based methodology, is a representative technique in transformed domain [4, 28]. Despeckling with the transform method requires to remove the speckle from the new formulation of the signal in the transformed domain followed by an inverse transformation applied on the image [4]. Image filtering in the domain of wavelet features a combination of a low-pass filter which guarantees a high-speed identification and a high-pass filter aimed to extract details. Combined with a MAP Bayesian estimation model [4, 12, 22], despeckling in the wavelet domain realizes superior performance in both noise reduction and detail preservation.

Apart from the techniques performed in the transformed domain, the non-local(NL) algorithm-based method is one of the most promising modern solutions in the field of image restoration. Several NL methods for despeckling [13, 25, 32] have also been proposed in recent years. The characteristic NL filters, e.g., the probabilistic patch-based (PPB) [13], provide favorable results from weighted averages of selected pixels according to their similarities with the fixed pixel. The NL principle has also been successfully integrated with the wavelet representation, e.g., the block-matching 3-D filter (BM3D) [11]. The NL means in BM3D is utilized to form the final 3-D groups of pixels by looking for suitable blocks of pixels throughout the image with an appointed one, after which Wiener filtering is applied to the wavelet coefficients of the obtained groups. An improved version of BM3D [25] takes the specific traits of SAR images into consideration and displays a more favorable performance than other techniques on SAR images.

Another important branch of modern denoising approaches is the variational method which transforms the despeckling image problem into a variational problem. In the variational method, a recovered image is obtained by minimizing the suitable energy functional which consists of a data fidelity term and a regularization term

inspired by an image prior to guarantee smoothness as well as edge preservation of recovered images. The variational regularization was firstly introduced for the Additive White Gaussian Noise (AWGN) denoising by Rudin et al. [29] followed by a number of variational models developed for removing multiplicative noise. Total variation (TV) regularization is one of the best-known regularizers and thus TV-based methods have been widely used for the despeckling tasks in different domains. In 2008, Aubert and Aujol [7] initially proposed the optimization model (termed as AA model) in the original intensity field and adopted a data fidelity term using a MAP method.

However, in term of non-uniqueness problem caused by the proposed non-convexity model defined in the original field, the logarithmic domain has been particularly considered later. In [30], the data term of the nonconvex model(AA model introduced in [7]) was subsequently converted to a convex model by Shi and Osher. And in [16], this model was modified to a simpler alternating minimization model by adding a quadratic term.

In this paper, we focus on the despeckling of SAR images by variational methods. Due to the non-convexity of the AA model, which causes the convergence and uniqueness problems, we propose a new convex model based on the statistical properties of the multiplicative Nakagami distribution. Moreover, we study the existence and uniqueness of the solution to the new model and use the generalized Bermudez-Moreno algorithm to solve the minimization problem. The numerical results in this paper show that our model has potential advantages over existing methods in terms of less staircasing artifacts widely exist in the variational methods.

The reminder of this paper is organized as follows. Section "A Convex Model for Despeckling" elaborates our new convex model of speckle reduction. Subsequently, section "Numerical Scheme Using Bermudez-Moreno Algorithm" shows the numerical scheme based on the Bermudez-Moreno algorithm to minimize the proposed model. Finally, section "Experimental Results and Analysis" demonstrates the experimental results and section "Conclusions" gives our conclusion.

A Convex Model for Despeckling

In this section, we first briefly introduce the statistical properties of speckle noise and then show our proposed despeckling model.

Speckle Noise

Given a connected bounded open subset $\Omega \subseteq R^2$ with compact Lipschitz boundary, assume that an image $u : \Omega \to R$ is a real function, the degraded image f can be modeled as:

$$f = un, \tag{6.1}$$

where $n \in L^2(\Omega)$ represents speckle noise with mean 1. Generally, f is assumed to be larger than 0. In the multiplicative speckle model, we focus on the assumption that n follows a Nakagami distribution, i.e., the conditional PDF of f given u is,

$$p(f|u, L) = \frac{2L^L}{\Gamma(L)u^{2L}} f^{2L-1} e^{-\frac{Lf^2}{u^2}}, \tag{6.2}$$

where L is the number of looks, and $\Gamma(\cdot)$ is the well-known Gamma function.

According to the Bayes rule, by using a maximum a posteriori (MAP) estimator, the following energy term via $E = -\log p(f|u, L)$ can be obtained,

$$E_1(u, f) = \int_\Omega \log u^{2L} + \frac{Lf^2}{u^2} dx \propto \int_\Omega \left(2\log u + \frac{f^2}{u^2}\right) dx. \tag{6.3}$$

Convex Variational Model for Despeckling

The classical TV-based despeckling model is as follows:

$$E(u) = \int_\Omega \left(2\log u + \frac{f^2}{u^2}\right) dx + \alpha \int_\Omega |Du|, \tag{6.4}$$

where $\int_\Omega |Du|$ is the seminorm in the BV(Ω), and denotes the space of functions of bounded variation. It is defined as [2, 7]:

$$\int_\Omega |Du| = \sup \left\{ \int_\Omega u \mathrm{div}\varphi dx;\ \varphi \in C_0^1(\Omega)^N,\ |\varphi|_{L^\infty(\Omega)} \leq 1 \right\}.$$

Unfortunately, Eq. (6.4) is nonconvex with respect to u, which will lead to the uncertainty of the denoising result. To avoid this drawback, we propose to use the following energy model,

$$E(u) = \alpha \int_\Omega |Du| + \int_\Omega \left(2\log u + \frac{f^2}{u^2}\right) dx + \mu G(\frac{u}{f}). \tag{6.5}$$

Here $G(\frac{u}{f}) = \int_\Omega g(\frac{u}{f})dx$, $g(\cdot) \in \mathbf{C}^2$ is a convex function, and $\mu > 0$ is a parameter which makes the energy (6.5) convex.

Additionally, we need to define a functional space in which we search for a minimizer for u. Here we set

$$S(\Omega) := \{u \in BV(\Omega) : u \geq 0\}.$$

Readily, $S(\Omega)$ is a convex and closed space. To ensure the completeness of the space, we set $\log 0 = -\infty$, $\frac{1}{0} = +\infty$ and $2\log 0 + \frac{f^2}{0^2} = \infty$.

Proposition 6.1 *If* $\mu \geq \sup_{y>0}\left\{\frac{2y^2-6}{y^4 g''(y)}\right\}$, *then the energy* (6.5) *is convex.*

Proof Given $y > 0$, let h be a function defined as:

$$h(y) = 2\log y + \frac{1}{y^2} + \mu g(y). \tag{6.6}$$

The second order derivative of h can be readily calculated by

$$h''(y) = -\frac{2}{y^2} + \frac{6}{y^4} + \mu g''(y). \tag{6.7}$$

The function (6.6) is convex if and only if $h''(y) \geq 0$ for all positive y. Thus, taking the convexity of $g(y)$ into account, we have

$$\mu \geq \sup_{y>0}\left\{\frac{2y^2-6}{y^4 g''(y)}\right\}. \tag{6.8}$$

Therefore, if μ satisfies (6.8), $h(y)$ is convex. For each $x \in \Omega$, setting $y = \frac{u(x)}{f(x)}$, we can obtain the convexity of the last two terms of (6.5). Since the term $\int_\Omega |Du|$ is also convex, the convexity of the energy (6.5) is reached.

The next issue is that we should determine the formulation of $G(\frac{u}{f})$ based on some certain properties of the speckle noise.

Actually, one simple nature is that the distribution of most data set can be modeled as Gaussian distribution. On the other hand, we randomly selected 500 clear images as f, and obtained corresponding u by adding speckle noise which follows Nakagami distribution to f. The pdfs of $\frac{u}{f}$ with different L are shown in Fig. 6.1. As we can see, their pdfs are very close to Gaussian distribution (with mean one). Moreover, as L increases, the pdf decreases more rapidly and approximates to symmetry, which actually are two essential properties of Gaussian distribution. Therefore, the functional G can be approximately defined as:

$$G(u) = \int_\Omega |\frac{u}{f} - 1|^2 dx.$$

Accordingly, (6.5) can be written as the following special case:

$$E(u) = \alpha \int_\Omega |Du| + \int_\Omega \left(2\log u + \frac{f^2}{u^2}\right) dx + \mu \int_\Omega |\frac{u}{f} - 1|^2 dx. \tag{6.9}$$

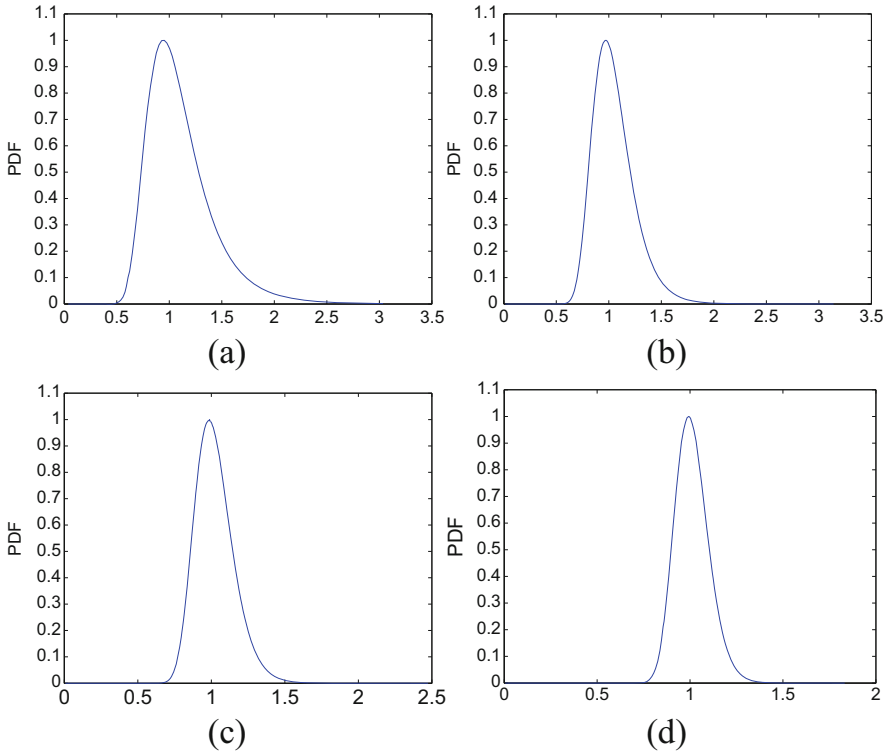


Fig. 6.1 The pdfs of $\frac{u}{f}$ with different L. (**a**) L=4. (**b**) L=8. (**c**) L=16. (**d**) L=32

In this case, the convex property of (6.9) can be directly given by:

Proposition 6.2 *If $\mu > \frac{1}{12}$, then the energy (6.9) is strictly convex.*

Proof Since $g(y) = |y-1|^2$, we have $g''(y) = 2$. According to Proposition 6.1, μ satisfies:

$$\begin{aligned} \mu &\geq \sup_{y>0}\left\{\frac{2y^2-6}{y^4 g''(y)}\right\} = \sup_{y>0}\left\{\frac{2y^2-6}{2y^4}\right\} \\ &= \sup_{y>0}\left\{-3(\tfrac{1}{y^2}-\tfrac{1}{6})^2+\tfrac{1}{12}\right\} = \tfrac{1}{12}. \end{aligned} \tag{6.10}$$

From above analysis, the h'' in (6.7) reaches its unique minimum $\frac{12\mu-1}{6}$ at $y = \sqrt{6}$. Since $\mu > \frac{1}{12}$, we have $h'' > 0$. Hence we can proof the strict convexity of energy (6.9) as h is strictly convex.

According to Proposition 6.2, we know that given a suitable μ, the energy (6.9) is a convex approximation of the non-convex energy (6.4). Next, we discuss the existence and uniqueness of the solution of the energy (6.9). Indeed, we have the following conclusion:

Theorem 6.1 *Let $f > 0$ be in $L^{\infty}(\Omega)$, then minimization problem (6.9) admits a solution $u^* \in S(\Omega)$ with*

$$0 < \inf_{\Omega} f \leq u^* \leq \sup_{\Omega} f.$$

Furthermore, u^* is unique if $\mu > \frac{1}{12}$.

Proof Let

$$F(u) = \frac{1}{\alpha}\int_{\Omega}\left(2\log u + \frac{f^2}{u^2}\right)dx + \frac{\mu}{\alpha}\int_{\Omega}|\frac{u}{f} - 1|^2 dx.$$

With $f > 0$, for all $x \in \Omega$, using (6.9), we have

$$E(u) \geq \int_{\Omega}\left(2\log u + \frac{f^2}{u^2}\right)dx \geq \int_{\Omega}(1 + 2\log f)\,dx.$$

That is, $E(u)$ in (6.9) is bounded from below. Thus, we can extract a minimizing sequence $\{u_i \in S(\Omega) : i = 1, 2, \ldots\}$. Since for each $x \in \Omega$, the real function

$$h(y) := 2\log y + \frac{f^2(x)}{y^2} + \mu|\frac{y}{f(x)} - 1|^2,$$

is decreasing if $y \in [0, f)$ and increasing if $y \in [f, +\infty)$. Therefore, $h(\min(y, M)) \leq h(y)$ is derived with $M > f(x)$. Thus, we can deduce that

$$F(\inf(u, \sup_{\Omega} f)) \leq F(u).$$

Furthermore, according to [14], with $\int_{\Omega}|D\inf(u, \sup_{\Omega} f)| \leq \int_{\Omega}|Du|$, we can deduce

$$E(\inf(u, \sup_{\Omega} f)) \leq E(u).$$

Similarly, we can get $E(\sup(u, \inf_{\Omega} f)) \leq E(u)$. Therefore, the assumption $0 < \inf_{\Omega} f \leq u^* \leq \sup_{\Omega} f$ is reasonable. That is, u_i is bounded in $L^1(\Omega)$.

Since $\{u_i\}$ is a minimizing sequence, we thus have $E(u_i)$ is bounded. Moreover, $\int_{\omega}|Du_i|$ is bounded, and then $\{u_i\}$ is bounded in the $BV(\Omega)$. Hence, There exists a subsequence $\{u_{i_k}\}$ and $u^* \in BV(\Omega)$ such that,

$$u_{i_k} \underset{L^1(\Omega)}{\longrightarrow} u^* \text{ and } u_{i_k} \underset{BV-\mathrm{w}^*}{\rightharpoonup} u^*.$$

With the space $S(\Omega)$ convex and closed, combining the Fatou's lemma and the w.l.s.c of the BV space, we can conclude that

$$\min_{u\in S(\Omega)} E(u) = \liminf_{i_k\to+\infty} E(u_{i_k}) \geq E(u^*),$$

i.e., u^* is a minimum point of $E(u)$, and $0 < \inf_\Omega f \leq u^* \leq \sup_\Omega f$.

Besides, $E(u)$ is strictly convex while $\mu > \frac{1}{12}$, therefore the uniqueness of u^* is guaranteed directly.

Numerical Scheme Using Bermudez-Moreno Algorithm

In this section, we propose to use Bermudez-Moreno (BM) algorithm [9] to solve the model (6.9). In what follows, we first present a generalized form of BM algorithm, then use this algorithm to solve our minimization problem.

Generalized Form

BM algorithm is a general minimization algorithm. It can provide efficient procedures to deal with the image processing problem [8]. The general minimization problem it can solve is as follows:

$$\min_{y\in V} \{F(y) + \varphi(y)\}, \tag{6.11}$$

where V is a Hilbert space and $\varphi(\cdot)$ is a proper lower semi-continuous (l.s.c.) convex function in V:

$$\varphi = \phi \circ B^*.$$

Here $\circ$ is the compound operator, $\phi : E \to R$ is a l.s.c. convex function, $B : E \to V$ is a bounded linear operator, B^* is the adjoint of B and E is a Hilbert space.

It has been proved that the subdifferential of a l.s.c. convex function ϕ (denoted by $H = \partial\phi$) is a maximal monotone operator [3, 8, 27], and we have the following remark:

Remark 6.1 If H is a maximal monotone operator, we denote by H_λ its Yosida approximation (L_λ is the resolvent of λH)[27]:

$$H_\lambda = \frac{I - L_\lambda}{\lambda}, \text{ where } L_\lambda = (I + \lambda H)^{-1}. \tag{6.12}$$

Given an arbitrary initial value y_0, we generalize Bermudez and Moreno algorithm to minimize (6.11):

$$\begin{cases} -F'(z_i) = By_i \\ y_{i+1} = H_\lambda(B^* z_i + \lambda y_i) \end{cases}, \tag{6.13}$$

where we suppose the F' invertible.

Application to the Proposed Method

We first rewrite our energy (6.9) as:

$$\min_{u \in S'(\Omega)} E(u) = J(u) + F(u), \tag{6.14}$$

where

$$S'(\Omega) := \{u \in L^2(\Omega) : u \geq 0\},$$

and $J(u)$ is the total variation of u extended to $L^2(\Omega)$ (we know that $BV(\Omega) \subset L^2(\Omega)$):

$$J(u) = \begin{cases} \int_\Omega |Du|, & u \in BV(\Omega) \\ +\infty, & \text{otherwise} \end{cases}.$$

From the definition, the formula (6.14) has the same minimizer to (6.9) clearly.

In fact, the formula (6.14) is an example of the problem (6.11). Let $V = L^2(\Omega)$ and $E = (L^2(\Omega))^2$, then $J(u) = \varphi(u) = \phi(B^*(u))$ with $B = -\text{div} = \nabla^*$ and $B^* = \nabla$. The ϕ is the support function of K:

$$\phi(w) = \sup_{v \in K} \langle v, w \rangle_E,$$

and

$$J(u) = \sup_{w \in K} \langle u, \text{div} w \rangle,$$

where the K, a closed convex set in E, is defined as:

$$K = \left\{ w \in E / \text{div} w \in V,\ \|w\|_\infty \leq 1,\ |w| = \sqrt{w_1^2 + w_2^2} \right\}.$$

Accordingly, $H_\lambda(w)$ is the orthogonal projection of $\frac{w}{\lambda}$ onto K [8, 27]. That is, $H_\lambda(w) = P_K(\frac{w}{\lambda})$, and

$$P_K(w) = \left(\frac{w_1}{\max\{1, |w|\}}, \frac{w_2}{\max\{1, |w|\}}\right).$$

In this case, the BM algorithm of the proposed model is:

$$\begin{cases} -F'(u_i) = -\text{div}y_i \\ y_{i+1} = P_K(\frac{1}{\lambda}\nabla u_i + y_i) \end{cases}. \tag{6.15}$$

The second line of equation (6.15) is a direct formula. As for the first line, it can be rewritten as,

$$\frac{2}{\alpha}\left(\frac{f^2u_i^2 - f^4 + \mu u_i^4 - \mu f u_i^3}{u_i^3 f^2}\right) = -\text{div}y_i, \tag{6.16}$$

which is equivalent to the following quartic equation,

$$\mu u_i^4 + eu_i^3 + f^2u_i^2 - f^4 = 0, \tag{6.17}$$

where $e = (\frac{\alpha}{2}\text{div}y_i f^2 - \mu f)$. Assume that $v_i = u_i + \frac{e}{4\mu}$, then (6.17) can be turned into

$$v_i^4 + av_i^2 + bv_i + c = 0. \tag{6.18}$$

Here

$$\begin{cases} a = \frac{f^2}{\mu} - \frac{3e^2}{8\mu^2} \\ b = \frac{e^3}{8\mu^3} - \frac{ef^2}{2\mu^2} \\ c = \frac{e^2f^2}{16\mu^3} - \frac{3e^4}{256\mu^4} - \frac{f^4}{\mu} \end{cases}.$$

It is easy to verify that the following equation is equivalent to Eq. (6.18) for any variable r,

$$(v_i^2 + a + r)^2 = (a + 2r)v_i^2 - bv_i + (a^2 - c + 2ar + r^2). \tag{6.19}$$

We can select a suitable r to ensure that the right hand side of the above equation is a perfect square. In this case, r should satisfy:

$$b^2 - 4(a + 2r)(a^2 - c + 2ar + r^2) = 0. \tag{6.20}$$

Equation (6.20) is a cubic equation, and the closed form solution of r can be obtained directly. Then Eq. (6.19) is simplified into two quadric equations:

$$v_i^2 + a + r = \pm\sqrt{a+2r}(v_i - \frac{b}{2(a+2r)}).$$

Hence,

$$v_i = \frac{1}{2}\left(-\sqrt{a+2r} \pm \sqrt{a+2r-4(a+r-\frac{b}{2\sqrt{a+2r}})}\right),$$

or

$$v_i = \frac{1}{2}\left(+\sqrt{a+2r} \pm \sqrt{a+2r-4(a+r+\frac{b}{2\sqrt{a+2r}})}\right).$$

Besides, we have the following proposition for the solution of u_i:

Proposition 6.3 *Let $\mu > \frac{1}{12}$ and $\alpha > 0$. Given the image $f > 0$, there only exists one positive solution with respect to u for the following equation:*

$$-F'(u_i) = -\mathrm{div}y_i \text{ (see (6.15))}.$$

Proof Readily, we have

$$-F'(u_i) = \frac{2}{\alpha}\left(\frac{1}{u_i} - \frac{f^2}{u_i^3} + \frac{\mu u_i}{f^2} - \frac{\mu}{f}\right).$$

Then, combining $f > 0$ and $\mu > \frac{1}{12}$, we can deduce that

$$\begin{aligned} F''(u_i) &= \frac{2}{\alpha}\left(-\frac{1}{u_i^2} + \frac{3f^2}{u_i^4} + \frac{\mu}{f^2}\right) \\ &= \frac{2}{\alpha f^2 u_i^4}\left(\mu u_i^4 - f^2 u_i^2 + 3f^4\right) \\ &= \frac{2}{\alpha f^2 u_i^4}\left(\mu\left(u_i^2 - \frac{f^2}{2\mu}\right)^2 + (3 - \frac{1}{4\mu})f^4\right) \\ &> 0. \end{aligned} \tag{6.21}$$

That is, $F'(u_i)$ is strictly monotonically increasing. Then, the equation $-F'(u_i) = -\mathrm{div}y_i$ has only one solution.

Besides, we can deduce that:

Proposition 6.4 *U is a solution of*

$$-F'(u) \in -\mathrm{div}\partial\phi(\nabla u), \tag{6.22}$$

if and only if (u, y) *is a solution of*

$$\begin{cases} -F'(u) = -\mathrm{div}y \\ y = P_K(\frac{1}{\lambda}\nabla u + y) \end{cases}. \tag{6.23}$$

Proof As aforementioned, $H = \partial\phi$ and $H_\lambda(w) = P_K(\frac{w}{\lambda})$. Since H is a maximal monotone operator, the following two conditions are equivalent (see Lemma 2.1 in [9]):

- $y \in H(z)$,
- $y = H_\lambda(z + \lambda y)$.

Then Proposition 6.4 is a direct consequence.

We also present a Lemma:

Lemma 6.1 *We have*

$$\frac{1}{\lambda^2}\|L_\lambda(z_1) - L_\lambda(z_2)\|^2 + \|H_\lambda(z_1) - H_\lambda(z_2)\|^2 \leq \frac{1}{\lambda^2}\|z_1 - z_2\|^2.$$

Proof This inequality can be immediately derivated by the definitions in (6.12).

Based on the analyses above, we have the following convergence result.

Theorem 6.2 *Assume that* $\mu > \frac{1}{12}$ *and* $\lambda > \alpha\|\nabla\|^2$, *then the sequence* (u^i, y^i) *defined in* (6.15) *is such that* $u^i \to u$ *(for the strong topology of* $L^2(\Omega)$*) and* $y^i \rightharpoonup y$ *(in* $L^2(\Omega) \times L^2(\Omega)$ *weak), with u the solution of* (6.15).

Proof First, since $\mu > \frac{1}{12}$, it is obvious that F is convex and strong Lipschitz differentiable. Besides, ϕ is a proper l.s.c. convex function.

From Lemma 6.1, we have:

$$\begin{aligned} &\frac{1}{\lambda^2}\|L_\lambda(\nabla u + \lambda y) - L_\lambda(\nabla u^i + \lambda y^i)\|^2 + \|y - y^{i+1}\|^2 \\ &\leq \frac{1}{\lambda^2}\|\nabla(u - u^i) + \lambda(y - y^i)\|^2 \\ &= \frac{1}{\lambda^2}\|\nabla(u - u^i)\|^2 + \frac{2}{\lambda}\langle\nabla(u - u^i), y - y^i\rangle + \|y - y^i\|^2. \end{aligned}$$

Using the Bermudez-Moreno algorithm, we know that:

$$F'(u) - F'(u^i) = -\text{div}(y^i - y).$$

Taking the inner product with $(u - u^i)$, we obtain:

$$\langle F'(u) - F'(u^i), u - u^i \rangle = \langle -\text{div}(y^i - y), u - u^i \rangle = \langle y^i - y, \nabla(u - u^i) \rangle.$$

Therefore,

$$\begin{aligned} \langle y^i - y, \nabla(u - u^i) \rangle &= \langle F'(u) - F'(u^i), u - u^i \rangle \\ &\leq -\frac{\alpha}{2} \|u - u^i\|^2 \\ &\leq -\frac{\alpha}{2\|\nabla\|^2} \|\nabla(u - u^i)\|^2. \end{aligned} \tag{6.24}$$

We now deduce that:

$$\begin{aligned} &\frac{1}{\lambda^2} \|L_\lambda(\nabla u + \lambda y) - L_\lambda(\nabla u^i + \lambda y^i)\|^2 + \|y - y^{i+1}\|^2 \\ &\leq \frac{1}{\lambda}\left(\frac{1}{\lambda} - \frac{\alpha}{\|\nabla\|^2}\right) \|\nabla(u - u^i)\|^2 + \|y - y^{i+1}\|^2. \end{aligned} \tag{6.25}$$

Since $\lambda > \alpha\|\nabla\|^2$, with $u^i \neq u$, we obtain:

$$\|y - y^{i+1}\| \leq \|y - y^i\|.$$

Thus we deduce that $\|y - y^i\|$ is a convergent sequence in R. Taking limit to the above inequality, we get:

$$\lim_{m \to \infty} \|\nabla(u - u^i)\| = 0.$$

Then, using (6.24), we have $u \to u^i$.

For the proof of the convergence of y^i, we first pass to the limit in (6.25) to obtain:

$$L_\lambda(\nabla u^i + \lambda y^i) \to L_\lambda(\nabla u + \lambda y).$$

Taking $L_\lambda = I - \lambda H_\lambda$ and the second line of (6.23), we get that:

$$L_\lambda(\nabla u + \lambda y) = (\nabla u + \lambda y) - \lambda H_\lambda(\nabla u + \lambda y) = \nabla u.$$

From the second line of (6.15), we obtain:

$$y^{i+1} = H_\lambda(\nabla u^i + \lambda y^i) = y^i + \frac{1}{\lambda}(\nabla u^i - L_\lambda(\nabla u^i + \lambda y^i).$$

Taking limit to the equation, we then have:

$$\lim_{m\to\infty}\{y^{i+1} - y^i\} = 0.$$

Moreover, since

$$z \in L^2(\Omega) \times L^2(\Omega) \to H_\lambda(\nabla u(z) + \lambda z),$$

with $u(z)$ the solution of $-F'(u) = -\text{div}z$ non-expansive [26], we can conclude that $y^i \rightharpoonup y$ in $L^2(\Omega) \times L^2(\Omega)$ is weak.

Remark 6.2 In the discrete scheme, $\|\nabla\|^2 \leq 8$ (see [10]). Thus the priori condition $\lambda > \alpha\|\nabla\|^2$ in Theorem 6.2 can be transformed into $\lambda > 8\alpha$.

Finally, it should be noted that our algorithm stops when the iteration number i reaches t_{max} or the relative error $\frac{\|u^i - u^{i-1}\|_2}{\|u^{i-1}\|_2} < \xi$. In addition, we set $t_{max} = 200$, $\xi = 10^{-4}$ in the following experiments.

Experimental Results and Analysis

In this section, in order to examine the effectiveness of the proposed method, we present and analyse the experimental results on some images. We also compare our model with three state-of-the-art methods, i.e., AA (Aubert and Aujol) model [6], SO (Shi and Osher) model [30] and I-divergence model [31]. Note that all the following experiments are implemented in Matlab R2013a on an Intel(R) 3.33 GHz PC with 12 GB RAM.

For illustrations, the results of three images "Water castle", "Cameraman" and "Boat" are presented. The original images are shown in Fig. 6.2. Besides, we use the mean of the peak signal to noise ratio (PSNR) [15], which is very famous and widely used in the image processing realm, to quantitatively measure the denoising results.

In each of the figures shown in Figs. 6.3, 6.4 and 6.5, test images are corrupted by multiplicative noise with the number of looks L=4, 8 and 16, respectively, and the denoising results of multiplicative noise with L= 4, 8 and 16, are shown in the 1st-3rd columns. Besides, the first row is the noisy image, and from top to bottom rows

Fig. 6.2 Original images. (**a**) "Castle" (size: 481 × 321), (**b**) "Cameraman" (size: 256 × 256), (**c**) "Boat" (size: 512 × 512)

are the denoising results produced by AA (2nd row), SO (3rd row), I-divergence (4th row) and proposed methods (5th row), respectively.

From these figures, we can observe that the proposed method empirically approach the ground truth of the original images, and the denoised results can restore some fine texture details. By contrast, the results of other three methods are either smoothing the details on the edge or remaining some noise in the smooth regions. As a classical denoising method, AA deals with the multiplicative noise using a minimization model which is directly derived from the distribution of the PDF. These results (i.e., Figs. 6.3, 6.4, and 6.5b) are thus reasonably good. However, since AA is non-convex, the global solution cannot be guaranteed, and the results may be worse than our proposed method. The SO method is a convex variation of the AA model by using the logarithmic transformation. There is still some noise in their results, especially in the case of small L. The I-divergence model is an alternative convex method. Its results are generally over-smoothed with much details vanish.

Taking the first column in Fig. 6.3 (L=4) as an example, much noise and unpleasant edges in (c) are reserved and most details of the castle in (d) are missing, while the results in (b) seem acceptable. Nevertheless, it is still slightly worse than the proposed result. For instance, we can see that the spire of the tower on the right side of the image is more clear in our result.

The above observations and analyses are also confirmed by the image quality indicator PSNR. As shown in Table 6.1, our results outperform the others with respect to all images and the number of the looks. Those show that our methods perform well on these images, and the results are of good quality with few artifacts.

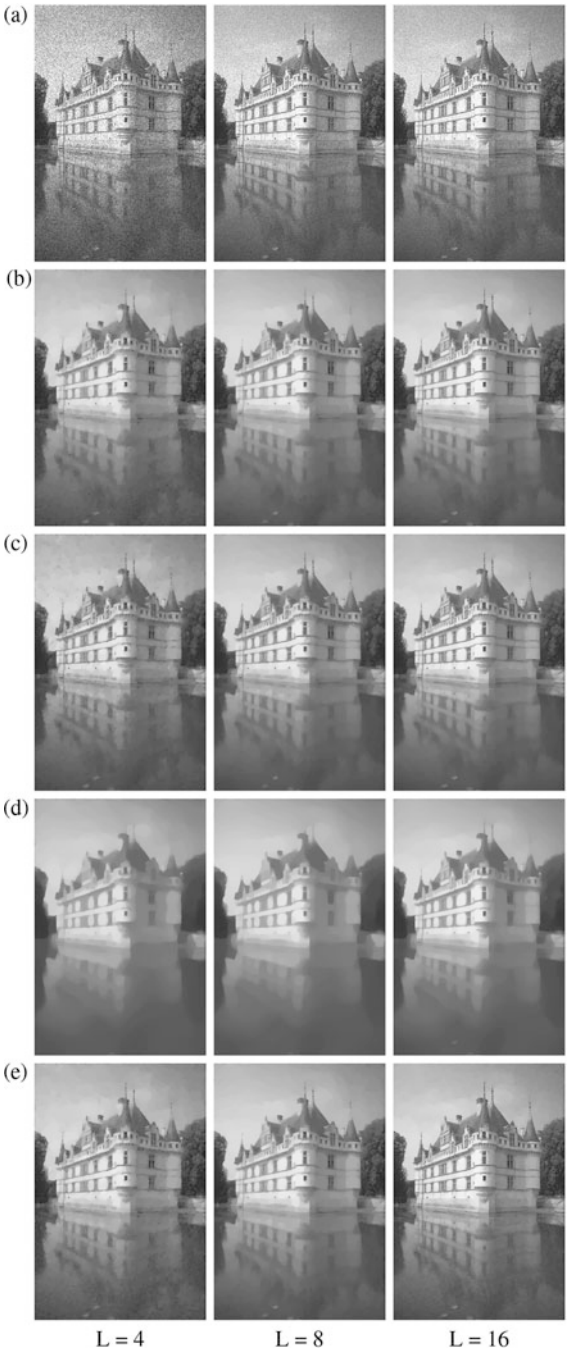


Fig. 6.3 Denoising results of different methods with different L on "Castle" image. (**a**) Noisy image. (**b**) AA. (**c**) SO. (**d**) I-divergence. (**e**) Proposed

Fig. 6.4 Denoising results of different methods with different L on "Cameraman" image. (**a**) Noisy image. (**b**) AA. (**c**) SO. (**d**) I-divergence. (**e**) Proposed

Fig. 6.5 Denoising results of different methods with different L on "Boat" image. (**a**) Noisy image. (**b**) AA. (**c**) SO. (**d**) I-divergence. (**e**) Proposed

Table 6.1 The comparisons of PSNR values by different methods

Images	L value	AA	SO	I-divergence	Proposed
Castle	4	24.85	24.39	22.75	**25.28**
	8	25.29	25.50	23.18	**27.39**
	16	28.33	27.10	27.42	**28.71**
Cameraman	4	25.89	24.91	22.90	**25.94**
	8	26.48	26.28	23.30	**27.90**
	16	27.95	28.13	24.92	**28.88**
Boat	4	26.08	25.18	23.63	**26.17**
	8	26.53	26.58	24.01	**28.15**
	16	28.04	28.36	25.48	**28.93**

The bold value is the best PSNR

Conclusions

We have introduced a convex variational model to remove the multiplicative noise. The proposed method first presents a general model based on the PDF of the speckle. And the convexity condition of the model is discussed. Using the statistical property of the noise, we then specify the proposed model as well as the convexity condition. Moreover, to deal with the minimizing problem, a Bermudez-Moreno algorithm is proposed and its convergence is analysed. Compared to other recently proposed methods, our methods appear to be reasonable and competitive.

Acknowledgements The authors would like to sincerely thank the reviewers for their valuable and constructive comments. This work is sponsored by "Chenguang Program" supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission, the key project of the National Natural Science Foundation of China (No. 61731009), the National Science Foundation of China (11271049, 61501188), RGC 12302714, and the Direct Grant for Research of the Chinese University of Hong Kong.

References

1. S. Aja-Fernandez, C. Alberola-Lopez, On the estimation of the coefficient of variation for anisotropic diffusion speckle filtering. IEEE Trans. Image Process. **15**(9), 2694–2701 (2006)
2. L. Ambrosio, N. Fusco, D. Pallara, *Functions of Bounded Variation and Free Discontinuity Problems* (Clarendon Press, Oxford, 2000)
3. F. Andreu-Vaillo, J.M. Mazón, V. Caselles, *Parabolic Quasilinear Equations Minimizing Linear Growth Functionals*, vol. 223 (Springer, Berlin, 2004)
4. F. Argenti, A. Lapini, T. Bianchi, L. Alparone, Fast MAP despeckling based on Laplacian-Gaussian modeling of wavelet coefficients. IEEE Sci. Remote Sens. Lett. **9**(1), 13–17 (2012)
5. F. Argenti, A. Lapini, T. Bianchi, L. Alparone, A tutorial on speckle reduction in synthetic aperture radar images. IEEE Geosci. Remote Sens. Mag. **1**(3), 6–35 (2013)
6. G. Aubert, J.F. Aujol, A variational approach to remove multiplicative noise. SIAM J. Appl. Math. **68**(4), 925–946 (2006)

7. G. Aubert, J. Aujol, A variational approach to removing multiplicative noise. SIAM J. Appl. Math. **68**(4), 925–946 (2008)
8. J.F. Aujol, Some first-order algorithms for total variation based image restoration. J. Math. Imag. Vis. **34**(3), 307–327 (2009)
9. A. Bermúdez, C. Moreno, Duality methods for solving variational inequalities. Comput. Math. Appl. **7**(7), 43–58 (1981)
10. A. Chambolle, An algorithm for total variation minimization and applications. J. Math. Imag. Vis. **20**(1–2), 89–97 (2004)
11. K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
12. M. Dai, C. Peng, A. Chan, D. Loguinov, Bayesian wavelet shrinkage with edge detection for sar image despeckling. IEEE Trans. Geosci. Remote Sens. **42**(8), 1642–1648 (2004)
13. C. Deledalle, L. Denis, F. Tupin, Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. IEEE Trans. Image Process. **18**(12), 2661–2672 (2009)
14. Y. Dong, T. Zeng, A convex variational model for restoring blurred images with multiplicative noise. SIAM J. Imag. Sci. **6**(3), 1598–1625 (2013)
15. J.D. Gibson, A. Bovik, *Handbook of Image and Video Processing* (Academic, Amsterdam, 2000)
16. Y. Huang, M. Ng, Y. Wen, A new total variation method for multiplicative noise removal. SIAM J. Imag. Sci. **2**(1), 20–40 (2009)
17. D. Kuan, A. Sawchuk, T. Strand, P. Chavel, Adaptive restoration of images with speckle. IEEE Trans. Acoust. Speech Signal Process. **35**(3), 373–383 (1987)
18. J. Lee, Digital image enhancement and noise filtering by use of local statistics. IEEE Trans. Pattern Anal. Mach. Intell. **2**(2), 165–168 (1980)
19. J. Lee, Refined filtering of image noise using local statistics. Comput. Vis. Graph. Image Process. **15**(4), 380–389 (1981)
20. J. Lee, A simple speckle smoothing algorithm for synthetic aperture radar images. IEEE Trans. Syst. Man Cybern. **13**(1), 85–89 (1983)
21. J. Lee, J. Wen, T. Ainsworth, K. Chen, A. Chen, Improved sigma filter for speckle filtering of SAR imagery. IEEE Geosci. Remote Sens. **47**(1), 202–213 (2009)
22. H. Li, W. Hong, Y. Wu, P. Fan, Bayesian wavelet shrinkage with heterogeneity-adaptive threshold for SAR image despeckling based on generalized gamma distribution. IEEE Trans. Geosci. Remote Sens. **51**(4), 2388–2402 (2013)
23. A. Lopes, E. Nezry, R. Touzi, H. Laur, Maximum a posteriori speckle filtering and first order texture models in SAR images, in *Geoscience and Remote Sensing Symposium* (1990), pp. 2409–2412
24. X. Ma, H. Shen, J. Yang, P. Li, Polarimetric-spatial classification of SAR images based on the fusion of multiple classifiers. IEEE J. Sel. Top. Appl. Earth Observations Remote Sens. **7**(3), 961–971 (2014)
25. S. Parrilli, M. Poderico, C. Angelino, L. Verdoliva, A nonlocal SAR image denoising algorithm based on LLMMSE wavelet shrinkage. IEEE Trans Geosci Remote Sens **50**(2), 606–616 (2012)
26. A. Pazy, On the asymptotic behavior of iterates of nonexpansive mappings in Hilbert space. Israel J. Math. **26**(2), 197–204 (1977)
27. A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, vol. 44 (Springer, Berlin, 1983)
28. J. Ranjani, S. Thiruvengadam, Dual-tree complex wavelet transform based SAR despeckling using interscale dependence. IEEE Trans. Geosci. Remote Sens. **48**(6), 2723–2731 (2010)
29. L. Rudin, S. Osher, E. Fatem, Nonlinear total variation based noise removal algorithms. Physica D **60**(1), 259–268 (1992)
30. J. Shi, S. Osher, A nonlinear inverse scale space method for a convex multiplicative noise model. SIAM J. Imag. Sci. **1**(3), 294–321 (2008)

31. G. Steidl, T. Teuber, Removing multiplicative noise by Douglas-Rachford splitting methods. J. Math. Imag. Vis. **36**(2), 168–184 (2010)
32. T. Teuber, A. Lang, Nonlocal filters for removing multiplicative noise, in *Scale Space and Variational Methods in Computer Vision* (Springer, Berlin, 2012), pp. 50–61
33. Y. Yu, S. Acton, Speckle reducing anisotropic diffusion. IEEE Trans. Image Process. **11**(11), 1260–1270 (2002)

Part III
3D Image Understanding and Classification

Chapter 7
Multi-Dimensional Regular Expressions for Object Detection with LiDAR Imaging

Todd C. Torgersen, V. Paúl Pauca, Robert J. Plemmons, Dejan Nikic, Jason Wu, and Robert Rand

Abstract Regular expressions are a fundamental technique for pattern matching in textual data and for lexical analysis in compiler design. They are ubiquitous in most systems used today, including operating systems (e.g. grep, awk), computer languages (e.g. Perl, Java, Python), and web search engines (e.g. Google). However, this highly useful way of exploring and mining data has thus far eluded non-textual datasets, such as images and 3D geometric data. Shape-based searching of 3D objects continues to be a core problem in computer vision. We propose a novel extension of traditional finite-automata-based methods to find multi-dimensional objects in spatial data sets. Our approach extends regular expressions and finite automata to multi-dimensional pattern models. While we demonstrate the effectiveness and efficiency of our approach for finding target objects in 3D LiDAR image data sets using an implicit geometry representation of the data, it is important to note that the proposed technique can be applied to any general data set of vertices in 3D space. Non-geometric information, such as material and spectral characteristics from hyperspectral image data can also be discretized and encoded into our approach.

T. C. Torgersen (✉) · V. P. Pauca
Department of Computer Science, Wake Forest University, Winston-Salem, NC, USA
e-mail: torgerse@wfu.edu; paucavp@wfu.edu

R. J. Plemmons
Department of Mathematics and Department of Computer Science, Wake Forest University, Winston-Salem, NC, USA
e-mail: plemmons@wfu.edu

D. Nikic · J. Wu
The Boeing Company, Seattle, WA, USA
e-mail: Dejan.Nikic@boeing.com; Jason.Wu@boeing.com

R. Rand
National Geo-Spatial Intelligence Agency, Springfield, VA, USA
e-mail: Robert.S.Rand@nga.mil

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_7

Introduction

Regular expressions and finite automata are powerful tools for encoding text patterns and searching for these patterns in textual data. For example, hashtags in social media messages and posts (e.g. `#OccupyWallStreet`) can be found using the notation `(^|\s)#([A-Za-z0-9_]+)`, where `(^|\s)#` denotes beginning of a line or a blank space followed by a `#` and `([A-Za-z0-9_]+)` denotes a sequence of one or more alphanumeric characters and the underscore symbol. Two important components of regular expressions are the ability to express sophisticated patterns using a prescribed notation and the capacity to search efficiently for such patterns in a given input text.

A number of retrieval methods for finding a few specific objects in 2D image data have been developed and popularized over the last decade. Face detection is one such method that is now employed by most digital photography management systems and social networks such as Facebook and Instagram. These methods typically require large amounts of training data and careful selection of feature descriptors for discerning between object classes.

However, techniques that enable the encoding of more general and complex shapes and 3-D geometries and efficient searching in high dimensional data sets, such as 3-D point clouds and meshes, are still lacking. This functionality is desirable in many applications involving 3-D data and 3-D modeling. For example, Funkhouser et al. [5] argue that in computer graphics the challenging question will shift from how to construct 3-D models to how to search for 3-D models in existing data using shape-based queries. Similarly, the need to search for specific geometries in 3-D meshes also arises in architectural design, where "searching and replacing" within a mesh or restoring object identity from a given polygon soup [21] are desirable.

In this paper, we examine the problem of efficiently searching for objects of interest in 3-D image data sets. Specifically, we are motivated to detect target objects of particular geometric shapes in 3-D LiDAR data sets [3, 6]. LiDAR is an optical imaging method that measures distance to a target by illuminating that target with a laser light and therefore captures geometric information about a scene. LiDAR, along with hyperspectral data, are the pervasive methods in remote sensing, e.g. [10]. The typical characteristics of LiDAR have also resulted in several applications which were not deemed feasible hitherto with the conventional techniques viz. mapping of transmission lines and adjoining corridors, change detection to assess damages (e.g. in buildings) after a disaster, etc. We assume here the raw LiDAR point cloud has been pre-processed, using methods such as implicit geometry [15], into a 3-dimensional data volume of (equal sized) voxels, where each voxel value comes from a finite set {0, 1, 2, . . . , s-1}.[1]

[1]Implicit geometry representation of point cloud data may be based on a number of metrics, including population, distance, and validity, see, e.g. [15]. The data presented in this paper uses a simple population metric for voxelization of the point cloud data.

Contribution of This Work

We introduce and implement a novel approach for shape-based searching in 3-D data sets based on multi-dimensional regular expressions (MDRE) and the corresponding deterministic and nondeterministic finite automata [11]. This approach consists of both a new notation for regular expressions able to encode geometry along various spatial dimensions as well as a parser implementation in C++ capable of reading such notation and searching for matching expressions in an input 3-D data set.

Related Work

The generalization of concepts and techniques of formal languages, such as regular and context-free languages, to two dimensions has been studied primarily within the theoretical computer science literature. Nirmal and Rama [16] proposed a linguistic model for the generation of 2-D pictures by the substitution of regular sets into well-known families of L-systems. Later, Giammarresi and Restivo [7] explored the notion of *two-dimensional languages* or *picture languages*, recognized by finite automata and denoted by a rectangular representation notation. They also introduced the notion of local picture languages or *tiles* leading to so-called tiling systems. Tiling rewriting grammars [18] and pure 2D picture grammars [20] extend these concepts to the context-free languages. Some of the implications in the extension to 2-D relative to the classical properties of 1D regular versus context-free languages have also been studied, see e.g. [2]. Unlike these generative models, the notation introduced in our approach generalizes regular expressions to 2-D and higher dimensions and the resulting patterns are not rectangular in nature. Moreover, our parser makes it possible to apply regular expressions in practical applications.

A technique perhaps closest to ours is that of Wurzer et al. [21] who developed a system for expressing and matching 3-D regular expressions while processing 3-D meshes generated in the course of architectural work. In their approach, a string is a sequence of characters, a mesh or a set of vertices connected by edges. They search for paths within this mesh, taking the sequence of angles between each pair of edges on that path as criterion (angular search) [21]. This is a powerful technique that is also invariant to rotations. The grammar models formalism for object detection in computer vision [8] is another related approach that uses the concept of a bag grammar for object representation.

While context free and bag grammars provide powerful models for object generation and detection, grammars present a number of difficulties including choice of parsing algorithm, NP-hardness of parsing for bag grammars [4], ambiguity, and Turing undecidability of grammar equivalence [1, 4, 19]. Regular expressions have several advantages over grammars including:

- Ambiguity is not an issue for regular expressions.
- Choice of parsing algorithm is not an issue for regular expressions. Object detection is accomplished efficiently by simulating a deterministic finite automata (DFA) on an input data set.
- Practical questions such as the equivalence of two regular expressions are decidable in polynomial time.

In addition to the provable properties above, a review of many practical examples suggests that the full expressive power of context free grammars (and their corresponding pushdown automata) is not needed for objects of practical interest. The fundamental difference between DFA and pushdown automata (PDA) is the inclusion of an unbounded stack memory in the PDA model. Target objects are usually contained in a relatively small (finite) support region, and therefore a model including unbounded stack memory is not necessary.

Readers unfamiliar with traditional regular languages are referred to [11, 19] for a review of these concepts.

Definitions and Notation

In this section, we introduce the definition of regular expressions and its formal extension to higher dimensions. Notation suitable for computer implementation is also presented.

1-D Regular Expressions and Regular Languages

Let Σ be a finite alphabet. We define a *language* to be a set of finite strings[2] over the alphabet Σ. The standard regular operations **union**, **concatenation**, and **Kleene closure** are summarized below:

Union Let L_1 and L_2 be languages over Σ. We define the *union* of two languages $L_1 \cup L_2$ as: $L_1 \cup L_2 = \{t \mid t \in L_1 \text{ or } t \in L_2\}$.

Concatenation Let L_1 and L_2 be languages over Σ. We define the *concatenation* of two languages $L_1 \cdot L_2$ as: $L_1 \cdot L_2 = \{u \cdot w \mid u \in L_1 \text{ and } w \in L_2\}$, where $u \cdot w$ denotes the concatenation of two strings u and w.

Kleene Closure Let L be a language over alphabet Σ and L^n be defined recursively by: $L^0 = \{\epsilon\}$, $L^n = L \cdot L^{n-1}$. We define the *Kleene closure* of L denoted by L^* as: $L^* = \cup_{i=0}^{\infty} L^i$.

[2]While each string in a regular language is a finite string, a regular language itself may be infinite.

Regular Expressions *Regular expressions* over alphabet Σ along with the language they represent are defined as follows:

1. The symbol ϕ is a regular expression denoting the empty language (the empty set).
2. The symbol ϵ is a regular expression denoting the language containing the empty string.
3. The symbol a where $a \in \Sigma$ is a regular expression denoting the language $\{a\}$.

If R is a regular expression, we use the notation $L(R)$ to denote the language associated with R.

1. If R and S are a regular expressions, then $(R \mid S)$ is a regular expression, denoting the language $L(R) \cup L(S)$.
2. If R and S are a regular expressions, then $(R \cdot S)$ is a regular expression, denoting the language $L(R) \cdot L(S)$.
3. If R is a regular expression, then (R^*) is a regular expression denoting the language $(L(R))^*$.

In practice, we avoid excessive use of parentheses by assigning a precedence ordering to the operations of **union**, **concatenation** and **Kleene closure** (in that order from lowest to highest), similar to the way in which addition, multiplication and exponents are used in traditional elementary algebra. Additionally, we adopt the notation a^n to denote n repetitions of the symbol a. Informally, we will also use the juxtaposition of two symbols ab to indicate $a \cdot b$.

For the purposes of object detection it is useful to have a short-hand notation for expressing a pattern which occurs at least m times, but no more than n times. Let a be a symbol. We propose the notation:

$$R = a^{m:n}$$

to denote such a range. For example, $a^{3:5}$ denotes a string of contiguous a symbols which is either length 3, or 4, or 5. Notice that this example is still regular, since we can express it as

$$R = aaa \mid aaaa \mid aaaaa$$

The symbol a could be replaced by any regular expression to allow patterns that are repeated a limited number of times.

Extending Regular Expressions to Higher Dimensions

We extend the definition of 1-D regular expressions using a naturally recursive definition. Let $R_{(n)}$ denote an n-dimensional regular expression. We formally define $R_{(n)}$ recursively in terms of a $(n-1)$-dimensional regular expressions as follows:

1. The expression $R_{(1)}$ is an 1-dimensional regular expression equivalent to a traditional regular expression, as defined in section "1-D Regular Expressions and Regular Languages".
2. The expression $R_{(n)} = [\ R_{(n-1)}\ ]$ is an n-dimensional regular expression of length one (along dimension n). The square brackets are part of the notation to enclose a sub-expression of one lower dimension. The language $L(R_{(n)})$ is the set of all n-dimensional objects which consist of a single $(n-1)$-dimensional object matching the expression $R_{(n-1)}$, and where that $(n-1)$-dimensional object is embedded in n-dimensional space.
3. If $R_{(n)}$ and $S_{(n)}$ are a n-dimensional regular expressions, then $(R_{(n)} \mid S_{(n)})$ is a regular expression, denoting the language $L(R_{(n)}) \cup L(S_{(n)})$.
4. If $R_{(n)}$ and $S_{(n)}$ are a regular expressions, then $(R_{(n)} \cdot S_{(n)})$ is a regular expression, denoting the language $L(R_{(n)}) \cdot L(S_{(n)})$.
5. If $R_{(n)}$ is a regular expression, then $({R_{(n)}}^*)$ is a regular expression denoting the language $(L(R_{(n)}))^*$.

We refer to the expressions defined above as the class of *Multi-Dimensional Regular Expressions* (MDRE).

A 2-D Example

We consider the objects illustrated in Fig. 7.1. We consider only two symbols in this example: 0 for an empty pixel, and 1 for an occupied pixel. The diagram on the left models a tower or pole that consists of a base of five occupied pixels (horizontally) and six occupied pixels (vertically). The object on the right models a structured object.

We impose an ordering on the dimensions: the horizontal direction is designated the top-level dimension, and the vertical direction is designated as the low-level dimension. The origin is the lower left corner of the image. Our regular expression

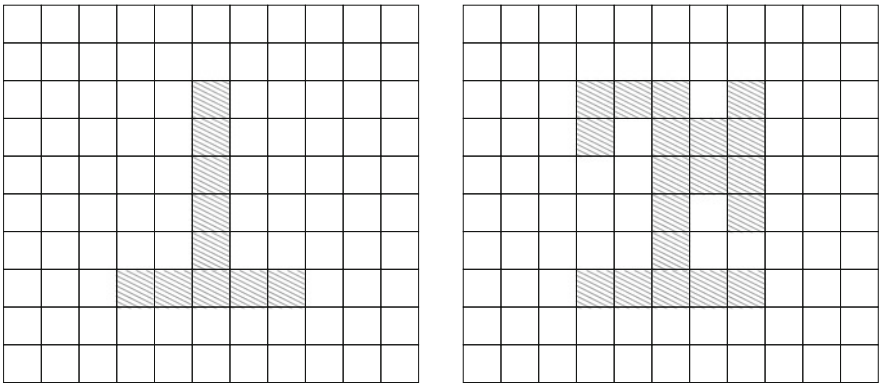


Fig. 7.1 Pole object (left) and structured object (right)

for the pole object is then:

$$R_{(2)} = [\,1\,][\,1\,][\,1\,1\,1\,1\,1\,1\,][\,1\,][\,1\,] = [\,1\,]^2 \cdot [\,1^6\,] \cdot [\,1\,]^2 \tag{7.1}$$

Notice that $R_{(2)}$ is a concatenation of 2-D regular expressions along the horizontal direction, each of which specifies 1-D regular expressions within square brackets along the vertical direction.

The regular expression (7.1) does not specify that the area surrounding the pole is not occupied. In this example, the regular expression (7.1) would also match the structured object illustrated in Fig. 7.1 (right). If we wish to be more specific regarding the surrounding area, we must build that specification into the regular expression. For example:

$$R_{(2)} = [\,1 \cdot 0\,] \cdot [\,1 \cdot 0^5\,] \cdot [\,1^6 \cdot 0\,] \cdot [\,1 \cdot 0^5\,] \cdot [\,1 \cdot 0\,] \tag{7.2}$$

Regular expression (7.2) will match the pole object in Fig. 7.1 (left), but will not match the structured object in Fig. 7.1 (right).

We can accommodate some variation in the object using our notation regarding ranges. Suppose the base is either one or two pixels on either side of the center tower, and suppose the tower height is somewhere between four and six pixels high. Our regular expression is then:

$$R_{(2)} = [\,1\,]^{1:2} \cdot [\,1^{4:6}\,] \cdot [\,1\,]^{1:2} \tag{7.3}$$

Intuitively, the concept of "symbol" in an n-dimensional regular expression is replaced by an $(n-1)$-dimensional regular expression. To match an n-dimensional regular expression with an n-dimensional input data set, we impose an ordering $d_1, d_2, \ldots d_n$ on the dimensions. As the input data is scanned in the top-level dimension, we encounter a sequence of starting points where an $(n-1)$-dimensional expression may be matched. Next, we discuss this n-dimensional regular expression matching approach in detail.

Expression Matching for n-Dimensional Objects

In two dimensions, a recursive algorithm scans along the top level dimension (e.g., horizontal). As each regular 1-dimensional expression is encountered, it is processed (vertically) using 1-dimensional techniques. This recursive approach can be generalized to an arbitrary number of dimensions. Each data dimension corresponds to a recursive level in the computation performed by an n-dimensional DFA. If the pattern extent implied by the regular expression is bounded by a constant, then the time complexity of a target search is bounded by a linear function of the number of voxels contained in the data set.

The first step in processing an n-dimensional regular expression is to construct an n-dimensional deterministic finite automata (DFA) equivalent to the given n-dimensional regular expression. We use the notation $M_{(n)}$ to denote an

n-dimensional DFA. We use the notation $L\left(M_{(n)}\right)$ to denote the language recognized by $M_{(n)}$. We now formally define an n-dimensional DFA as follows:

Base Case A 1-dimensional finite automata is a traditional DFA.[3]

General Case An n-dimensional deterministic finite automata is a 5-tuple:

$$M_{(n)} = \left(Q_{(n)}, \Sigma_{(n)}, \delta_{(n)}, q_{0_{(n)}}, F_{(n)} \right) \tag{7.4}$$

where

- $Q_{(n)}$ is a finite set of states.
- $\Sigma_{(n)}$ is a finite set of $(n-1)$-dimensional deterministic finite automata.
- δ is a function: $\delta : Q_{(n)} \times L\left(M_{(n-1)}\right) \rightarrow Q_{(n)}$ where $M_{(n-1)}$ is an $(n-1)$-dimensional DFA.
- $q_{0_{(n)}} \in Q_{(n)}$ is a starting state.
- $F_{(n)} \subseteq Q_{(n)}$ is a set of accepting states.

Fortunately, well-known 1-D construction techniques extend naturally to the higher dimensional cases. Given an n-dimensional regular expression, the construction of an equivalent n-dimensional DFA proceeds by a dimensionally recursive application of standard algorithms for converting regular expressions to equivalent DFA. A regular expression is first converted to a nondeterministic finite automata (NFA) using a construction based on the recursive definition of regular expressions. The concept of ϵ-closure [11] is used to construct a DFA which is equivalent to the intermediate NFA form. The states in the newly constructed DFA consist of sets of states from the NFA. Further details regarding 1-D construction techniques can be found in [1].

Implementation

The usual implementation of 1-dimensional DFA-based pattern matching code represents a 1-dimensional DFA as a table indexed by states (i.e., the row index) and by alphabet symbols (i.e., the column index). For an n-dimensional DFA, we take a similar table-based approach: the column index refers to one of the (finitely many) $(n-1)$-dimensional DFA which define the n-dimensional DFA. For example, in the 2-D case, the traditional operation of matching a symbol (in the 1-D case) is replaced by simulating a 1-D DFA to decide a match in the appropriate direction. The code for the n-dimensional DFA was implemented using the C++ language.

***n*-Dimensional Regular Expression Parser** To make our system practical, we also implemented a parser capable of reading n-dimensional regular expressions,

[3]The reader is referred to [19] for further details.

such as those shown in (7.1), (7.2) and (7.3), in a linearized ASCII form. For example Eq. (7.2) may be written as

```
[1 0] [1 0^5] [1^6 0] [1 0^5] [1 0]
```

Here blank spaces denote concatenation, circumflex (^) denotes repetition. In addition, vertical bar (|) denotes set union. We use the acronym MDRE to refer to our computer implementation of the ideas presented in this paper.

Target Search in a 2-D Example

We illustrate 2-D expression matching on a 256 × 256 simulated scene shown in Fig. 7.2 (left). Let the dimensions be ordered as: (vertical, horizontal). This ordering corresponds conveniently to row-column indexing where the pixels in the scene are arranged in a matrix. If we use the symbols 1 for an occupied pixel, and 0 for an un-occupied pixel, the inverted "T" target objects seen in Fig. 7.2 can be specified by the regular expression:

```
R = [ 1 1 0 ]^4 [ 1^17 0 ]^3 [ 1 1 0 ]^4
```

Alternately, the ordering on the dimensions can be chosen to be (horizontal, vertical). In this case, a regular expression appropriate to the target object is:

```
Ra = [ 1^11 ]^2 [ 0^4 1^3 0^4 ]^15 [ 0^11 ]
```

MDRE successfully finds the seven occurrences of the object specified by the regular expressions above (using either dimension ordering), and records a bounding box for each instance. The result of finding the target object is illustrated in Fig. 7.2 (right).

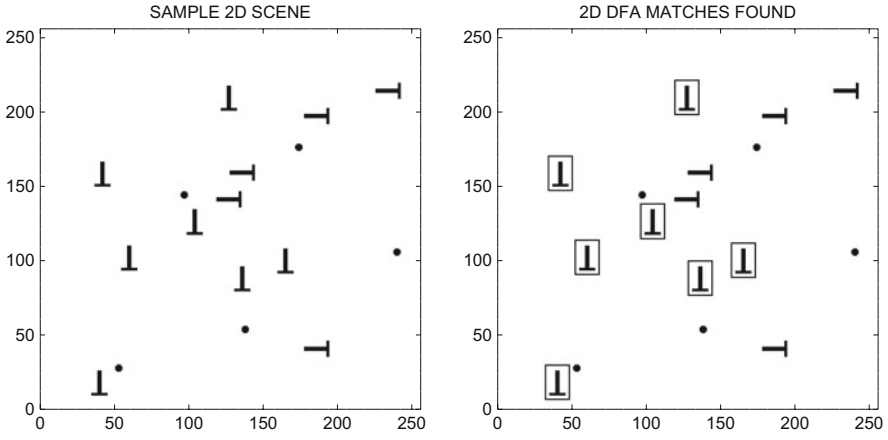


Fig. 7.2 A sample 2-D scene (left) and corresponding MDRE result (right)

Next, we apply MDRE and 3D regular expressions to search for objects of specific geometry in LiDAR data sets.

3D Regular Expressions for Implicit Geometry Data Sets

We consider a sample (real) implicit geometry data set and give a 3D regular expression for finding light poles in the observed region. This dataset is a tile from a larger collection of measurements of Ottawa, Canada obtained using ground and air-based LiDAR equipment at 15 cm resolution [14]. The regular expression given in (7.5) is best understood by visualizing a stack of squares measuring 6×6 voxels at the base and extending 9 voxels vertically. Within the 6×6 square, in the central 4×4 region at least one occupied voxel is required. Alternates (set unions denoted by "|") within the regular expression are used to accommodate sampling issues, including aliasing, noise, and variations (e.g., attachments) in the shape of the poles. Additionally, we require 1 voxel of unoccupied space surrounding the central 4×4 region. The complete expression is given in Eq. (7.5):

$$
\begin{aligned}
R_{\text{pole}} = \; & [\ [\ 0\ 0\ 0\ 0\ 0\ 0\ ]\ [\ \ 0\ (\ 0\ |\ 1\ )\hat{}4\ \ 0\ ] \\
& (\ (\ \ [\ \ 0\ (\ 0\ |\ 1\ )\ (\ 0\ |\ 1\ )\ 1\ \ (\ 0\ |\ 1\ )\ 0\ ] \\
& \quad\ [\ \ 0\ (\ 0\ |\ 1\ )\hat{}4\ \ 0\ ]\ )\ \ | \\
& \ \ (\ \ [\ \ 0\ (\ 0\ |\ 1\ )\ 1\ (\ 0\ |\ 1\ )\ \ (\ 0\ |\ 1\ )\ 0\ ] \\
& \quad\ [\ \ 0\ (\ 0\ |\ 1\ )\hat{}4\ \ 0\ ]\ \ \ )\ \ | \\
& \ \ (\ \ [\ \ 0\ (\ 0\ |\ 1\ )\hat{}4\ \ 0\ ] \\
& \quad\ [\ \ 0\ (\ 0\ |\ 1\ )\ 1\ (\ 0\ |\ 1\ )\ (\ 0\ |\ 1\ )\ 0\ ]\ \ )\ | \\
& \ \ (\ \ [\ \ 0\ (\ 0\ |\ 1\ )\hat{}4\ \ 0\ ] \\
& \quad\ [\ \ 0\ (\ 0\ |\ 1\ )\ (\ 0\ |\ 1\ )\ 1\ \ (\ 0\ |\ 1\ )\ 0\ ]\ \ )\ ) \\
& [\ \ 0\ (\ 0\ |\ 1\ )\hat{}4\ \ 0\ ] \\
& [\ 0\ 0\ 0\ 0\ 0\ 0\ ]\ \ ]\hat{}8
\end{aligned}
\tag{7.5}
$$

The current implementation converts the regular expression into an equivalent 3D DFA and performs lexical analysis (pattern finding) by simulating the 3D DFA on an IG data set. Figures 7.3 and 7.4 illustrate the light poles identified by this process.

We now turn our attention to a more difficult target: trees. The issue here is that there is no obvious or easily defined pattern that we can definitively declare to be the unambiguous characterization of a tree. On examining a few LiDAR/IG data sets, it is clear that the lower portion of tree trunks are not well represented in the 3D data sets, apparently because the trunk is mostly occluded by the upper branches and leaves. A 3D regular expression for recognizing trees is given in (7.6). The basic idea is to look for a region in which:

- most (but not all) voxels in the region are occupied, and
- the region is an appropriate distance from the ground.

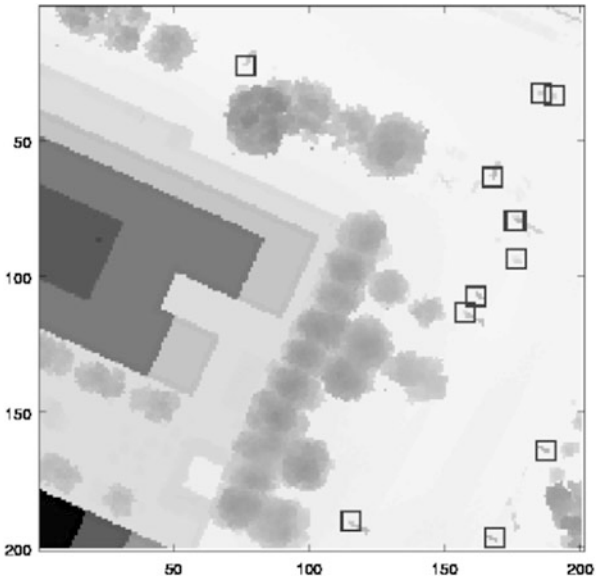


Fig. 7.3 Light poles recognized in an IG data set using MDRE

The structure of the expression in (7.6) is similar to the structure of (7.5): the expression is organized as a stack of 9 × 9 squares. The initial portion of (7.6), containing `[ [ ( 0 | 1 ) ] ]^6`, acts to lift the stack of 9 × 9 squares 6 voxel units above the ground level. The remaining portion of (7.6) uses alternates (set unions) and concatenation to specify three consecutive regions (in the vertical direction):

1. a 5 × 5 central region in which at least 3/5 of the voxels are occupied,
2. a 3 × 3 central region in which at least 1/3 of the voxels are occupied, and
3. a 9 × 9 unoccupied region.

A number of patterns could be defined which could use alternate characterizations of the appearance of trees in IG data sets.

A sample IG scene together with the identified trees is illustrated in Fig. 7.5. We observe that not all trees are found in this example. The 3D regular expression can be made more inclusive (i.e., allow more alternates) in an effort to recognize more trees. However, patterns that become too inclusive tend to falsely identify other scene objects as trees. Preliminary investigation suggests that our 3D regular expressions can be modified to incorporate probabilistic measures to better allow for natural variations in the target objects and to avoid classification errors due to noise. Probabilistic measures also hold the possibility of allowing us to simplify the regular expressions: variations in the target object will be built in to the probabilistic framework rather than a large number of alternates within the regular expression itself. An example tree recognized by the MDRE expression in Eq. (7.6) is shown in Fig. 7.6.

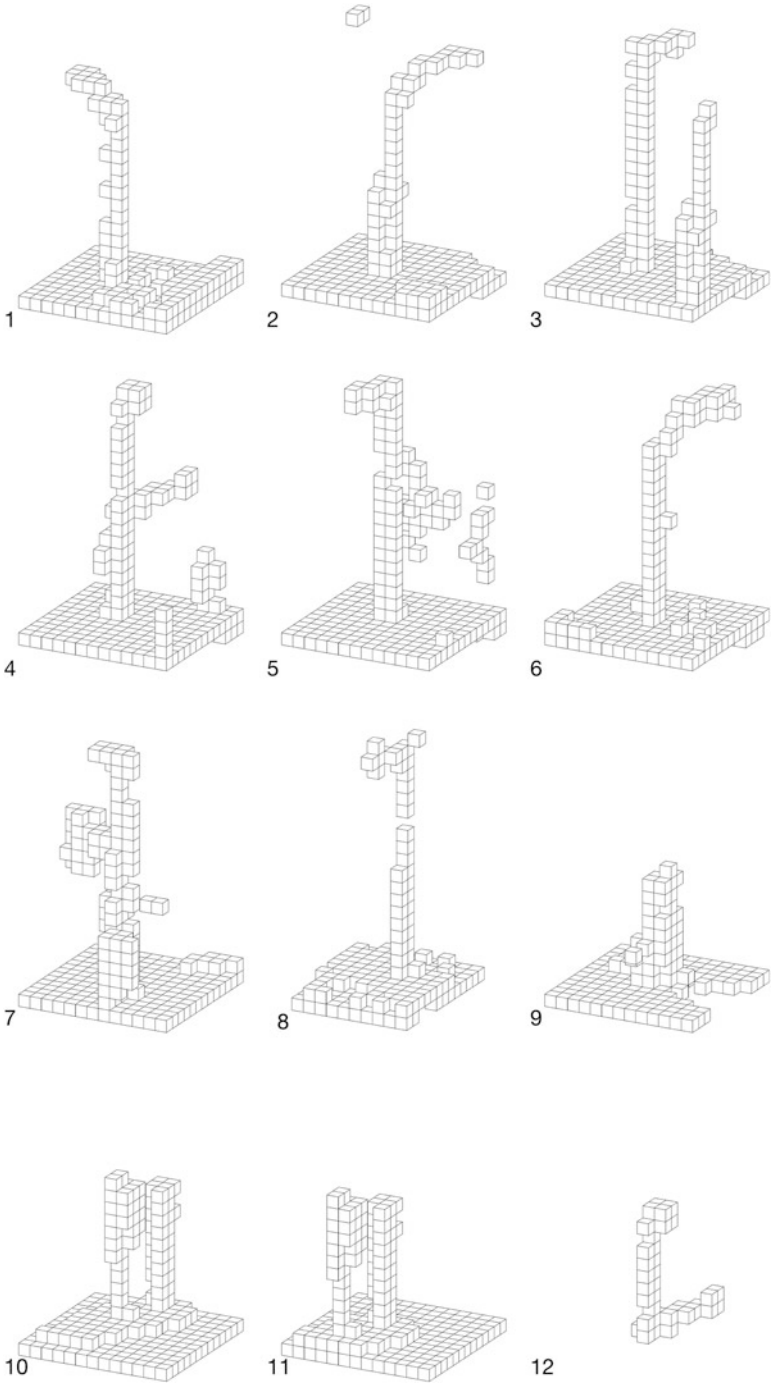


Fig. 7.4 Sample light pole structures recognized using MDRE. Notice that the structure in the bottom right is a false positive

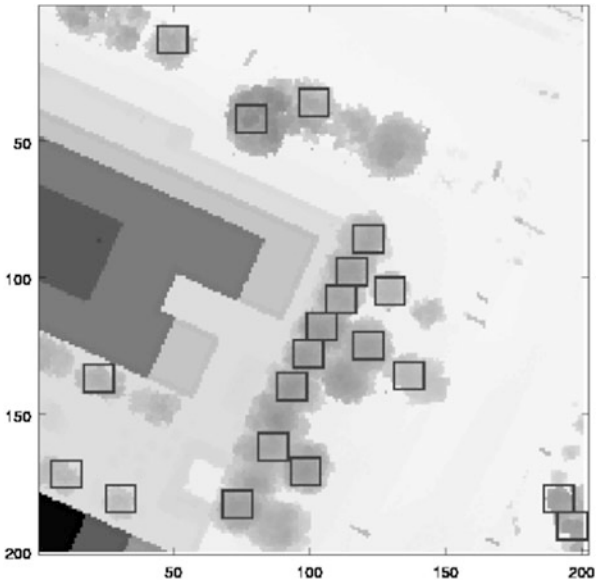


Fig. 7.5 Trees recognized in an IG data set using MDRE

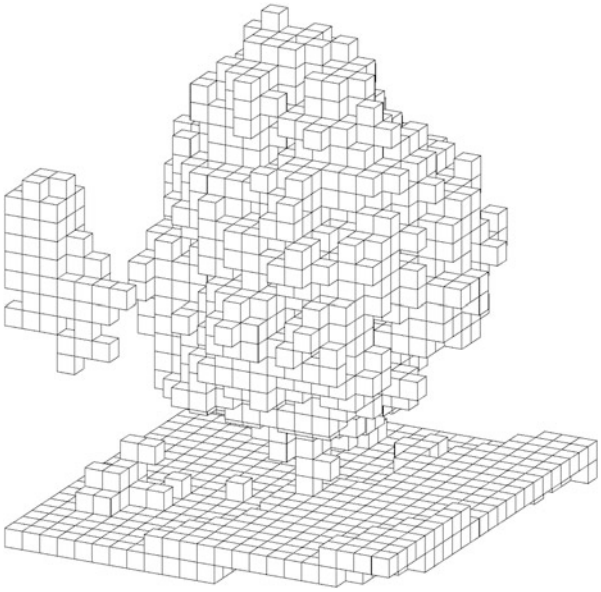


Fig. 7.6 Sample tree recognized using MDRE

$$
\begin{aligned}
R_{\text{tree}} = {} & \texttt{[[(0 | 1)]]\^{}6} \\
& \texttt{[} \\
& \quad \texttt{[0 (0 | 1)\^{}7 0]} \\
& \quad \texttt{[(0 | 1)\^{}9]} \\
& \quad \texttt{[(0 | 1)\^{}2} \\
& \quad\quad \texttt{(1\^{}5 | 0 1\^{}4 | 1 0 1\^{}3 | 1 1 0 1 1 | 1\^{}3 0 1 | 1\^{}4 0 |} \\
& \quad\quad\quad \texttt{0 0 1 1 1 | 0 1 0 1 1 | 0 1 1 0 1 | 0 1 1 1 0 |} \\
& \quad\quad\quad \texttt{1 0 0 1 1 | 1 0 1 0 1 | 1 0 1 1 0 | 1 1 0 0 1 |} \\
& \quad\quad\quad \texttt{1 1 0 1 0 | 1 1 1 0)} \\
& \quad\quad \texttt{(0 | 1)\^{}2]\^{}5} \\
& \quad \texttt{[(0 | 1)\^{}9]} \\
& \quad \texttt{[0 (0 | 1)\^{}7 0]\^{}2} \\
& \texttt{]} \\
& \texttt{[} \\
& \quad \texttt{[0 0 0 (0 | 1)\^{}3 0 0 0]} \\
& \quad \texttt{[0 0 (0 | 1)\^{}5 0 0]} \\
& \quad \texttt{[0 (0 | 1)\^{}7 0]} \\
& \quad \texttt{[((0 | 1)\^{}3} \\
& \quad\quad \texttt{(1\^{}3 | 0 1 1 | 1 0 1 | 1 1 0 | 0 0 1 | 0 1 0 | 1 0 0)} \\
& \quad\quad \texttt{(0 | 1)\^{}3)} \\
& \quad \texttt{]\^{}3} \\
& \quad \texttt{[0 (0 | 1)\^{}7 0]} \\
& \quad \texttt{[0 0 (0 | 1)\^{}5 0 0]} \\
& \quad \texttt{[0 0 0 (0 | 1)\^{}3 0 0 0]} \\
& \texttt{]} \\
& \texttt{[} \\
& \quad \texttt{[0 (0 | 1)\^{}7 0]} \\
& \quad \texttt{[(0 | 1)\^{}9]\^{}7} \\
& \quad \texttt{[0 (0 | 1)\^{}7 0]} \\
& \texttt{]} \\
& \texttt{[[0\^{}9]\^{}9]}
\end{aligned}
\tag{7.6}
$$

Hamming Distance for Deterministic Finite Automata (DFA)

A regular expression may fail to match an input data set due to errors (noise) in the LiDAR data set. We observe that for the implicit geometry data sets studied here, insertion errors and deletion errors do not occur because of the method by which an implicit geometry representation is computed from a raw point-cloud data set. For this reason, we use the Hamming distance measure instead of the Levenshtein

distance [12, 13]. I.e., we consider only substitution errors. Mismatches between a multidimensional DFA and the input data are detected only during 1D DFA matching. For this reason, it is sufficient to define Hamming distance between a DFA and a string in the 1D case.

To further simplify our analysis, we consider regular expressions which do not include the Kleene closure operation[4]; the resulting DFA graphs contain no cycles.

Let x denote a 1D input string, and let

$$S(M, x) = \{w \mid w \in L(M) \text{ and } |w| = |x|\}$$

where $L(M)$ denotes the language accepted by DFA M. We define the Hamming distance between a string x and a DFA M to be:

$$d(x, M) = \begin{cases} \min\limits_{w \in S(M,x)} h(x, w) & \text{if } S(M, x) \neq \phi \\ +\infty \quad \text{otherwise} \end{cases} \tag{7.7}$$

where $h(x, w)$ is the Hamming distance between strings x and w.

We can use Eq. (7.7) to accept a (noisy) string x if and only if

$$d(x, M) \leq \tau \tag{7.8}$$

for some (nonnegative integer) threshold τ of tolerable error count. Computing $d(x, M)$ is not as trivial as one would hope. An algorithm to find the minimum number of substitutions needed for acceptance can not proceed by performing a substitution at the point where the first symbol in error is detected. It is possible that a fault in a previously scanned symbol, e.g., symbol b in position t, caused the DFA to transition to an incorrect state q. Continuing from state q, several symbols may match correctly, followed by a large number of mis-matched symbols. However, correcting the single erroneous symbol in position t could lead to a state q' from which no further errors are encountered.

At first glance, it would appear that backtracking is necessary, leading to a computationally inefficient method. Fortunately, for a given tolerance τ, we can construct a polynomial time algorithm for deciding Eq. (7.8) without resorting to backtracking.

Hamming Automata

Given a string of symbols w and a positive integer $\tau < |w|$, the corresponding Levenshtein automata accepts any string y which can be formed from w using τ

[4]In practice the Kleene operation is rarely helpful to express a pattern, since objects of interest are never infinitely extensible.

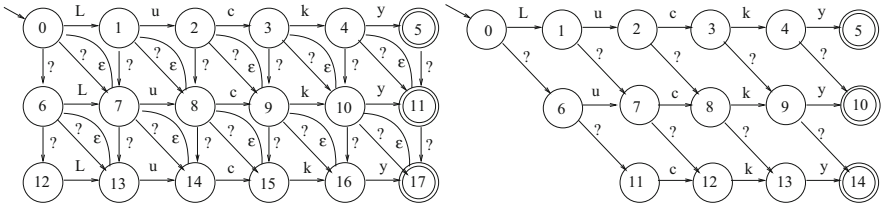


Fig. 7.7 Example Levenshtein automata (left) and Hamming automata (right)

or fewer insertion, deletion and substitution operations (i.e., within Levenshtein distance τ). A simple modification of the Levenshtein idea can be used to limit editing operations to substitution errors only, resulting in an automata which will accept any string within a chosen Hamming distance. We propose the term **Hamming Automata** to refer to a finite automata which accepts all strings y which are within Hamming distance τ of string w. A Levenshtein automata (left) and a Hamming automata (right) are shown in Fig. 7.7 for the string $w =$ "Lucky" with $\tau = 2$. A transition labeled by a question mark (?) denotes a transition on any symbol, except those symbols explicitly shown. Observe that while the Levenshtein automata is an NFA with ϵ-transitions, the Hamming automata is a simple DFA.

A polynomial time algorithm to decide Eq. (7.8) is given below.

Input: A DFA M, a string x and a positive integer τ.

Output: "Accept" if $d(x, M) \leq \tau$, "reject" otherwise.

Method:

1. Construct a Hamming automata H for input string x and tolerance τ.
   Note: This construction can be done in time $\mathcal{O}(|x|\tau)$.

2. If $L(M) \cap L(H)$ is non-empty, then accept x, otherwise reject.

The implementation of step 2 above is straightforward. Given two DFA M_1 and M_2 with n_1 and n_2 states respectively, a DFA $\widehat{M}$ recognizing $L(M_1) \cap L(M_2)$ can be constructed in time $\mathcal{O}(n_1 n_2)$ using a well known algorithm; see [19]. Depth first search of a DFA (treated as a directed graph) may be used to decide if there exists a path from the start state to any accepting state.

Comparison to Other Methods

A statistically robust comparison of MDRE to other methods is beyond the scope of this paper. However, some insight can be gained by a closer examination of the pole finding example shown in section "3D Regular Expressions for Implicit Geometry Data Sets".

Figure 7.4 shows an isolated 3-D view of the poles found in Fig. 7.3. We observe in Fig. 7.4(4) that there is sufficient bulk midway up the pole to match the base portion of the pattern in Eq. (7.5). Observe the upper portion of the pole in Fig. 7.4(4) is identical to the region shown in Fig. 7.4(12). We consider Fig. 7.4(12) a false positive. Also, in Fig. 7.4(3) a second pole is seen near the edge of the region; the shape of the second pole does not match any of the other found structures. We consider Fig. 7.4(3) to indicate a false negative. This false negative is caused by the close proximity of occupied voxels near the bases. In the current implementation, the search is restarted (after finding a match) at the next voxel beyond the matched region. In our pole-finding example, the accuracy of our method appears competitive with the 90% accuracy reported in [17] for finding pole-like objects using linear discriminant analysis and support vector machines.

In the special case of Fig. 7.4(12), the false positive can be eliminated by a simple post processing step which compares the LiDAR range of the base pattern to the range of the ground plane. We also observe that two closely spaced poles are resolved in Fig. 7.4(10) and (11).

Discussion

The MDRE approach presented here contrasts sharply with traditional classification methods such as linear discriminant analysis (LDA) and support vector machines (SVM). Foremost, MDRE is based on formal language theory and is inherently *discrete*. In contrast, both LDA and SVM rely on continuous mathematics and optimization principles. In the case of LDA, the within-class scatter matrix may be ill-conditioned, leading to numerical instability. No such numerical difficulties exist for MDRE.

Support vector machines depend critically on the choice of a kernel function. For many data sets there are few intuitive clues as to which kernel function best serves the intended purpose. Often, a "popular" kernel function is chosen solely because it has performed adequately on some (other) classification problem. Writing MDRE expressions requires some 3-D pre-visualization skills on the part of the user. However, thinking in terms of "stacks of 2-D slices" is often sufficient to construct an effective MDRE.

Both LDA and SVM benefit from being mature techniques with very broad applicability to many types of classification problems. They have the advantage that a practical classifier can often be algorithmically constructed given sufficient training data. In its current form, MDRE relies on a user-specified expression which the authors acknowledge can be somewhat tedious to develop. While algorithmic construction of multi-dimensional regular expressions from training data has not yet been demonstrated, we believe the metrics described in section "Hamming Distance for Deterministic Finite Automata (DFA)" can be used with evolutionary algorithms to combine MDRE with supervised learning.

MDRE differs from other formal language based approaches such as bag grammars in a number of important ways. By restricting the underlying model to (multi-dimensional) *regular* languages, many of the computational difficulties associated with bag grammars can be avoided. For example, if $\mathscr{G}$ denotes a bag grammar, deciding the question whether $w \in L(\mathscr{G})$ for input w is known to be NP-complete [4]. Further, the equivalence question for two grammars is Turing undecidable [19]. In contrast, all relevant questions regarding regular languages can be decided in polynomial time.

Another fundamental strength of grammars is that they can represent nested structures to an arbitrary depth (e.g., nested loops in a programming language). However, man-made objects generally have no such recursive structures. For example, a bicycle consists of parts such as: frame, seat, handlebar, and wheels. In turn, wheels consist of parts such as: rim, spokes, and axle. Also the various parts and sub-parts form a n-way tree structure corresponding to the target object; grammars easily represent such a structure.

For man-made objects, the part/sub-part tree structure is relatively shallow and never involves recursion. In our bicycle example, we observe that rims, spokes, and axles do not recursively contain seats or handlebars. This observation strongly suggests that *regular* patterns are sufficient for describing many man-made objects. Natural fractal patterns are an exception to the regularity of man-made objects.

Conclusions

We have presented a novel approach for efficient representation and detection of geometric objects in n-dimensional space by means of n-dimensional regular expressions and their corresponding finite automata. A fundamental advantage of regular expressions over context free or bag grammars is that practical questions regarding regular languages can be answered algorithmically in polynomial time. In contrast, some practical questions relating to context free and bag grammars, e.g., equivalence of two context free grammars is Turing undecidable [19].

Higher order regular and context free languages have been studied in the literature, specifically within the field of theoretical computer science, see e.g. [2, 16, 20]. Our work here includes representation methods suitable for direct computer processing. The language of multi-dimensional regular expressions as illustrated in Eqs. (7.5) and (7.6) is itself context free.

The authors acknowledge that writing patterns such as Eqs. (7.5) and (7.6) "by hand" may be tedious for some target objects. An essential point of the work presented here is to serve as an efficient low-level computer based representation of geometric patterns, comparable to the role of assembly language for representing computer programs. In practice, higher level tools are needed to enable users to more easily generate multi-dimensional patterns. The authors believe the framework presented here provides several opportunities. For example, machine learning

techniques may succeed at discovering multi-dimensional regular expressions from a set of training data.

Future Work

We outline a few directions in which the work presented here may be extended for object detection and retrieval purposes.

1. Performance improvements may be possible by adapting the concepts used by the Knuth-Morris-Pratt algorithm [9].
2. Fused image datasets, such as co-registered LiDAR and HSI datasets [6] may be processed by extending the alphabet of the n-dimensional DFA to also include a larger but finite set of features representative of the fused dataset.
3. Scale invariance may be introduced into the object detection algorithm by applying n-dimensional DFAs at various spatial scales and selection scale automatically by minimizing the Hamming distance. Similarly, rotation invariance may be introduced in several ways. For example, one may first detect the principal direction using a histogram of gradients approach for a region of interest and then rotating the coordinate space appropriately to match the regular expression.
4. Hamming automata may be extended to provide a distance metric between two n-dimensional DFAs (or regular expressions). Discrete optimization techniques, often used in machine learning approaches, may utilize such distance criteria for *learning* n-dimensional regular expressions from training data.

Acknowledgements This research was supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant no. FA9550-15-1-0286 and by the U.S. National Geospatial-Intelligence Agency (NGA) under Contract HM1582-10-C-0011, public release number PA Case 13-192.

References

1. A.V. Aho, J.D. Ullman, *Principles of Compiler Design* (Addison-Wesley, Boston, 1977)
2. A. Cherubini, S.C. Reghizzi, M. Pradella, P. San Pietro, Picture languages: tiling systems versus tile rewriting grammars. Theor. Comput. Sci. **356**, 90–103 (2006)
3. M. Dalponte, L. Bruzzone, D. Gianelle, Fusion of hyperspectral and LiDAR remote sensing data for classification of complex forest areas. IEEE Trans. Geosci. Remote Sens. **46**(5), 1416–1427 (2008)
4. P.F. Felzenszwalb, D. McAllester, Object detection grammars, in *ICVV Workshops* (2011), p. 691
5. T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A search engine for 3d models. ACM Trans. Graph. **22**(1), 83–105 (2003)
6. P. Gader, A. Zare, R. Close, G. Tuell, Co-registered hyperspectral and LiDAR long beach, Mississippi data collection. University of Florida, University of Missouri, and Optech International, 2010

7. D. Giammarresi, A. Restivo, Two-dimensional languages, in *Handbook of Formal Languages*, vol. 3, ed. by G. Rozenberg, A. Salomaa (Springer New York, New York, 1997), pp. 215–267
8. R.B. Girshick, P.F. Felzenszwalb, D.A. McAllester, Object detection with grammar models, in *Advances in Neural Information Processing Systems 24*, ed. by J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, K.Q. Weinberger (Curran Associates, Red Hook, 2011), pp. 442–450
9. D. Knuth, J. Morris, V. Pratt, Fast pattern matching in strings. SIAM J. Comput. **6**(2), 323–350 (1977)
10. S. Lavender, A. Lavender, *Practical Handbook of Remote Sensing* (CRC Press, Boca Raton, 2015)
11. M.V. Lawson, *Finite Automata* (CRC Press, Boca Raton, 2003)
12. V. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals. Sov. Phys. Dokl. **10**(8), 707–710 (1966)
13. G. Navarro, A guided tour to approximate string matching. ACM Comput. Surv. **1**, 31–38 (2001)
14. Neptec, Wright state 100 (2009), www.daytaohio.com/Wright.State100.php
15. D. Nikic, V. Pauca, R. Plemmons, J. Wu, P. Zhang, A novel approach to environment reconstruction in LiDAR and HSI datasets, in *AMOS Technical Conference Proceedings*, Maui (2012)
16. N. Nirmal, R. Rama, Picture generation and developmental matrix systems. Comput. Vis. Graphics Image Process. **43**(1), 67–80 (1988)
17. C. Ordóñez, C. Cabo, E. Sanz-Ablanedo, Automatic detection and classification of pole-like objects for urban cartography using mobile laser scanning data. Sensors **1465**(17), 1–10 (2017)
18. S.C. Reghizzi, M. Pradella, Tile rewriting grammars and picture languages. Theor. Comput. Sci. **340**(2), 257–272 (2005)
19. M. Sipser, *Introduction to the Theory of Computation*, 2nd edn. (Thomson, Boston, 2006)
20. K.G. Subramanian, R.M. Ali, M. Geethalakshmi, A.K. Nagar, Pure 2D picture grammars and languages. Discret. Appl. Math. **157**, 3401–3411 (2009)
21. G. Wurzer, B. Martens, K. Bühler, 3d regular expressions - searching shapes in meshes, in *eCAADe 2013: Computation and Performance*, vol. 2 (2013), pp. 279–288

Chapter 8
Relaxed Optimisation for Tensor Principal Component Analysis and Applications to Recognition, Compression and Retrieval of Volumetric Shapes

Hayato Itoh, Atsushi Imiya, and Tomoya Sakai

Abstract The mathematical and computational backgrounds of pattern recognition are the geometries in Hilbert space used for functional analysis and the applied linear algebra used for numerical analysis, respectively. Organs, cells and microstructures in cells dealt with in biomedical image analysis are volumetric data. We are required to process and analyse these data as volumetric data without embedding into higher-dimensional vector spaces from the viewpoint of object-oriented data analysis. Therefore, sampled values of volumetric data are expressed as three-way array data. The aim of the paper is to develop relaxed closed forms for tensor principal component analysis (PCA) for the recognition, classification, compression and retrieval of volumetric data. Tensor PCA derives the tensor Karhunen-Loève transform, which compresses volumetric data, such as organs, cells in organs and microstructures in cells, preserving both the geometric and statistical properties of objects and spatial textures in the space.

Introduction

For computer assisted diagnosis, inspection and biopsy in precision medicine, abnormality detection based on pattern recognition is a fundamental technique. From cells to the human body, the medical data used for diagnosis are multi-way

H. Itoh (✉)
Graduate School of Informatics, Nagoya University, Nagoya, Japan
e-mail: hitoh@mori.m.is.nagoya-u.ac.jp

A. Imiya
Institute of Management and Information Technologies, Chiba University, Chiba, Japan

T. Sakai
Graduate School of Engineering, Nagasaki University, Nagasaki, Japan

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_8

array data. Organs, cells in organs and microstructures in cells, which are dealt with in biomedical image analysis, possess statistical properties in the form of spatial textures. These biological objects also possess volumetric structures with spatial geometric and topological properties in the form of three-dimensional objects [10, 12, 26, 28, 30]. Although their local volumetric structures are computed from geometric and topological properties, their textures are used to estimate both local and global statistical properties of these objects. Since organs are essentially spatial textures defined in finite regions, the outer boundaries of these regions define the shapes of the organs. For the data analysis of these volumetric data, methods which simultaneously process geometrical and topological structures and spatial texture properties are required.

A pattern is assumed to be a square integrable function in a linear space and to be defined on a finite support in a higher-dimensional Euclidean space [6, 9, 25]. For planar and volumetric patterns, the dimensions of the Euclidean spaces are two and three, respectively. For the achievement of pattern recognition by numerical computation, sampled patterns are dealt with. In traditional pattern recognition, these sampled patterns are embedded in an appropriate-dimensional Euclidean space as vectors. The multi-way array is the other way to deal with sampled patterns. These multi-way array data are expressed as tensors [4, 15, 16, 18, 20, 22] to preserve the linearity of the original pattern space since tensors express three-way array data in multilinear forms. Therefore, multi-way principal component analysis (PCA) of tensor data is used to extract features from multi-dimensional objects for pattern recognition, classification, compression and data retrieval.

We apply three-way PCA to volumetric data analysis in biomedical information processing. For three-way PCA, we developed a relaxed closed form of tensor PCA computation based on Tucker-3 tensor decomposition, although Tucker-3 tensor decomposition [4, 15, 17] is achieved by solving variational optimisation problems iteratively. Our method solves a system of variational optimisation problems derived from the original Tucker-3 decomposition with the orthogonal constraints for solutions. We also numerically clarified that data compression by the discrete cosine transform (DCT) [27] efficiently approximates the data compression procedure based on tensor PCA since the DCT approximates the Karhunen-Loève (K-L) transform [7, 24]. This method is used for compression and retrieval of volumetric data preserving the volumetric structures with the spatial geometries and statistical properties of shapes [5, 32].

These orthogonal-projection-based data compression methods for three-way data arrays extract outline volumetric shapes [31]. Mathematically, a shape is a finite closed region in a Euclidean space. The boundaries of planar and volumetric shapes are closed simple planar curves and closed simple two-dimensional manifolds, respectively. An outline shape is a smoothed profile of a shape. For a planar shape, an outline shape is generated by smoothing the boundary contour of the shape. For a volumetric shape, an outline shape is generated by smoothing the closed boundary manifold of the shape. These properties imply that outline shapes are generated as smoothed approximations of the original shapes. Outline shapes of volumetric images of organs provide fundamental features for information filtering in medical

diagnosis and data retrieval. Furthermore, if a shape is expressed as a series expansion using base functions, an outline of the shape is a finite truncation of this series expansion of the shape. We introduce a basis system which simultaneously extracts both the outline shape of an object and global statistical properties of the interior texture of the object.

We are required to process and analyse volumetric data as three-way array data without embedding sampled values in vector space from the viewpoint of the object-oriented data analysis (OODA) [21]. We derive the tensor subspace and mutual subspace for tensors using tensor PCA based on the Tucker-3 tensor decomposition. The mutual tensor subspace method is stable against geometric perturbation of queries for pattern recognition since the method assumes that a query is an element of a low-dimensional tensor subspace.

Principal Component Analysis and Pattern Recognition

Subspace Method for Pattern Recognition

A volumetric pattern is assumed to be a square integrable function in a linear space and to be defined on a finite support in three-dimensional Euclidean space [6, 9, 25] such that

$$\int_{\Omega} |f|^2 d\boldsymbol{x} < \infty \tag{8.1}$$

for $\Omega \subset \mathbb{R}^3$. Furthermore, we assume

$$\int_{\Omega} |\nabla f|^2 d\boldsymbol{x} < \infty \tag{8.2}$$

$$\int_{\Omega} tr\{(\nabla\nabla^\top f)^\top (\nabla\nabla^\top f)\} d\boldsymbol{x} < \infty, \tag{8.3}$$

where $\nabla\nabla^\top f$ is the Hessian matrix of f. The collection of these functions defines the Hilbert space $\mathfrak{H}$.

Setting (f, g) to be the inner product in $\mathfrak{H}$, the relation $|f|^2 = (f, f)$ is satisfied. For the orthogonal projection $\boldsymbol{P}_\perp = \boldsymbol{I} - \boldsymbol{P}$, $f^{\parallel} = \boldsymbol{P} f$ and $f^{\perp} = \boldsymbol{P}_\perp f$ are the canonical element and canonical form of f with respect to $\boldsymbol{P}$ and $\boldsymbol{P}_\perp$, respectively, where $\boldsymbol{I}$ is the identity operation in $\mathfrak{H}$. If $\boldsymbol{P}$ is the projection to the space spanned by the constant element, the operation $\boldsymbol{P}_\perp f$ is called the constant canonicalisation. Let $\boldsymbol{P}_i$ be the orthogonal projection to the linear subspace corresponding to category $\mathscr{C}_i \subset \mathfrak{H}$. For a pattern f, if $|\boldsymbol{P}_{i*}(f/|f|)| \leq \delta$ for an appropriately small positive number δ, we conclude that $f \in \mathscr{C}_{i*}$.

Let θ be the canonical angle between a pair of linear subspaces L_1 and L_2. Setting $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ to be the orthogonal projections to L_1 and L_2, respectively, $\cos^2\theta$ is the maximiser of $(\boldsymbol{P}_1 f, \boldsymbol{P}_2 g)^2$ with respect to the conditions $|f| = 1$, $|g| = 1$, $\boldsymbol{P}_1 f = f$ and $\boldsymbol{P}_2 g = g$. The relation $\cos^2\theta = \lambda^2_{\max}$ is satisfied, where $\lambda_{\max}$ is the maximal singular value of $\boldsymbol{P}_2\boldsymbol{P}_1$.

Since, in the mutual subspace method (MSM) [19], a query f is expressed by using a set of local bases, we set $\boldsymbol{Q}_f$ to be the orthogonal projection to the linear subspace expressing query f. Then, if the canonical angle between $\boldsymbol{Q}_f$ and $\boldsymbol{P}_i$ satisfies the relation $\angle(\boldsymbol{Q}_f, \boldsymbol{P}_i) < \angle(\boldsymbol{Q}_f, \boldsymbol{P}_i^*)$ for all $\mathscr{C}_i$, we conclude that $f \in \mathscr{C}_{i*}$.

Setting $\boldsymbol{\delta}$ and ε to be a small vector and a small positive number, respectively, we have the relation

$$|f(\boldsymbol{x}+\boldsymbol{\delta}) - (f(\boldsymbol{x}) + \boldsymbol{\delta}^\top \nabla f + \frac{1}{2}\boldsymbol{\delta}^\top(\nabla\nabla^\top f)\boldsymbol{\delta})| < \varepsilon, \tag{8.4}$$

for local geometric perturbations. For $n = 3$, all f, f_x, f_y, f_z, f_{xx}, f_{yy}, f_{zz}, f_{xy}, f_{yz} and f_{zx} are independent if f is not sinusoidal in each direction. Therefore, Eq. (8.4) implies that, for a pattern defined in two- and three-dimensional Euclidean spaces, the local dimensions of the pattern are four and ten, respectively, if the local geometric perturbations and local bending deformation of the pattern are assumed as local transformations of the pattern. This property of the local dimensionality allows us to establish the MSM, which deals with a query as a pattern in a subspace [8].

Setting $\boldsymbol{P}_i$ to be the orthogonal projection to linear subspace $\mathfrak{L}_i$ corresponding to category C_i, we compute the orthogonal projection $\boldsymbol{Q}$ which maximises the criterion

$$I(\boldsymbol{Q}) = \sum_{i=1}^{n} |\boldsymbol{Q}\boldsymbol{P}_i|_2^2 \tag{8.5}$$

with respect to the condition $\boldsymbol{Q}^*\boldsymbol{Q} = \boldsymbol{I}$, where $\boldsymbol{Q}^*$ is the conjugate of $\boldsymbol{Q}$ and $|\boldsymbol{A}|$ is the trace norm of the operator $\boldsymbol{A}$ in Hilbert space $\mathfrak{H}$. The minimisation problem

$$J(\boldsymbol{Q}) = \sum_{i=1}^{n} |\boldsymbol{Q}\boldsymbol{P}_i|_2^2 + tr(\boldsymbol{\Lambda}(\boldsymbol{I} - \boldsymbol{Q}^*\underline{\mathrm{Q}})) \tag{8.6}$$

derives the eigen-operator problem

$$\left(\sum_{i=1}^{n} \boldsymbol{P}_i\right)\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{\Lambda}. \tag{8.7}$$

Though operation $\boldsymbol{Q}f$ removes the common part for all categories from f, $(\boldsymbol{I}-\boldsymbol{Q})f$ essentially preserves significant parts for pattern recognition of f.

For f and g in $\mathfrak{H}$, we define the metric d for $\mu(f)$ and $\mu(g)$ as $d(\mu(f), \mu(g))$ using an appropriate one-to-one transform μ from $\mathfrak{H}$ to its subset. Furthermore,

using an appropriate mapping Φ, we define the measure

$$s(f, g) = \Phi(d(\mu(f), \mu(g))). \tag{8.8}$$

If we set $\mu(f) = \frac{f}{|f|}$ and set d and Φ as the geodesic distance on the unit sphere in $\mathfrak{H}$ and $\Phi(x) = \cos x$, respectively, $s(f, g)$ becomes the similarity measure based on the angle between f and g. For $f' = f + \delta_f$ and $g' = g + \delta_g$, setting

$$\min(|f|, |g|) = \Lambda, \quad \max(\delta_f, \delta_g) = \Delta, \tag{8.9}$$

we have the relation

$$\left| \left(\frac{f'}{|f'|}, \frac{g'}{|g'|} \right) - \left(\frac{f}{|f|}, \frac{g}{|g|} \right) \right| = c \frac{\Delta}{\Lambda}, \tag{8.10}$$

for a positive constant c. Therefore, $s(f, g)$ is stable and robust against perturbations and noises in f and g.

For patterns in $\mathfrak{H}$, we have the following property.

Property 8.1 For $|f| = 1$ and $|g| = 1$, assuming $|f - g| \leq \frac{1}{3} \cdot \frac{\pi}{2}$, the geodesic distance $\theta = d_S(f, g)$ between f and g satisfies the relation $|\theta - |f - g|| < \varepsilon$ for a positive small number ε.

Figure 8.1a, b show geometric properties of the subspace method and multiclass recognition using the subspace method, respectively. Let φ_1 and φ_2 be the basis of a linear subspace for a pattern. For an input g, the similarity is computed using the length of the orthogonal projection of g to the pattern space. The subspace method allows us to achieve multiclass recognition using orthogonal projections. Setting $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ to be the orthogonal projections to subspaces $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively,

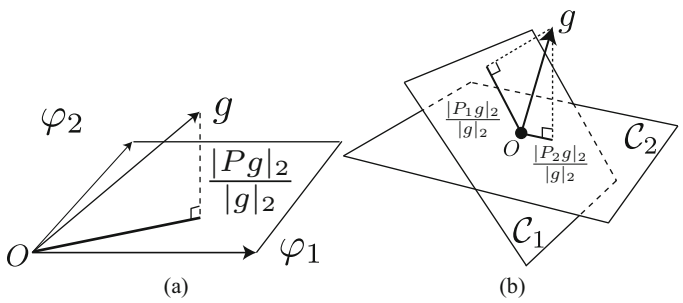


Fig. 8.1 Subspace method. (**a**) Setting φ_1 and φ_2 to be the basis of a linear subspace corresponding to a pattern, for an input g, the similarity between g and a pattern in a pattern space is measured by the length of the orthogonal projection of g to the pattern space. (**b**) Setting P_1 and P_2 to be operators for subspaces $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively, the input g is labelled as being in the first class, since the length of the orthogonal projection of g to $\mathscr{C}_1$ is greater than the length of the orthogonal projection of g to $\mathscr{C}_2$

the input g is recognised as an element in the first class, since the length of the orthogonal projection of g to $\mathscr{C}_1$ is greater than the length of the orthogonal projection of g to $\mathscr{C}_2$. The ratio $|Pg|/|g|$ is called the cumulative contribution ratio (CCR) of g to the linear subspace defined by P.

Principal Component Analysis

The variational average of data $\{f_i\}_{i=1}^n$ defined in a metric space $\mathfrak{M}$ is the minimiser of the variational problem

$$J(g) = \sum_{i=1}^{n} d(f_i, g)^2 + \lambda P(g), \tag{8.11}$$

using the metric $d(\cdot, \cdot)$ in $\mathfrak{M}$, where g in the first term of the right-hand side of the equation is the Frechet mean of $\{f_i\}_{i=1}^n$ and the second term is the regulariser for g.

For volumetric images $\{f_i\}_{i=1}^n$ defined in a finite closed region of the three-dimensional Euclidean space $\mathbb{R}^3$, the normalised average is the minimiser u of the optimisation criterion

$$J_\alpha(u) = \sum_{i=1}^{n} |f_i - \alpha u|^2, \tag{8.12}$$

for $\alpha > 0$ with the condition $|u|^2 = 1$. The normalised average is the maximiser of the variational problem

$$J_\lambda(u) = \sum_{i=1}^{n} |(f_i, u)|^2 + \lambda(1 - |u|^2). \tag{8.13}$$

Variation on Eq. (8.13) implies that u is the eigenfunction of the correlation of $\{f_i\}_{i=1}^n$ associated with the maximal eigenvalue.

Three-way array data derived from a sampled discrete function $f_{ijk} = f(\Delta i, \Delta j, \Delta k)$ of f defined in $\mathbb{R}^3$ are embedded into $\mathbb{R}^n$ as a vector $\boldsymbol{f} = (f_1, f_2, \cdots, f_n)^\top$. Let $\mathbb{D} = \{\boldsymbol{f}_i\}_{i=1}^m$ be a collection of vectors in n-dimensional Euclidean space $\mathbb{R}^n$ with the condition $\sum_{i=1}^m \boldsymbol{f}_i = 0$. Setting $(\boldsymbol{f}, \boldsymbol{g}) = \boldsymbol{f}^\top \boldsymbol{g}$ and $|\boldsymbol{f}| = \sqrt{|\boldsymbol{f}^\top \boldsymbol{f}|}$ to be the inner product and the norm induced by the inner product in $\mathbb{R}^n$, respectively, the principal component of $\mathbb{D} \subset \mathbb{R}^n$ is the minimiser of the criterion

$$J(\boldsymbol{u}) = \frac{1}{m} \sum_{i=1}^{m} |\boldsymbol{f} - (\boldsymbol{f}_i, \boldsymbol{u})\boldsymbol{u}|^2 \tag{8.14}$$

with the condition $|\boldsymbol{u}|^2 = 1$. $\boldsymbol{u}$ is the eigenvector of the correlation of $\{\boldsymbol{f}_i\}_{i=1}^n$ associated with the maximal eigenvalue. Therefore, setting

$$\boldsymbol{M} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{f}_i \boldsymbol{f}_i^\top \tag{8.15}$$

for $\sum_{i=1}^n \boldsymbol{f}_i = 0$, the eigenvectors $\{\boldsymbol{u}\}_{i=1}^n$ of

$$\boldsymbol{M}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i, \quad \lambda_i \geq 0 \tag{8.16}$$

define the principal components of $\mathbb{D}$. Using $\{\boldsymbol{u}\}_{i=1}^n$, we can define a linear subspace $\mathit{\Pi} = \mathsf{L}(\{\boldsymbol{u}_i = 1^k\})$ for $k \leq n$ as the category $\mathscr{C}$ defined by $\{\boldsymbol{f}_i\}_{i=1}^m$. Furthermore, the orthogonal projection to the category $\mathscr{C}$ is $\boldsymbol{P}_{\mathscr{C}} = \sum_{i=1}^k \boldsymbol{u}\boldsymbol{u}^\top$. We note that for practical applications the relation $k \ll n \ll m$ is expect

Since the DCT [27] is asymptotically equivalent to the matrix of the K-L transform [24] for data observed from a first-order Markov model [7, 24], the dimension reduction by PCA is performed using the DCT as

$$f_i^n = \sum_{i'=0}^{k-1} \varphi_{i'i} g_{i'}, \quad g_i = \sum_{i'=0}^{n-1} \varphi_{ii'} f_{i'} \tag{8.17}$$

for $k \leq n$, where

$$\boldsymbol{\Phi}_{(n)} = ((\epsilon \cos \frac{(2j+1)i}{2\pi n})) = ((\varphi_{ij})), \quad \epsilon = \begin{cases} 1 & \text{if } j = 0 \\ \frac{1}{\sqrt{2}} & \text{otherwise} \end{cases} \tag{8.18}$$

is the DCT-II matrix of order n. If we apply the fast cosine transform to the computation of the DCT-II matrix, the computational complexity is $\mathscr{O}(n \log n)$.

There are several extensions of PCA. Let a collection of structured data $\mathfrak{T} = \{\boldsymbol{t}_i\}_{i=1}^n$ be a subset of a metric space $\mathfrak{M}$. An outline of these data is computed as the mean by minimising the criterion

$$\overline{\boldsymbol{t}} = \arg\{\min_{t} \sum_{i=1}^{n} d(\boldsymbol{t}_i, \boldsymbol{t})^2\}. \tag{8.19}$$

We can extend the PCA algorithm as follows.

1. Compute the centroid $\bar{\boldsymbol{t}}$ of $\mathfrak{T}$.
2. Select a metric $d(\boldsymbol{t}, \boldsymbol{t}')$ in $\mathfrak{M}$.
3. Select a geodesic path P^* which minimises the criterion

$$J = \sum_{i=1}^{n} d(\boldsymbol{t}_i, P)^2 \tag{8.20}$$

with the condition $\bar{\boldsymbol{t}} \in P$ in $\mathfrak{M}$.

This data processing is called principal geodesic analysis (PGA) [23]. If we can define a manifold of combinatorial structures such as trees and graphs, PGA can be used for the classification and retrieval of structured objects. For geodesic PCA (GPCA), the curvature of spaces is extended from zero to nonzero. Shape spaces and collections of phylogenetic trees are examples of positive- and negative-curvature spaces, respectively. GPCA in a shape space is used for longitudinal analysis (follow-up analysis) of cancers in organs. GPCA for phylogenetic trees computes the mean of the trees in the data space. This extension is derived on the basis of OODA, which deals with data without embedding into vector spaces, in which the computation of PCA is achieved.

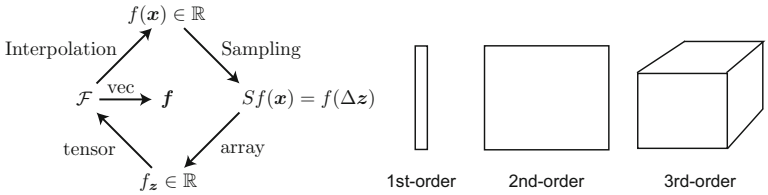
Tensor Analysis and Sampling

For the triplet of positive integers I_1, I_2 and I_3, the third-order tensor $\mathbb{R}^{I_1 \times I_2 \times I_3}$ is expressed as $\mathscr{X} = ((x_{ijk}))$. Indices i, j and k are called the 1-mode, 2-mode and 3-mode of $\mathscr{X}$, respectively. The tensor space $\mathbb{R}^{I_1 \times I_2 \times I_3}$ is interpreted as the Kronecker product of three vector spaces $\mathbb{R}^{I_1}$, $\mathbb{R}^{I_2}$ and $\mathbb{R}^{I_3}$ such that $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \mathbb{R}^{I_3}$. We set $I = \max(I_1, I_2, I_3)$.

For a square integrable function $f(\boldsymbol{x})$, which is zero outside of a finite support Ω in three-dimensional Euclidean space, the sample $Sf(\Delta \mathbf{z})$ for $\mathbf{z} \in \mathbf{Z}^3$ and $|\mathbf{z}|_\infty \leq I$ defines an $I \times I \times I$ three-way array $\mathbf{F}$. To preserve the multi-linearity of the function $f(\boldsymbol{x})$, we deal with the array $\mathbf{F}$ as a third-order tensor $\mathscr{F}$. The operation $vec\mathscr{F}$ derives a vector $\boldsymbol{f} \in \mathbb{R}^{I_{123}}$ for $I_{123} = I_2 \cdot I_2 \cdot I_3$. We can reconstruct f from $\mathscr{F}$ using an interpolation procedure. Figure 8.2a shows the relations among sampled data and multi-way data. Figure 8.2b shows a data compression procedure for multi-way data.

For the outer product of N vectors, if the tensor $\mathscr{X}$ satisfies the condition

$$\mathscr{X} = \boldsymbol{u}^{(1)} \circ \boldsymbol{u}^{(2)} \circ \boldsymbol{u}^{(3)}, = ((u_i^1 u_j^2 u_j^3)) \tag{8.21}$$



(a) Sampling procedure (b) Type of tensors

Fig. 8.2 Sampling and tensor expression of multi-way data. We can reconstruct f from $\mathscr{F}$ using an interpolation procedure. (**a**) Relations among sampled data and multi-way data. The sampled values of a multivariate function derive multi-way array data. These multi-way array data are dealt with as a higher-order tensor to preserve the multilinear properties of the data. (**b**) Data compression procedure for multi-way data by deriving a small-size tensor from the original one

for $\boldsymbol{u}^{(k)} = (u_1^k, u_2^k, \cdots, u_{I_k}^k)^\top$, where $\circ$ denotes the outer product, we call this tensor $\mathscr{X}$ a rank-one tensor. For $\mathscr{X}$, the n-mode vectors, $n = 1, 2, 3$, are defined as the I_n-dimensional vectors obtained from $\mathscr{X}$ by varying this index i_n while fixing all the other indices.

The unfolding of $\mathscr{X}$ along the n-mode vectors of $\mathscr{X}$ is defined as matrices such that

$$\mathscr{X}_{(1)} \in \mathbb{R}^{I_1 \times I_{23}}, \quad \mathscr{X}_{(2)} \in \mathbb{R}^{I_2 \times I_{13}}, \quad \mathscr{X}_{(3)} \in \mathbb{R}^{I_3 \times I_{12}} \tag{8.22}$$

for $I_{12} = I_1 \cdot I_2$, $I_{23} = I_2 \cdot I_3$ and $I_{13} = I_1 \cdot I_3$, where the column vectors of $\mathscr{X}_{(j)}$ are the j-mode vectors of $\mathscr{X}$ for $i = 1, 2, 3$. We express the j-mode unfolding of $\mathscr{X}_i$ as $\mathscr{X}_{i,(j)}$. Figure 8.3a, b show unfolding procedures of second- and third-order tensors, respectively.

For matrices $\boldsymbol{U} = ((u_{ii'})) \in \mathbb{R}^{I_1 \times I_1}$, $\boldsymbol{V} = ((v_{jj'})) \in \mathbb{R}^{I_2 \times I_2}$ and $\boldsymbol{W} = ((w_{kk'})) \in \mathbb{R}^{I_3 \times I_3}$, the n-mode products for $n = 1, 2, 3$ of a tensor $\mathscr{X}$ are the tensors with entries

$$x_{[1]ijk} = \sum_{i'=1}^{I_1} x_{i'jk} u_{i'i}, \quad x_{[2]ijk} = \sum_{j'=1}^{I_2} x_{ij'k} v_{j'j}, \quad x_{[3]ijk} = \sum_{k'=1}^{I_3} x_{ijk'} w_{k'k}, \tag{8.23}$$

where $(\mathscr{X})_{ijk} = x_{ijk}$ is the ijkth element of tensor $\mathscr{X}$. The inner product of two tensors $\mathscr{X}$ and $\mathscr{Y}$ in $\mathbb{R}^{I_1 \times I_2 \times I_3}$ is

$$\langle \mathscr{X}, \mathscr{Y} \rangle = \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} x_{ijk} y_{ijk}. \tag{8.24}$$

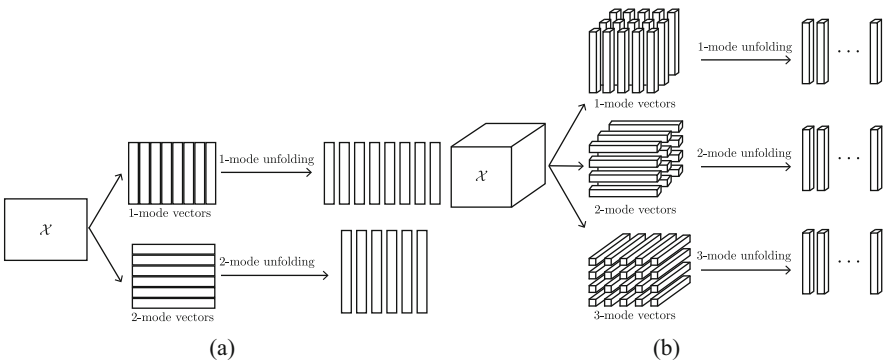


Fig. 8.3 Unfolding of tensors. (**a**) Unfolding of a second-order tensor. For a tensor in $\mathbb{R}^{6\times8}$, unfolding of 1- and 2-modes yields eight 1-mode vectors and six 2-mode vectors, respectively. (**b**) Unfolding of a third-order tensor. For a tensor in $\mathbb{R}^{4\times5\times3}$, unfolding for 1-, 2- and 3-modes yields fifteen 1-mode vectors, twelve 2-mode vectors and twenty 3-mode vectors, respectively

Using this inner product, we have the Frobenius norm of a tensor $\mathscr{X}$ as $|\mathscr{X}|_F = \sqrt{\langle \mathscr{X}, \mathscr{X} \rangle}$. The Frobenius norm $|\mathscr{X}|_F$ of tensor $\mathscr{X}$ satisfies the relation $|\mathscr{X}|_F = |\boldsymbol{f}|_2$, where $|\boldsymbol{f}|_2$ is the Euclidean norm of vector $\boldsymbol{f} = vec\mathscr{F}$ of the vectorisation of tensor $\mathscr{F}$.

To project a tensor $\mathscr{X}$ in $\mathbb{R}^{I_1 \times I_2 \times I_3}$ to the tensor $\mathscr{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1 \times P_2 \times P_3}$, where $P_n \leq I_n$, three projection matrices $\{\boldsymbol{P}^{(i)}\}_{i=1}^3$ for $\boldsymbol{P}^{(i)} \in \mathbb{R}^{I_n \times P_n}$ are required for $i = 1, 2, 3$.

$$\mathscr{Y} = \mathscr{X} \times_1 \boldsymbol{P}^{(1)\top} \times_2 \boldsymbol{P}^{(2)\top} \times_3 \boldsymbol{P}^{(3)\top}. \tag{8.25}$$

This projection is established in three steps, where in each step, each i-mode vector is projected to a P_n-dimensional space by $\boldsymbol{P}^{(i)}$ for $i = 1, 2, 3$.

Tensor Principal Component Analysis

Setting the data matrix $\boldsymbol{X}$ to be $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 & \cdots & \boldsymbol{f}_m \end{pmatrix}$ for data vectors $\{\boldsymbol{f}_i\}_{i=1}^m$ in $\mathbb{R}^N$, whose mean is zero, the K-L transform is established by computing $\widehat{\boldsymbol{f}}_i = \boldsymbol{U}\boldsymbol{f}_i$ for $\boldsymbol{U}$ which minimises

$$J_1 = |\boldsymbol{U}\boldsymbol{X}|_F^2 \tag{8.26}$$

with the condition $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_N$. The orthogonal matrix $\boldsymbol{U}$ is the minimiser of

$$J_{11} = |\boldsymbol{U}\boldsymbol{X}|_F^2 + (\boldsymbol{U}^\top \boldsymbol{U} - \boldsymbol{I})\boldsymbol{\Lambda}, \tag{8.27}$$

where $\boldsymbol{\Lambda} = Diag(\lambda_1, \lambda_2, \cdots, \lambda_N)$ for $\lambda_1 \geq \lambda_2 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$. The minimiser of Eq. (8.27) is the solution of the eigenmatrix problem

$$\boldsymbol{M}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Lambda}, \ \ \boldsymbol{M} = \boldsymbol{X}\boldsymbol{X}^\top. \tag{8.28}$$

The row vectors of $\boldsymbol{U}$ are the principal components.

The compression of $\boldsymbol{f}_i$ to a low-dimensional linear subspace is achieved by computing the transform $\boldsymbol{P}_n\boldsymbol{U}\boldsymbol{f}$, where $\boldsymbol{P}_n$ is the orthogonal projection such that

$$\boldsymbol{P}_n = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{O} \\ \boldsymbol{O}^\top & \boldsymbol{O} \end{pmatrix} \tag{8.29}$$

for $n < N$.

Using three orthogonal matrices $\boldsymbol{U}^{(i)}$ for $i = 1, 2, 3$, we have the tensor orthogonal decomposition for a third-order tensor as

$$\mathscr{Y} = \mathscr{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top}. \tag{8.30}$$

For a collection $\{\mathscr{X}_k\}_{k=1}^m$ of third-order tensors, the orthogonal-projection-based dimension reduction procedure is achieved by maximising the criterion

$$J_3 = E_k(|\mathscr{X}_k \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top}|_F^2) \tag{8.31}$$

with respect to the conditions $\boldsymbol{U}^{(i)^\top}\boldsymbol{U}^{(i)} = \boldsymbol{I}$ for $i = 1, 2, 3$. The Euler-Lagrange equation of this conditional optimisation problem is

$$\begin{aligned} J_{33}(\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}^{(3)}) = {} & E_k(|\mathscr{X}_k \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top}|_F^2) \\ & + \sum_{i=1}^{3} |(\boldsymbol{I} - \boldsymbol{U}^{(i)^\top}\boldsymbol{U}^{(i)})\boldsymbol{\Lambda}^{(i)}|_F^2. \end{aligned} \tag{8.32}$$

This minimisation problem is solved by the following iteration procedure.

Algorithm 8.1:

1: $\boldsymbol{U}_{(0)}^{(i)} := \boldsymbol{Q}^{(i)}$ such that $\boldsymbol{Q}^{(i)\top}\boldsymbol{Q}^{(i)} = \boldsymbol{I}$ and $\alpha = 0$.
2: $\boldsymbol{U}_{(\alpha+1)}^{(1)} = arg\min J_{33}(\boldsymbol{U}^{(1)}, \boldsymbol{U}_{(\alpha)}^{(2)}, \boldsymbol{U}_{(\alpha)}^{(3)})$.
3: $\boldsymbol{U}_{(\alpha+1)}^{(2)} = arg\min J_{33}(\boldsymbol{U}_{(\alpha+1)}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}_{(\alpha)}^{(3)})$.
4: $\boldsymbol{U}_{(\alpha+1)}^{(3)} = arg\min J_{33}(\boldsymbol{U}_{(\alpha+1)}^{(1)}, \boldsymbol{U}_{(\alpha+1)}^{(2)}, \boldsymbol{U}^{(3)})$.
5: if $|\boldsymbol{U}_{(\alpha+1)}^{(i)} - \boldsymbol{U}_{(\alpha)}^{(i)}|_F \leq \varepsilon$, then stop, else $\alpha := \alpha + 1$ and go to step 2.

For the practical computation of tensor PCA, we call this iteration-based method the higher-order singular value decomposition (HOSVD).

For

$$\begin{aligned} J_{33}(\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}^{(3)}) = {} & E_k(|\mathscr{X}_k \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)^\top}|_F^2) \\ & + \sum_{i=1}^{3} |(\boldsymbol{I} - \boldsymbol{U}^{(i)^\top}\boldsymbol{U}^{(i)})\boldsymbol{\Lambda}^{(i)}|_F^2 \end{aligned} \tag{8.33}$$

setting $\boldsymbol{U}_1^{(i)} := \boldsymbol{I}$, the system of minimisation problems

$$\begin{aligned} \boldsymbol{U}^{(1)} &= arg\min f(\boldsymbol{U}^{(1)}, \boldsymbol{I}, \boldsymbol{I}) \\ \boldsymbol{U}^{(2)} &= arg\min f(\boldsymbol{I}, \boldsymbol{U}^{(2)}, \boldsymbol{I}) \\ \boldsymbol{U}^{(3)} &= arg\min f(\boldsymbol{I}, \boldsymbol{I}, \boldsymbol{U}^{(3)}) \end{aligned} \tag{8.34}$$

is derived. This system of minimisation problems derives the following system of

eigenmatrix problems:

$$\begin{aligned}\nabla_{\boldsymbol{U}^{(1)}} J_{33}(\boldsymbol{U}^{(1)}, \boldsymbol{I}, \boldsymbol{I}) &= 0\\ \nabla_{\boldsymbol{U}^{(2)}} J_{33}(\boldsymbol{I}, \boldsymbol{U}^{(2)}, \boldsymbol{I}) &= 0\\ \nabla_{\boldsymbol{U}^{(3)}} J_{33}(\boldsymbol{I}, \boldsymbol{I}, \boldsymbol{U}^{(3)}) &= 0.\end{aligned} \tag{8.35}$$

We call this method matrix PCA. In matrix PCA, if we set the number of bases to the size of the original tensors in Algorithm 8.1, we call the method full projection (FP). If we set the number of bases to fewer than the size of the original tensors in Algorithm 8.1, we call the method full-projection truncation (FPT).

From Eq. (8.35), as an extension of the two-dimensional problem, we define the system of optimisation problems [14]

$$J_j = E(|\boldsymbol{U}^{(j)\top} \mathscr{X}_{i,(j)} \boldsymbol{U}^{(j)}|_F^2) + (\boldsymbol{U}^{(j)\top} \boldsymbol{U}^{(j)} - \boldsymbol{I}_j)\boldsymbol{\Lambda}^{(j)} \tag{8.36}$$

for $i = 1, 2, 3$, as a relaxation of the iteration procedure, where $\mathscr{X}_{i,(j)}$ is the ith column vector of the unfolding matrix $\mathscr{X}_{(j)}$. These optimisation problems derive the system of eigenmatrix problems

$$\boldsymbol{M}^{(j)} \boldsymbol{U}^{(j)} = \boldsymbol{U}^{(j)} \boldsymbol{\Lambda}^{(j)}, \quad \boldsymbol{M}^{(j)} = \frac{1}{N} \sum_{i=1}^{N} \mathscr{X}_{i,(j)} \mathscr{X}_{i,(j)}^{\top} \tag{8.37}$$

for $j = 1, 2, 3$.

Setting $\boldsymbol{P}^{(j)}$ to be an orthogonal projection in the linear space $\mathscr{L}(\{\boldsymbol{u}_i^{(j)}\}_{i=1}^{I_j})$ spanned by the column vectors of $\boldsymbol{U}^{(j)}$, data reduction is computed by

$$\mathscr{Y} = \mathscr{X} \times_1 (\boldsymbol{P}^{(1)} \boldsymbol{U}^{(1)})^{\top} \times_2 (\boldsymbol{P}^{(2)} \boldsymbol{U}^{(2)})^{\top} \times_3 (\boldsymbol{P}^{(3)} \boldsymbol{U}^{(3)})^{\top}. \tag{8.38}$$

This expression is equivalent to the vector form

$$vec\mathscr{Y} = (\boldsymbol{P}^{(3)} \otimes \boldsymbol{P}^{(2)} \otimes \boldsymbol{P}^{(1)})(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)} \otimes \boldsymbol{U}^{(1)}) vec\mathscr{X}. \tag{8.39}$$

The eigenvalues of the eigenmatrices of Tucker-3 orthogonal decomposition satisfy the following theorem.

Theorem 8.1 *The eigenvalues of* $\boldsymbol{U} = \boldsymbol{U}^{(1)} \otimes \boldsymbol{U}^{(2)} \otimes \boldsymbol{U}^{(3)}$ *define a semi-order.*

Proof For the eigenvalues $\lambda_i^{(1)}, \lambda_j^{(2)}, \lambda_k^{(3)}$ of the 1-, 2- and 3-modes of tensors, the inequalities

$$\begin{aligned}\lambda_i \lambda_j \lambda_k &\geq \lambda_{i+1} \lambda_j \lambda_k,\\ \lambda_i \lambda_j \lambda_k &\geq \lambda_i \lambda_{j+1} \lambda_k,\\ \lambda_i \lambda_j \lambda_k &\geq \lambda_i \lambda_j \lambda_{k+1},\end{aligned} \tag{8.40}$$

define semi-orders among the eigenvalues as

$$\lambda_i^{(1)}\lambda_j^{(2)}\lambda_k^{(3)} \succeq \left\langle \lambda_i^{(1)}\lambda_j^{(2)}\lambda_{k+1}^{(3)}, \lambda_i^{(1)}\lambda_{j+1}^{(2)}\lambda_k^{(3)}, \lambda_{i+1}^{(1)}\lambda_j^{(2)}\lambda_k^{(3)} \right\rangle \tag{8.41}$$

is satisfied. □

Regarding the selection of the dimension of the tensor subspace, Theorem 8.1 implies the following theorem.

Theorem 8.2 *The dimension of the subspace of the tensor space for data compression is* $\frac{1}{6}n(n+1)(n+2)$ *if we select n principal components in each mode of three-way array data.*

Proof For a positive integer n, the number s_n of eigenvalues $\lambda_i^{(1)}\lambda_j^{(2)}\lambda_k^{(3)}$ is

$$s_n = \sum_{i+j+k=0, i,j,k\geq 0}^{n-1} (i+j+k) = \sum_{l=1}^{n}\sum_{m=1}^{l} m = \frac{1}{6}n(n+1)(n+2). \qquad \square$$

If $n = 1, 2, 3, 4$, we have $N = 1, 4, 10, 20$, respectively, for tensors $\mathscr{X} = ((x_{ijk}))$ in $\mathbb{R}^{I\times I\times I}$.

Setting $\{\boldsymbol{P}^{(i)}\}_{i=1}^3$ to be orthogonal projection matrices, the orthogonal projection of a third-order tensor $\mathscr{X}$ to the linear subspace Π_{123} by $\{\boldsymbol{P}^{(i)}\}_{i=1}^3$ is computed as

$$\mathscr{Y} = \mathscr{X} \times_1 \boldsymbol{P}^{(1)} \times_2 \boldsymbol{P}^{(2)} \times_3 \boldsymbol{P}^{(3)}. \tag{8.42}$$

Since $|\mathscr{Y}|_F$ is the length of the part of tensor $\mathscr{X}$ on the linear subspace Π_{123}, the ratio $0 \leq |\mathscr{Y}|_F/|\mathscr{X}|_F \leq 1$ is the CCR of $\mathscr{X}$ to Π_{123}. The dimension of Π_{123} is computed by Theorems 8.1 and 8.2.

Since the DCT [27] is asymptotically equivalent to the matrix of the K-L transform [24] for data observed from a first-order Markov model [7, 24], the dimension reduction by PCA is performed using the DCT as

$$f_{ijk}^n = \sum_{i'j'k'=0}^{k-1} g_{i'j'k'}\varphi_{i'i}\varphi_{j'j}\varphi_{k'k}, \quad g_{ijk} = \sum_{i'j'k'=0}^{n-1} f_{i'j'k'}\varphi_{ii'}\varphi_{jj'}\varphi_{kk'} \tag{8.43}$$

for $k \leq n$, where $\boldsymbol{\Phi}_{(n)}$ is the DCT-II matrix of order n. If we apply the fast cosine transform to the computation of the 3D-DCT-II matrix, the computational complexity is $O(3n\log n)$.

In the vector and tensor forms, the transforms are expressed as

$$vec\mathscr{F}^n = (\boldsymbol{\Phi}_{(n)} \otimes \boldsymbol{\Phi}_{(n)} \otimes \boldsymbol{\Phi}_{(n)})^\top (\boldsymbol{P}_k \otimes \boldsymbol{P}_k \otimes \boldsymbol{P}_k)(\boldsymbol{\Phi}_{(n)} \otimes \boldsymbol{\Phi}_{(n)} \otimes \boldsymbol{\Phi}_{(n)}) vec\mathscr{F} \tag{8.44}$$

$$\mathscr{F}^n = \mathscr{F} \times_1 (\boldsymbol{\Phi}_{(n)}^\top \boldsymbol{P}_k \boldsymbol{\Phi}_{(n)}) \times_2 (\boldsymbol{\Phi}_{(n)}^\top \boldsymbol{P}_k \boldsymbol{\Phi}_{(n)}) \times_3 (\boldsymbol{\Phi}_{(n)}^\top \boldsymbol{P}_k \boldsymbol{\Phi}_{(n)}). \tag{8.45}$$

Since

$$vec(\boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{w}) = \boldsymbol{u} \otimes \boldsymbol{v} \otimes \boldsymbol{w}, \tag{8.46}$$

the outer product of vectors redescribes the DCT-based transform as

$$\mathscr{F} = \sum_{i,j,k=0}^{n-1} a_{ijk} \boldsymbol{\varphi}_i \circ \boldsymbol{\varphi}_j \circ \boldsymbol{\varphi}_k, \;\; a_{ijk} = \langle \mathscr{F}, (\boldsymbol{\varphi}_i \circ \boldsymbol{\varphi}_j \circ \boldsymbol{\varphi}_k) \rangle, \tag{8.47}$$

where

$$\boldsymbol{\Phi}_{(n)} = \left(\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_{n-1}\right). \tag{8.48}$$

The DCT matrix $\boldsymbol{\Phi}_{(n)}$ is the eigenmatrix of the discrete Laplacian with the Neumann boundary condition. We can define the order of the column vectors of DCT matrix using the order of the eigenvalue $\{\lambda_i\}_{i=0}^{n-1}$ of the discrete Laplacian. Since $\lambda_i \lambda_j \lambda_k$ derives the same semi-order with relation of Eq. (8.41), we define the order for the outer product of the column vectors $\{\boldsymbol{\varphi}_{i=0}^{n-1}\}$

$$\begin{aligned} \boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_j \otimes \boldsymbol{\varphi}_k &\succ \boldsymbol{\varphi}_{i+1} \otimes \boldsymbol{\varphi}_j \otimes \boldsymbol{\varphi}_k, \\ \boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_j \otimes \boldsymbol{\varphi}_k &\succ \boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_{j+1} \otimes \boldsymbol{\varphi}_k, \\ \boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_j \otimes \boldsymbol{\varphi}_k &\succ \boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_j \otimes \boldsymbol{\varphi}_{k+1}. \end{aligned} \tag{8.49}$$

This order is used for the definition of the dimension of subspace for the relaxed PCA with DCT. On this order, the kth elements lies on the surface of the oct-sphere of the radius $k-1$ with the l_1-distance. Therefore, this order defines the low-pass filter of which path window is the oct-diamond in discrete space.

Classification of Three-Way Array Data

Tensor Subspace Method

As an extension of the subspace method [9, 29] for three-way data, we introduce a linear tensor subspace method (TSM) for third-order tensors. This method is a three-dimensional version of the two-dimensional TSM [13].

For a third-order tensor $\mathscr{X}$, we set $\boldsymbol{U}^{(j)}$ for $j = 1, 2, 3$, to be projection matrices of the tensor-to-tensor projection of $\mathscr{X}$ to $\mathscr{Y}$. For a collection of normalised tensors $\{\mathscr{X}_i\}_{i=1}^{M}$, such that $\mathscr{X}_i \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, $\|\mathscr{X}_i\|_{\mathrm{F}} = 1$ and $\mathrm{E}(\mathscr{X}_i) = 0$, the solutions of

$$\{U^{(j)}\}_{j=1}^{3} = \arg\max \mathrm{E}\left(\|\mathscr{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top}\|_{\mathrm{F}} / \|\mathscr{X}_i\|_{\mathrm{F}}\right) \tag{8.50}$$

with respect to $\boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)} = \boldsymbol{I}$ for $j = 1, 2, 3$ define a multilinear subspace that approximates $\{\mathscr{X}_i\}_{i=1}^{M}$. Therefore, using projection matrices $\{\boldsymbol{U}_k^{(j)}\}_{j=1}^{3}$ obtained as

the solutions of Eq. (8.50) for the kth category $\mathfrak{C}_k$, if a query tensor $\mathscr{G}$ satisfies the condition

$$\arg\left(\max_l \|\mathscr{G} \times_1 \boldsymbol{U}_l^{(1)\top} \times_2 \boldsymbol{U}_l^{(2)\top} \times_3 \boldsymbol{U}_l^{(3)\top}\|_{\mathrm{F}} / \|\mathscr{G}\|_{\mathrm{F}}\right) = \{\boldsymbol{U}_k^{(j)}\}_{j=1}^3, \tag{8.51}$$

we conclude that $\mathscr{G} \in \mathfrak{C}_k$, $k, l = 1, 2, \ldots, N_{\mathscr{C}}$, where $\mathfrak{C}_k$ and $N_{\mathfrak{C}}$ are the tensor subspace of the kth category and the number of categories, respectively.

Since a pattern represented by a tensor includes perturbation, we define the kth category as

$$\mathfrak{C}_k(\delta) = \{\mathscr{X} \mid \|\mathscr{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top} - \mathscr{X}\|_{\mathrm{F}} \ll \delta\}, \tag{8.52}$$

where the positive constant δ is the bound for the perturbation to the pattern. Therefore, by defining similarity and dissimilarity between a tensor subspace and a query, we can construct tensor-subspace-based classifiers that are robust and stable against small perturbations to patterns.

Mutual Tensor Subspace Method

Setting $\mathfrak{C}_{\mathrm{q}} = \{\mathscr{G}_{i'}\}_{i'=1}^{M'}$ to be a collection of query tensors, the orthogonal matrices

$$\{\boldsymbol{V}^{(j)}\}_{j=1}^3 = \arg\max \mathrm{E}\left(\|\mathscr{G} \times_1 \boldsymbol{V}^{(1)\top} \times_2 \boldsymbol{V}^{(2)\top} \times_3 \boldsymbol{V}^{(3)\top}\|_{\mathrm{F}} / \|\mathscr{G}_i\|_{\mathrm{F}}\right) \tag{8.53}$$

are orthogonal projections in each mode for $\mathfrak{C}_{\mathrm{q}}$. We have the projected tensors

$$\mathscr{A}_{i'} = \mathscr{G}_{i'} \times_1 \boldsymbol{U}_k^{(1)\top} \times_2 \boldsymbol{U}_k^{(2)\top} \times_3 \boldsymbol{U}_k^{(3)\top} \tag{8.54}$$

in the category subspace $\mathfrak{C}_k$ and

$$\mathscr{B}_{i'} = \mathscr{G}_{i'} \times_1 \boldsymbol{V}^{(1)\top} \times_2 \boldsymbol{V}^{(2)\top} \times_3 \boldsymbol{V}^{(3)\top} \tag{8.55}$$

in the query subspace $\mathfrak{C}_{\mathrm{q}}$. These tensors define the dissimilarity between $\mathscr{C}_k$ and $\mathscr{C}_{\mathrm{q}}$ as

$$d(\mathfrak{C}_l, \mathfrak{C}_{\mathrm{q}}) = \mathrm{E}\left(\|\overline{\mathscr{A}_{i'}} - \overline{\mathscr{B}_{i'}}\|_{\mathrm{F}}^2\right) \tag{8.56}$$

for

$$\overline{\mathscr{A}_{i'}} = \mathscr{A}_{i'} \times_1 (\boldsymbol{P}\boldsymbol{U}_k^{(1)})^\top \times_2 (\boldsymbol{P}\boldsymbol{U}_k^{(2)})^\top \times_3 (\boldsymbol{P}\boldsymbol{U}_k^{(3)})^\top \tag{8.57}$$

$$\overline{\mathscr{B}_{i'}} = \mathscr{B}_{i'} \times_1 (\boldsymbol{P}\boldsymbol{V}^{(1)})^\top \times_2 (\boldsymbol{P}\boldsymbol{V}^{(2)})^\top \times_3 (\boldsymbol{P}\boldsymbol{V}^{(3)})^\top, \tag{8.58}$$

with the conditions $|\mathscr{A}_{i'}|_F = 1$ and $|\mathscr{B}_{i'}| = 1$, where the projection matrix $\boldsymbol{P}$ selects bases for each mode of tensors. Therefore, if the queries $\{\mathfrak{G}_{i'}\}_{l=1}^{M'}$ satisfy the condition

$$\arg\left(\min_{l} d(\mathfrak{C}_l, \mathfrak{C}_{\mathrm{q}})\right) = \mathscr{C}_k, \tag{8.59}$$

we conclude that $\{\mathscr{G}_{i'}\}_{i'=1}^{M'} \in \mathfrak{C}_k(\delta)$ for $k, l = 1, 2, \cdots, N_{\mathrm{C}}$, where N_{C} is the total number of categories in the pattern space. We call the classification and recognition of query subspaces using Eq. (8.59) the MTSM as the extension of the mutual subspace method for vector data to array data.

Numerical Examples

Reconstruction Errors of Volumetric Data

The performances of tensor PCA, the three-dimensional DCT (3D-DCT) and the pyramid transform (PT) are compared for the approximation of volumetric data. For tensor-based expressions of the PT, see the Appendix. The volumetric data of human livers in the computational anatomy (CA) dataset[1] and of human left ventricles in the cardiac MRI dataset [1] are used for comparisons. Table 8.1 summarises the numbers and sizes of the volumetric data. Figures 8.4 and 8.5 illustrate the original and approximated volumetric images of a human liver and a left ventricle, respectively. Figure 8.6 summarises the reconstruction errors of the three methods in terms of the compression ratio.

Figures 8.4 and 8.5 illustrate that, in terms of appearance, the 3D-DCT efficiently approximates the K-L transform as a relaxation of tensor PCA for three-way array data derived from volumetric images. The PT for volumetric grey-valued images is an acceptable approximation of the K-L transform for a low compression ratio. Since the PT is a convolution operation, the time complexity of the PT is $\mathscr{O}(n \log_2 n)$. However, for a high compression ratio, the PT loses details of the interior texture, although the PT preserves the appearance of outline shapes of the volumetric images.

Figure 8.7 shows the dependences of the reconstruction error and the CCR on the numbers of dimensions of the linear subspace for the reconstruction of volumetric images by using the 3D-DCT as a relaxation of tensor PCA. In Fig. 8.7a, we have 85, 93 and 71 nonzero eigenvalues for modes 1, 2 and 3, respectively. In Fig. 8.7b, we have 77, 70 and 63 nonzero eigenvalues for modes 1, 2 and 3, respectively. The

[1]The project "Computational Anatomy for Computer-aided Diagnosis and Therapy: Frontiers of Medical Image Sciences" funded by Grant-in-Aid for Scientific Research on Innovative Areas, MEXT, Japan. http://www.comp-anatomy.org/wiki/index.php?Computational.

Table 8.1 Size and number of volumetric data

	♯data	Data size [voxel]	Reduced data size [voxel]
CA dataset (see footnote 1)	32	$89 \times 97 \times 76$	$32 \times 32 \times 32$
Cardiac MRI dataset [1]	340	$81 \times 81 \times 63$	$16 \times 16 \times 16$

♯data represents the number of volumetric data. The data size is the original size of the volumetric data. The reduced data size is the size of the volumetric data after tensor-representation-based dimension reduction

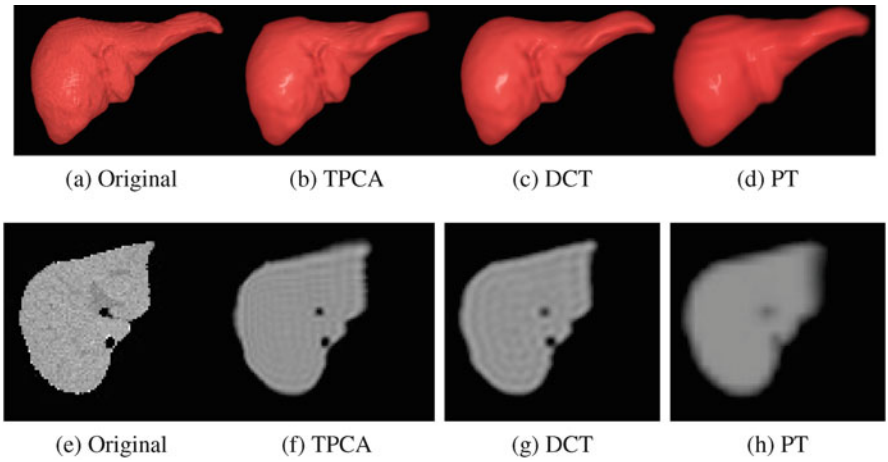


Fig. 8.4 Original and reconstructed volumetric image of a human liver. The left column illustrates the original volumetric data. The other three columns illustrate volumetric images reconstructed from the data compressed by tensor PCA, the 3D-DCT and the PT. (**a**)–(**d**) Rendered volumetric images. (**e**)–(**h**) 30th axial slice of the volumetric images. The size of the reduced volumetric data is $32 \times 32 \times 32$. The compression ratio is 0.05, that is, the size is 5.0% of the original size of $89 \times 97 \times 76$

reconstruction error decreases and the CCR increases as the number of dimensions of the linear subspace increases.

Figure 8.7a, b show that the reconstruction errors and the CCR have similar mathematical properties for data compression ratios of $1/4$ and $1/2$ if the 3D-DC Tand PT are accepted as relaxations of tensor PCA Therefore, the 3D-DCT is an acceptable relaxation method for the tensor K-L transform defined by tensor PCA.

The time complexity of PCA for vector is $\mathcal{O}(n^3)$ since the main procedure of PCA is achieved by solving eigenvalue problem. TCPA requires $\mathcal{O}(k \times n^3)$ computation times where k is the iteration times to guaranty numerically convergence of the iteration process. The time complexity of the computation of 3D-DCT is $\mathcal{O}(n \log n)$. Figure 8.8 shows the computational time of dimension reduction for tensors by using the HOSVD, the FP, FPT and 3D-DCT. Curve profiles in Fig. 8.8 support theoretical analysis of computation times.

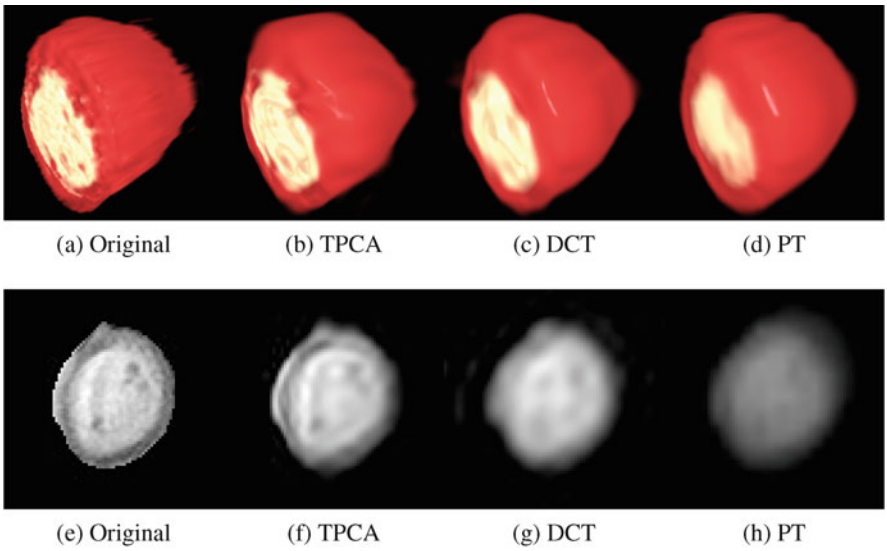


Fig. 8.5 Original and reconstructed volumetric images of a human left ventricle. The left column illustrates the original volumetric images. The other three columns illustrate the volumetric images reconstructed from the data compressed by tensor PCA, the 3D-DCT and the PT. (**a**)–(**d**) Rendered volumetric images. (**e**)–(**h**) First axial slice of the volumetric images. The size of the reduced volumetric data is $16 \times 16 \times 16$. The compression ratio is 0.01, that is, the size is 1.0 % of the original size of $81 \times 81 \times 63$

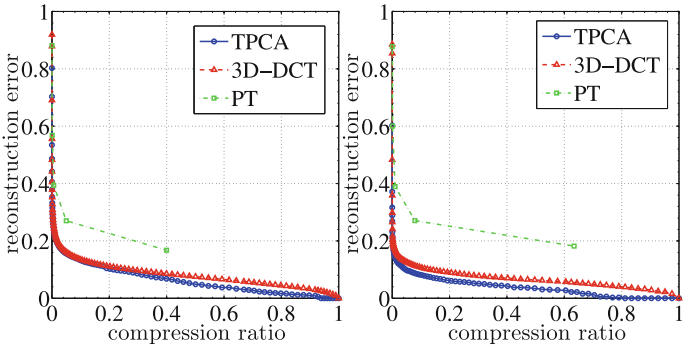


Fig. 8.6 Reconstruction error vs compression ratio. The reconstruction error is given as the relative error $\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_{\mathrm{F}}/\|\boldsymbol{X}\|_{\mathrm{F}}$, where $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$ are the original and dimension-reduced volume data, respectively. The compression ratio is given as d/n, where d and n are the reduced size and the original size, respectively. In (**a**) and (**b**) the original sizes are $86 \times 97 \times 76$ and $81 \times 81 \times 63$, respectively. The reduced size is given by $i \times j \times k$ for the original size $I \times J \times K$, where $i = 1, 2, \ldots I$, $j = 1, 2, \ldots J$ and $k = 1, 2, \ldots K$ for tensor PCA and the 3D-DCT. For the PT, the reduced sizes are $l \times l \times l$ for $l = 4, 8, 16, 32, 64$

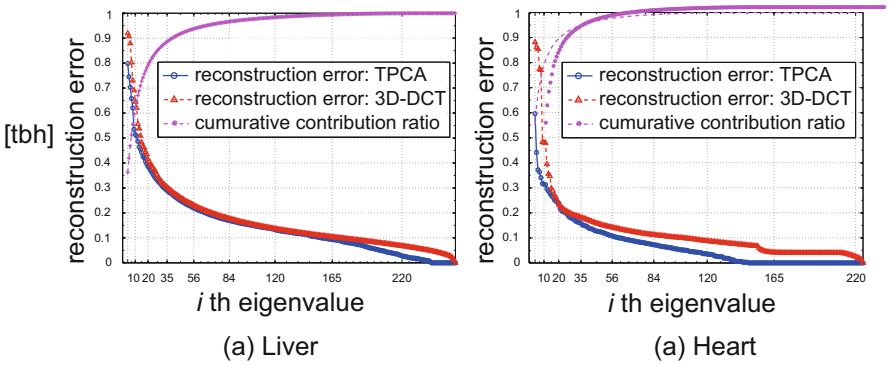


Fig. 8.7 Reconstruction errors and the CCR by using the 3D-DCT and PT as a relaxation of tensor PCA. The reconstruction error is given as the relative error $\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_{\mathrm{F}}/\|\boldsymbol{X}\|_{\mathrm{F}}$, where $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$ are the original and dimension-reduced volumetric images, respectively. From the 1st to nth frequency, the CCR is computed by $\sum_{l=1}^{n} \lambda_l / \sum_{l=1}^{N} \lambda_l$, where λ_l are the eigenvalues of the three modes in descending order. In (**a**) and (**b**) the original sizes are $76 \times 86 \times 97$ and $81 \times 81 \times 63$, respectively. The reduced sizes are given by $i \times j \times k$ for the original size $I \times J \times K$ and $i = 1, 2, \ldots I$, $j = 1, 2, \ldots J$ and $k = 1, 2, \ldots K$

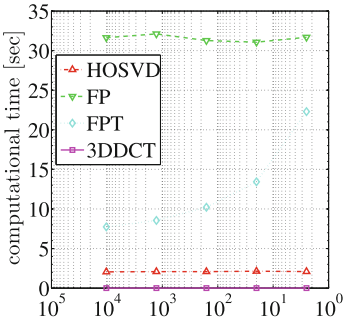


Fig. 8.8 Computational time of dimension reduction for tensors of the order three. The computational time of construction of projection matrices for 306 sequences of silhouette images and 35 voxel images of livers, respectively. We compare the HOSVD, FP, FPT and 3DDCT. The vertical and horizontal axes represent the computational time and compression ratio, respectively [14]

Average Computation of Volumetric Organ Data

For the grey-matter parts of 20 volumetric images in BrainWeb [2], we apply tensor PCA and the 3D-DCT to reduce image data sizes. The $271 \times 181 \times 181$-voxel volumetric images are reduced to $64 \times 64 \times 64$-voxel images. Table 8.2 shows the size of the original data. In Fig. 8.9, a rendered volumetric image and a slice are illustrated. For 56, 84, 120 and 165 major components, dimension-reduced volumetric images are illustrated in Figs. 8.10, 8.11, and 8.12, which show outlines

Table 8.2 Sizes and numbers of volumetric data of brains [2]

	♯data	Data size [voxel]	Reduced data size [voxel]
Volumetric data of brains	20	$217 \times 181 \times 181$	$64 \times 64 \times 64$

♯data represents the numbers of livers and brains. The data size is the original size of the volumetric data. The reduced data size is the size of the volume data after tensor-representation-based dimension reduction

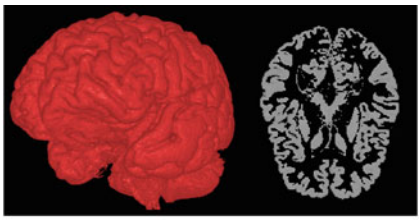


Fig. 8.9 Original volume data of brain. (**a**) Rendered volumetric image. (**b**) 80th slice of the volumetric image

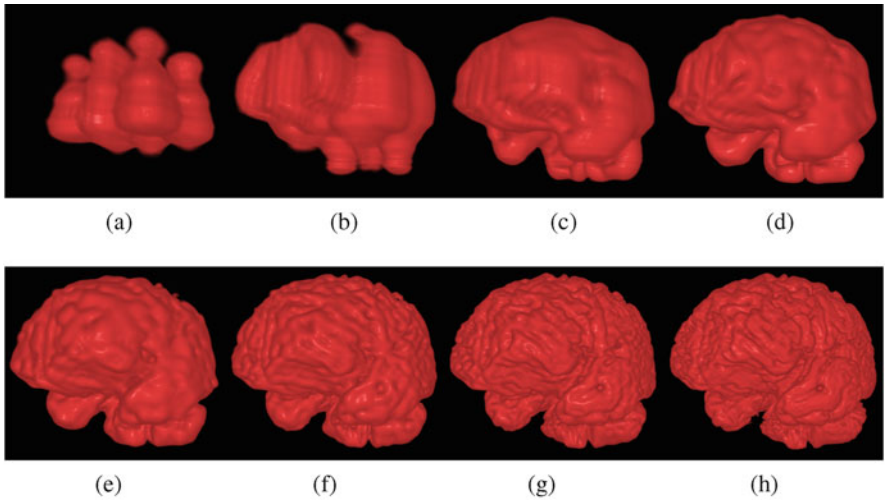


Fig. 8.10 Rendered volume images compressed by the FP of tensor principal components. The tensor principal components are computed from 20 volume data. (**a**) 4 majors, (**b**) 10 majors, (**c**) 20 majors, (**d**) 35 majors, (**e**) 56 majors, (**f**) 84 majors, (**g**) 120 majors, (**h**) 165 majors

obtained using the FP, FPT and 3D-DCT, respectively. Figures 8.13, 8.14, and 8.15 show slice images corresponding to Figs. 8.10, 8.11, and 8.12, respectively.

Figure 8.16 shows the reconstruction error for the reconstructed volume data. For the dimension reduction of volume data, tensor PCA and 3D-DCT are used. The size of the dimension-reduced data is $64 \times 64 \times 64$ voxels. The curves in Fig. 8.16 imply

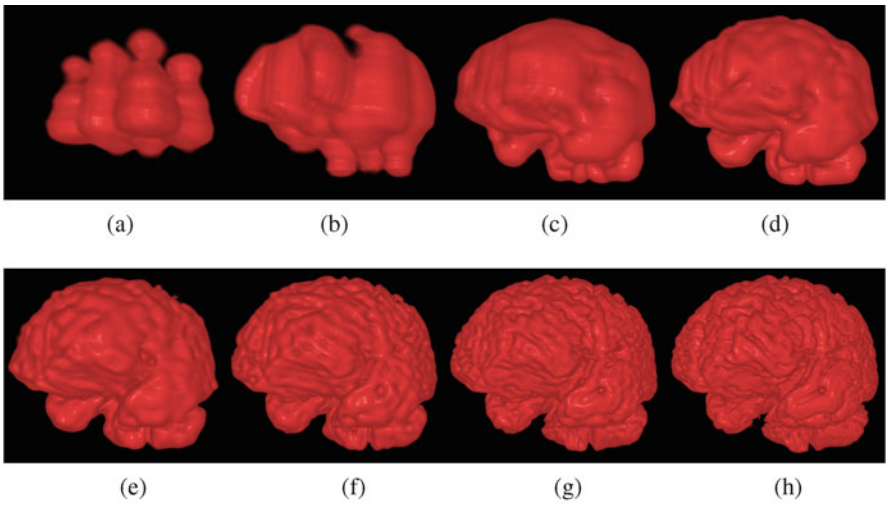


Fig. 8.11 Rendered volumetric images compressed by using several principal components. For reduction, the FPT is used. (**a**) 4 majors, (**b**) 10 majors, (**c**) 20 majors, (**d**) 35 majors, (**e**) 56 majors, (**f**) 84 majors, (**g**) 120 majors, (**h**) 165 majors

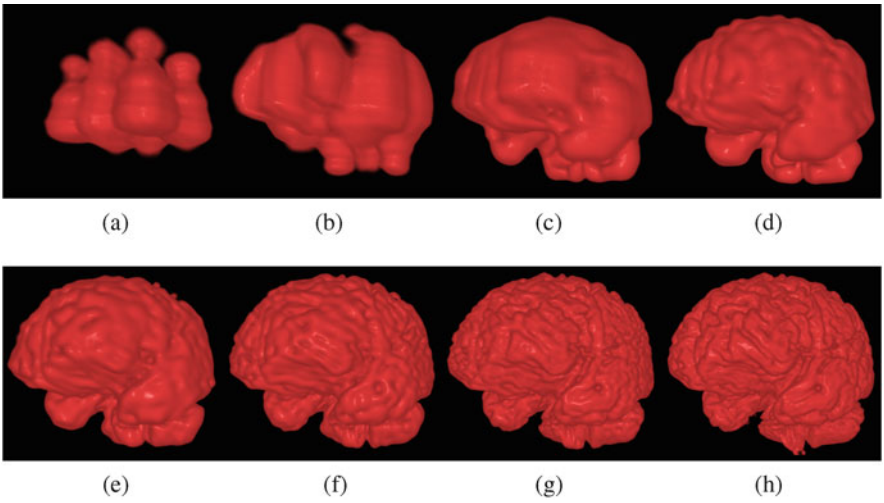


Fig. 8.12 Rendered volume images compressed by 3D-DCT. (**a**) 4 majors, (**b**) 10 majors, (**c**) 20 majors, (**d**) 35 majors, (**e**) 56 majors, (**f**) 84 majors, (**g**) 120 majors, (**h**) 165 majors

that tensor PCA and the 3D-DCT are comparable methods for the K-L transform of volumetric images.

Figures 8.17 and 8.18 show 10 right lung images of males and females (see footnote 1), respectively. The sizes of the data are listed in Table 8.3. These volumetric images are aligned using the centroids and mechanical moment axis.

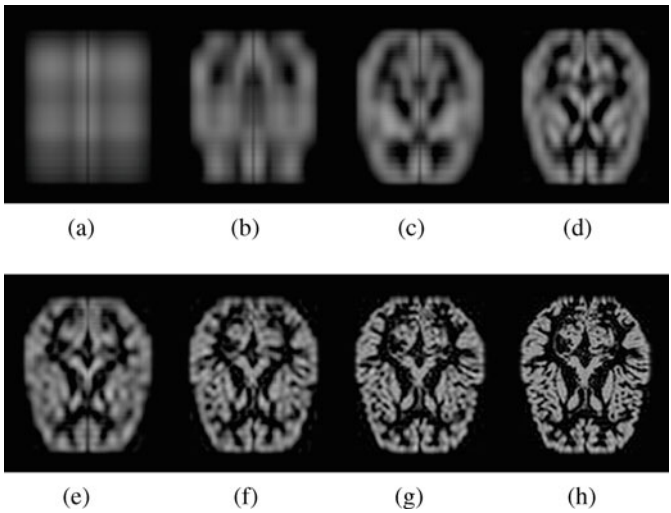


Fig. 8.13 Slice images for reconstructed volumetric images shown in Fig. 8.10. (**a**) 4 majors, (**b**) 10 majors, (**c**) 20 majors, (**d**) 35 majors, (**e**) 56 majors, (**f**) 84 majors, (**g**) 120 majors, (**h**) 165 majors

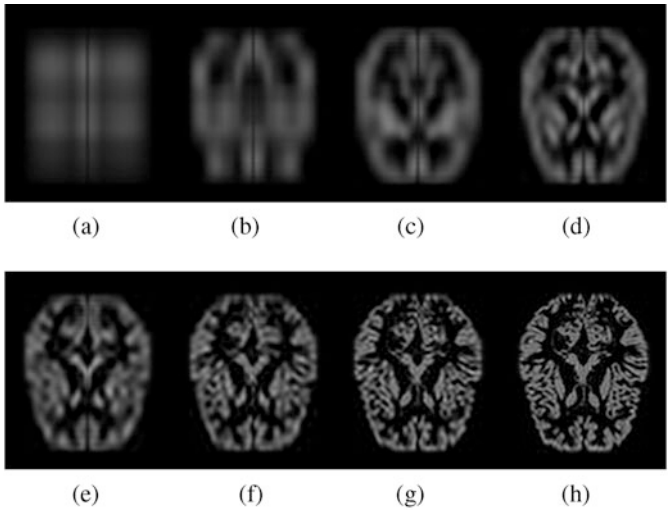


Fig. 8.14 Slice images for reconstructed volumetric images shown in Fig. 8.11. (**a**) 4 majors, (**b**) 10 majors, (**c**) 20 majors, (**d**) 35 majors, (**e**) 56 majors, (**f**) 84 majors, (**g**) 120 majors, (**h**) 165 majors

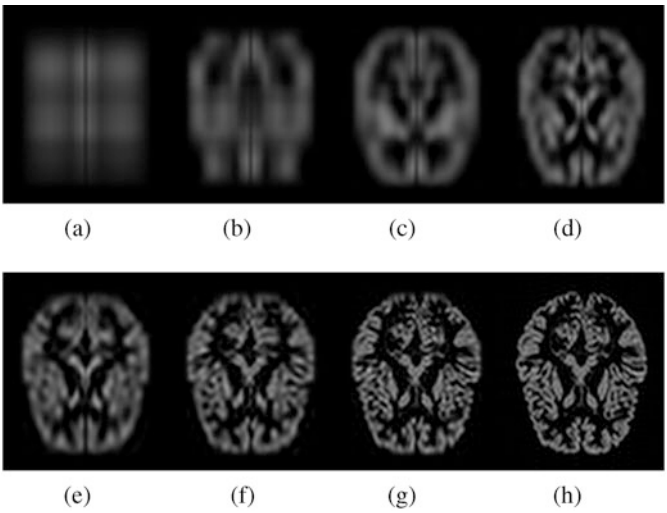


Fig. 8.15 Slice images for reconstructed volumetric images shown in Fig. 8.12. (**a**) 4 majors, (**b**) 10 majors, (**c**) 20 majors, (**d**) 35 majors, (**e**) 56 majors, (**f**) 84 majors, (**g**) 120 majors, (**h**) 165 majors

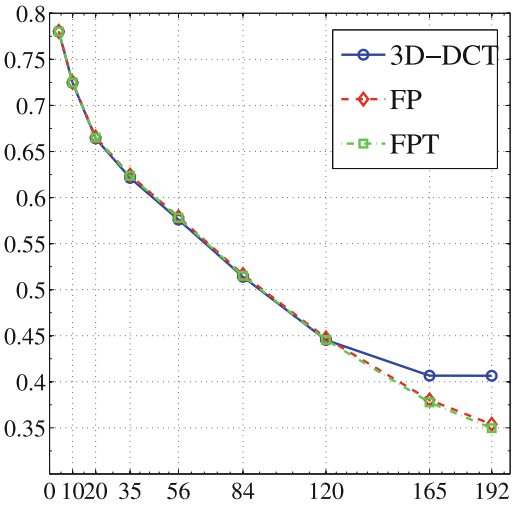


Fig. 8.16 Reconstruction error for reconstructed volumetric images. For the dimension reduction of volume data, the FP, FPT and 3D-DCT are used. The size of the dimension-reduced data is $64 \times 64 \times 64$ voxels. The vertical axis shows reconstruction errors between the original and reconstructed volumetric images evaluated by Frobenius norms. The horizontal axis shows the number of eigenvectors used for reconstruction

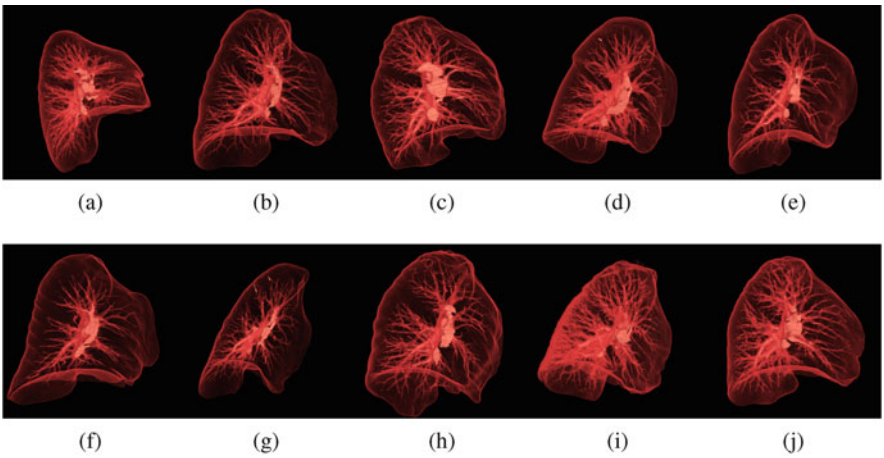


Fig. 8.17 Rendered volume images of right lungs of males. Volumetric images are extracted from CT images using annotated labels. Extracted lungs are aligned using the centroids and principal axis of the mechanical moment. From (**a**) to (**j**), the interior textures of ten volumetric male-lung images are shown

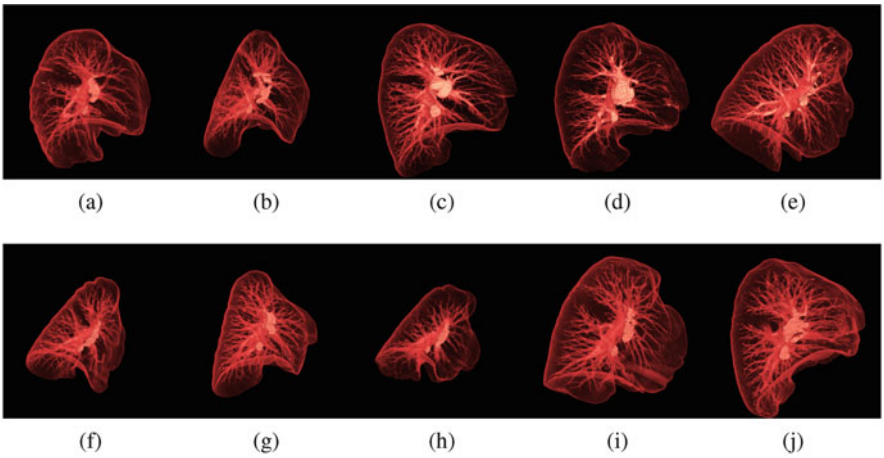
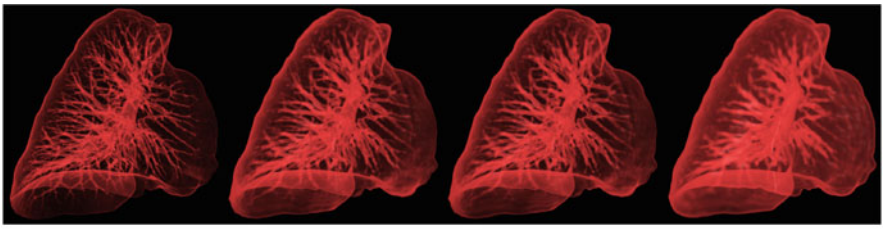


Fig. 8.18 Rendered volume images of volumetric right lungs of females. Volumetric images are extracted from CT images using annotated labels. Extracted lungs are aligned using the centroids and principal axis of the mechanical moment. From (**a**) to (**j**), the interior textures of ten volumetric female-lung images are shown

Table 8.3 Sizes and numbers of volumetric right lungs of males and females (see footnote 1)

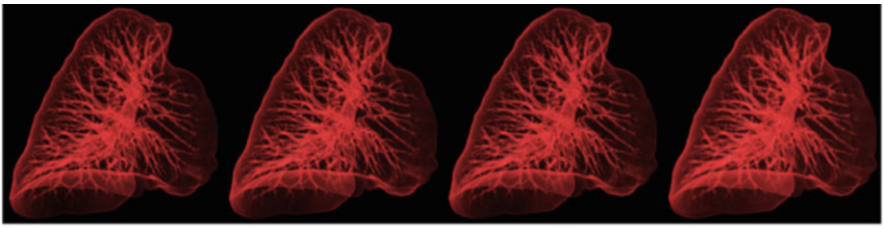
Name	♯category	♯ data/category	Original size [voxel]	Reduced size [voxel]
CA (see footnote 1) dataset	2	10	$361 \times 361 \times 361$	$128 \times 128 \times 128$

♯ represents the number of data. The original size is the original size of the volumetric data. The reduced size is reduced by using the tensor-representation-based dimension reduction



(a) (b) (c) (d)

Fig. 8.19 Comparison among original and reconstructed volumetric right lungs of a male (part 1). From left to right, the rendered original volumetric images and the volumetric images reconstructed by the FP, FPT, and 3D-DCT are shown. Images are compressed to $64 \times 64 \times 64$ voxels. (**a**) Original, (**b**) FP, (**c**) FPT, (**d**) 3D-DCT



(a) (b) (c) (d)

Fig. 8.20 Comparison among original and reconstructed volumetric right lungs of a male (part 2). From left to right, the rendered original volumetric images and the volumetric images reconstructed by the FP, FPT, and 3D-DCT are shown. Images are compressed to $128 \times 128 \times 128$ voxel. (**a**) Original, (**b**) FP, (**c**) FPT, (**d**) 3D-DCT

We evaluated the data compression results by three methods: the FP, FPT and 3D-DCT. Figures 8.19 and 8.20 show the reconstructed results for $64 \times 64 \times 64$ and $128 \times 128 \times 128$ voxels, respectively, from the lung data of Fig. 8.17b. These results show that the 3D-DCT and tensor PCA reduce the sizes of volumetric images while preserving the interior air-tube trees of the lungs. The profile curves of the CCRs of tensor PCA by three methods FP, FPT and 3D-DCT in Fig. 8.22 imply that the mathematical abilities of these three methods are compatible. Finally for the

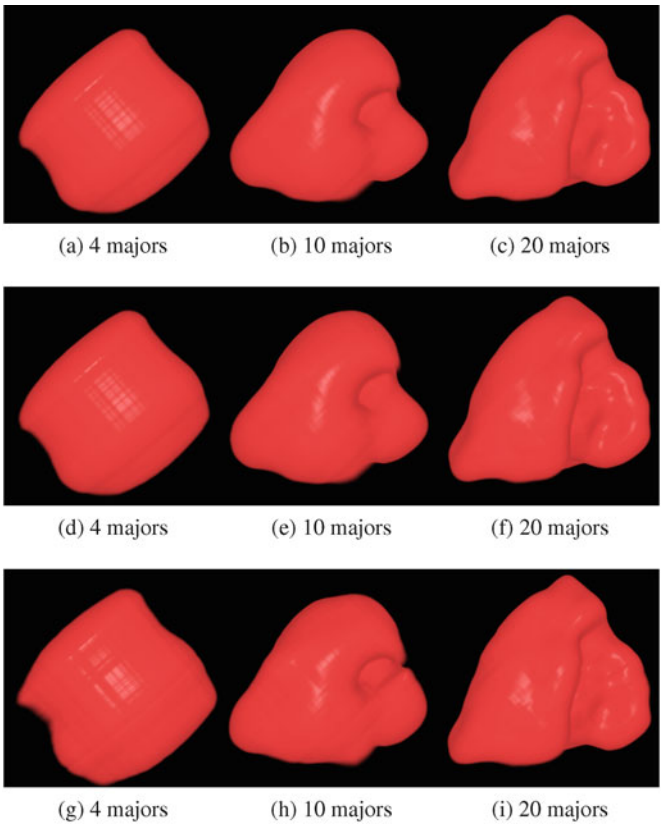


Fig. 8.21 Principal components of dimension-reduced data. For the compression of volume data, the FP, FPT and 3D-DCT are used. Top, middle and bottom rows show the results obtained by the FP, FPT and 3D-DCT, respectively. From left to right, data reconstructed by using the 4, 10 and 20 major principal components of FP are shown

lungs, we show the reconstruction results using several major principal components using three methods for the reduction of the size of the volumetric images to $128 \times 128 \times 128$ voxels. In Fig. 8.21, the top, middle and bottom rows show the results obtained by the FP, FPT and 3D-DCT, respectively. From left to right, data reconstructed by using the 4, 10 and 20 major principal components of FP are shown. These results show that the major components of the three methods possess similar geometric properties for the reconstruction of volumetric images.

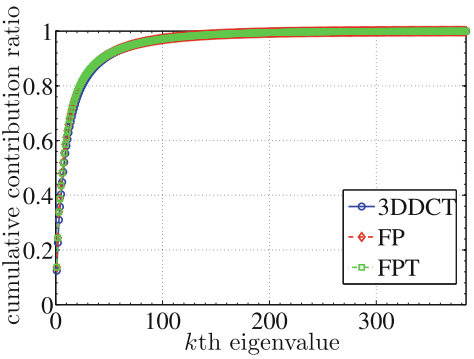


Fig. 8.22 CCR of FP for reduced volume data. The data is reduced by the FP, FPT and 3D-DCT. For the computation of the CCR, all eigenvalues of modes 1, 2 and 3 are used after sorting the eigenvalues in decreasing order

Classification and Recognition

Tensor Subspace Method

While the left ventricles are beating, they are deforming from averages [11]. These temporal deformations of moving left ventricles cause geometric perturbations to the temporal averages. These moving organs with temporal perturbations can be modelled as elements in subspaces [19].

Using the volumetric images of lungs shown in Figs. 8.17 and 8.18 we have evaluated the performance of the classification. The reduction of volumetric data by tensor PCA is applied for the classification. The original volumetric images are compressed to $128 \times 128 \times 128$ voxels by using the FP, FPT and 3D-DCT. 10 volumetric images for each gender are separated to a test set of 5 images and the learning set of 5 images.

The recognition rate is plotted against the compression rate $(361 \times 361 \times 361)/(k \times k \times k)$ for $k = 1, 2, \cdots, 32$, where k is the number of principal components in each mode. The curves in Fig. 8.22 imply that the three methods are compatible for the recognition of volumetric images.

Mutual Tensor Subspace Method

Volumetric sequences of left ventricles are extracted by using the landmarks of the endocardium of left ventricles. These landmarks are manually given and provided as part of the dataset. Figure 8.23 illustrates the extracted sequences of volumetric data for 17 patients. Figure 8.24 shows the sagittal slices of the original cardiac MRI dataset with landmarks. Table 8.4 summarises the number and size of the extracted volumetric data in all phases. Figure 8.23 illustrates the extracted sequences of volumetric data for 17 patients. Figure 8.24 shows the sagittal slices of the original cardiac MRI dataset with landmarks.

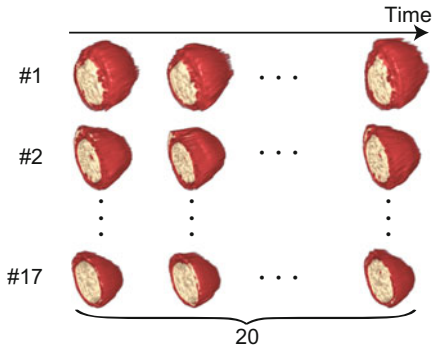


Fig. 8.23 Illustration of extracted cardiac MRI dataset. These sequences of volumetric data were extracted from the cardiac MRI dataset with landmarks of the endocardium of left ventricles [1]. As shown in Table 8.4, we have 17 sequences of volumetric data of left ventricles for 17 patients. Each sequence of volumetric data represents one cardiac beat by 20 frames. Every sequence starts with the maximally expanded state. Red and white parts of the volume rendering of the data represent the muscle and inner space of left ventricles, respectively. We set the centre of the first sagittal slice of each volume data to the centre of the slice

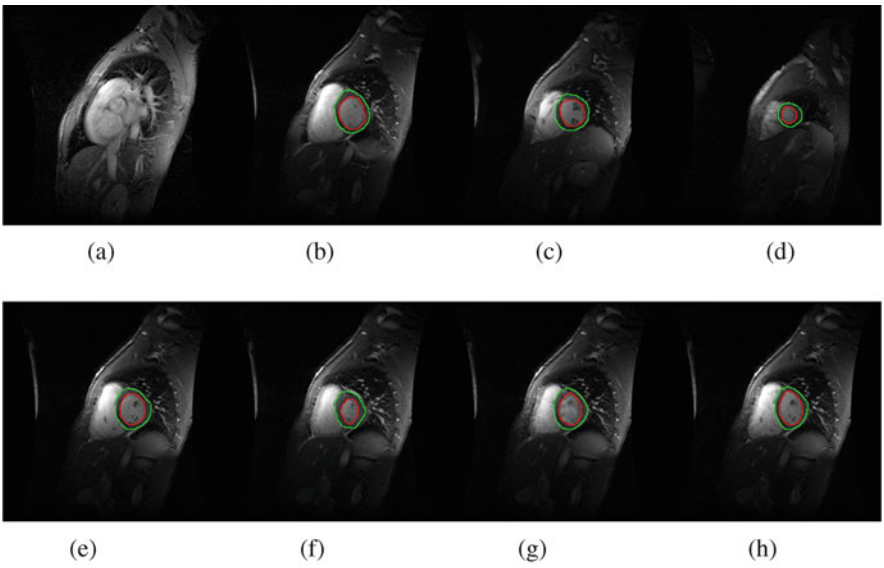


Fig. 8.24 Cardiac MRI dataset [1]. These images represent the sagittal slices of volumetric data. The sagittal direction is expressed by the z-axis for the description of depth. Green and red lines depict the endocardium and epicardium of the left ventricle, respectively. From (**e**) to (**h**), the ventricle shrinks then expands. (**a**) $z = 1, t = 1$, (**b**) $z = 4, t = 1$, (**c**) $z = 7, t = 1$, (**d**) $z = 10, t = 1$, (**e**) $z = 5, t = 1$, (**f**) $z = 5, t = 5$, (**g**) $z = 5, t = 10$, (**h**) $z = 5, t = 15$

Table 8.4 Size and number of volumetric data of left ventricles (see footnote 1)

	♯category	♯data /category	Data size [voxel]	Reduced data size [voxel]
CA data (see footnote 1)	17	20	$81 \times 81 \times 63$	$d \times d \times d$

♯category represents the number of individuals. ♯data/category represents the number of frames in one sequence of left ventricles. The data size is the original size of the volumetric data. The reduced data size is the size of the volume data after reduction. We set $d \in \{8, 16, 32\}$

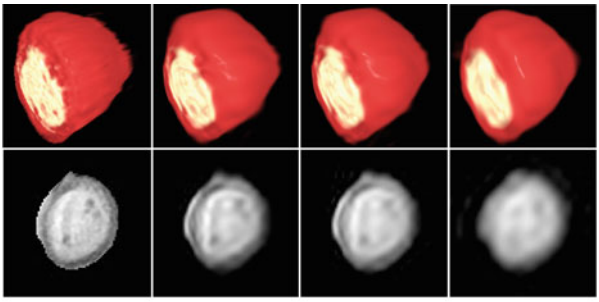


Fig. 8.25 Shape and inner texture of original image and volumetric images of left ventricles reconstructed from compressed data by the FP, FPT and 3D-DCT. Upper and lower rows show rendered volume data and sagittal slice of the volumetric data, respectively. In upper row, red and white parts depict the muscle of the heart and the texture, respectively, for the original and approximated volumetric images. In these approximated volumetric images the data are reduced to the size $16 \times 16 \times 16$ voxels. (**a**) Original, (**b**) FP, (**c**) FPT, (**d**) 3D-DCT

We separate these dimension-reduced data into training and test data. From the training data set, the tensor subspace of each category is constructed. For the dimension reduction, we apply HOSVD, the FP and FPT to all the extracted volumetric data in all categories. For evaluations of the robustness and stability of the methods with respect to the size of the data, we set the sizes of the dimension-reduced data to $8 \times 8 \times 8$, $16 \times 16 \times 16$ and $32 \times 32 \times 32$ voxels.

Figure 8.25 illustrates the results of the compression between the original and dimension-reduced data for the three methods. In Fig. 8.25a–d, rendered images of the original and reconstructed volume data are presented. For the data reduced by the FP and FPT, the shapes of the volumetric data reconstructed from the compressed data appear to be almost the same. The data reconstructed from the data reduced by the 3D-DCT have the closest shape to the original volumetric data. In Fig. 8.25e–h, the differences in the appearances between the sagittal slices of the reconstructed data and the original shape are compared. Compared to the original data shown in Fig 8.25a, the 3D-DCT gives a blurred inner texture as shown in Fig. 8.25h. As shown in Figs. 8.25f, g, the compression by the FP and FPT extracts the outline shape of the ventricle without the inner texture. Figure 8.26 illustrates data reconstructed from the principal components of the dimension-reduced volume data. This result shows that the principal components of the dimension-reduced volume data are almost the same for the three methods.

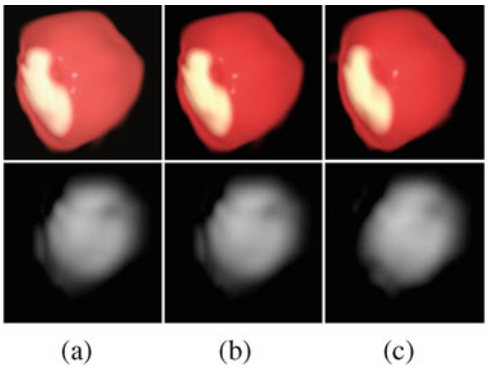


Fig. 8.26 Extracted principal components of dimension-reduced volume data. The dimension-reduction of data is archived by the FP, FPT and 3D-DCT. The major 20 principal eigenvectors for three mode are selected for extraction. Top and bottom images show shapes and inner textures of the reconstructed volumetric images. (**a**) FP, (**b**) FPT, (**c**) 3D-DCT

In the dimension-reduced data, each sequence consists of 20 frames. We use odd and even frames in the compressed data as training and test data, respectively. Applying the FP to training data of each category, we construct 17 categories as tensor subspaces for the TSM and MTSM. We use only odd frames for the construction of tensor subspaces of categories to evaluate robustness. If the MTSM classifies a category of even frames, we conclude that this classifier is robust to the small geometric changes between frames.

The recognition rate is defined as the successful classification ratio of individuals in 1000 classifications. In the selection of query for the TSM, we randomly select one of 17 individuals and one of the test data of the individual. From a left ventricle sequence of a patient, we construct the query subspaces for the MTSM. We set the dimensions of query subspaces are one, two and three. Figures 8.27 and 8.28 show the recognition rate of left ventricles for the TSM and MTSM, respectively.

In Fig. 8.27, the profiles of recognition curves for HOSVD, the FP, FPT and 3D-DCT are almost the same for compression ratios higher than 10^3. Furthermore, for the compression ratios higher than 10^3, the data compressed by these four methods derive almost the same recognition rates. These recognition rates are the same as those of the tensors of the original size. Moreover, the TSM with five major eigenvectors in each mode provides accurate recognition rate. Figure 8.28a–c show the recognition rate of the MTSM if the query subspace is spanned by one query. The results show that the recognition properties are almost the same for data with the reduced sizes of $8 \times 8 \times 8$, $16 \times 16 \times 16$ and $32 \times 32 \times 32$ voxels. The results in Fig. 8.28d–i imply that the MTSM achieves more robust recognition than the TSM against small geometric perturbations by using a query subspace if the query subspace is spanned by a few queries with geometric perturbations. These numerical examples lead to the conclusion that the 3D-DCT accurately approximates the performance of tensor PCA. Furthermore, the recognition by the TSM and MTSM is accurate and robust for volumetric images containing geometric perturbations as temporal deformation.

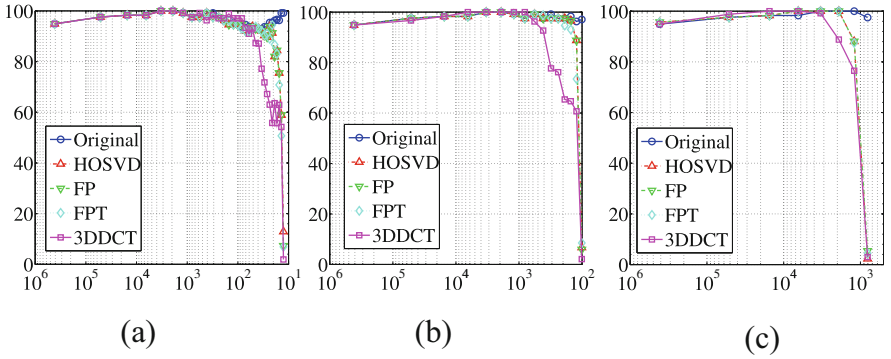


Fig. 8.27 Recognition rates of left ventricles for original and compressed tensors using the TSM. The volumetric images are reduced to (**a**) $32 \times 32 \times 32$, (**b**) $16 \times 16 \times 16$ and (**c**) $8 \times 8 \times 8$ voxels. HOSVD, the FP, FPT and 3D-DCT reduces the image sizes. Vertical and horizontal axes represent recognition rate and compression ratio, respectively. For the original size $D = 81 \times 81 \times 63$ and the reduced size $K = k \times k' \times k'$, the compression ratio is given by D/K

Conclusions

We have developed two relaxed closed forms for tensor principal component analysis (PCA). The first method solves a system of eigenmatrix problems using the unfolding of a tensor instead of solving a variational optimisation problem iteratively. Our method solves a system of variational optimisation problems derived from the original expression of the Tucker-3 decomposition with the orthogonal constraints for solutions. The second method is based on the low-pass filtering of multidimensional signals using the discrete cosine transform (DCT) since the DCT efficiently approximates the Karhunen-Loève (K-L) transform. Such orthogonal-projection-based data compression extracts outline shapes of biomedical objects such as organs and the interior structures of cells. Furthermore, we have numerically evaluated the performance of these algorithms for compressing volumetric medical images. Moreover, we expressed the pyramid transform (PT) of volumetric data as the mode decomposition of tensors. This algebraic property of the PT allow us to geometrically compare data reduction by the PT with that of tensor PCA for data compression and reduction.

We applied three-way tensor PCA to the extraction of outline shapes of volumetric data and their classification. In the numerical examples, we demonstrated that three-way tensor PCA extracts the outline shape of volumetric images. Furthermore, the tensor subspace method (TSM) accurately classifies the extracted outline shapes. Moreover, we showed that the 3D-DCT-based reduction approximated both the outline shape and the texture of volumetric images.

Furthermore, we developed tensor-based multilinear classifiers, the tensor subspace method (TSM) and mutual tensor subspace method (MTSM) for third-order tensors. In the numerical examples, we evaluated the performance of dimension reduction by HOSVD, the FP, FPT and 3D-DCT for the recognition of individual

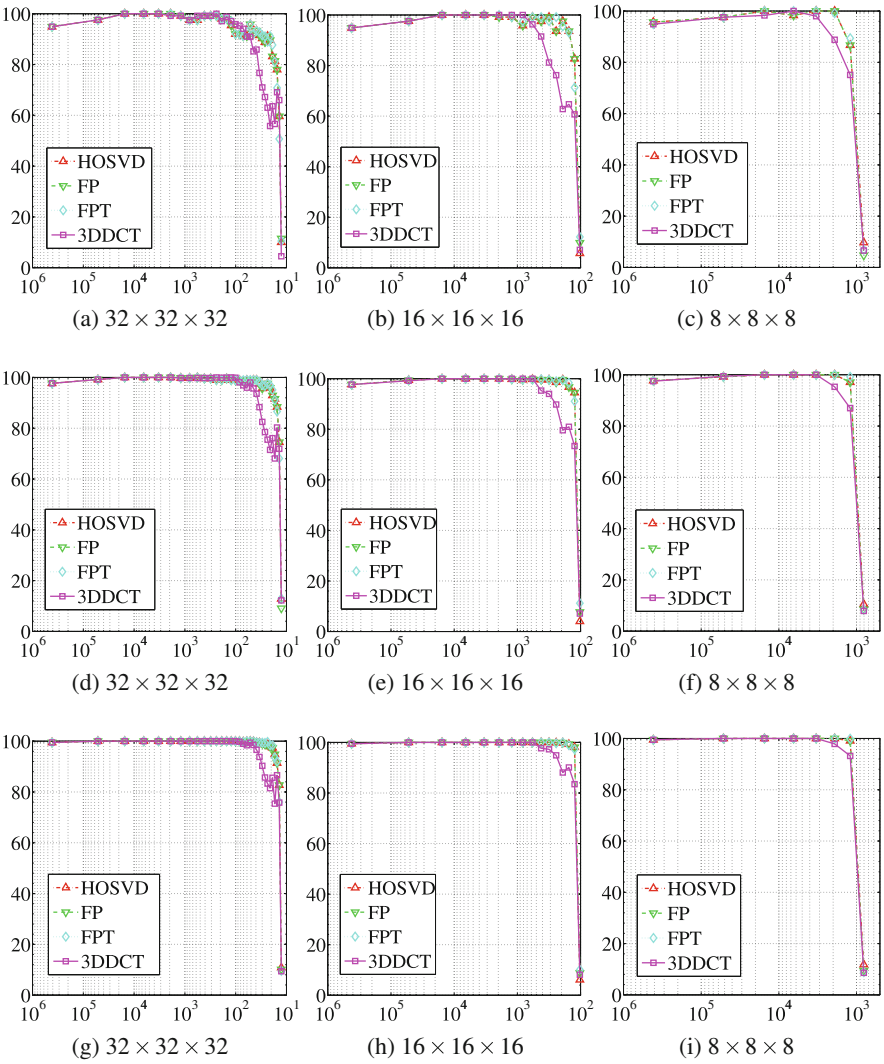


Fig. 8.28 Recognition rates of left ventricles for compressed tensors. We adopt the reduces sizes of HOSVD, the FP, FPT and 3D-DCT compress the original volumetric image to $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. To construct a query subspaces of TMSM, we select one, two and three queries. The top, middle and bottom rows show recognition rates for the case of one, two and three images for the reconstruction of query subspaces, respectively. Vertical and horizontal axes represent the recognition rate and compression ratio, respectively. The original size $D = 81 \times 81 \times 63$ is reduced size $K = k \times k' \times k'$. The compression ratio is measured by D/K, where the original and compressed sizes are $D = 81 \times 81 \times 63$ and $K = k \times k' \times k'$, respectively

left ventricles. In the reduction, the FP and FPT extracted the outline shapes of left ventricles. The results of the evaluations showed that the dimension reduction by HOSVD possesses the same performance as the FP and FPT, which are non-iterative computation procedures. In the evaluations, the TSM and MTSM accurately recognised individual left ventricles even though cardiac MRI images include geometric perturbations. Using a query subspace spanned by more than one frame, the MTSM achieved more stable recognition and robustness against geometric perturbations than the TSM.

In traditional methods in medical image analysis, the outline shapes of objects such as organs and the statistical properties of interior textures are independently extracted using separate methods. However, tensor PCA for volumetric images allows us to simultaneously extract both the outline shapes of volumetric objects and the statistical properties of the interior textures of volumetric images from data projected onto a low-dimensional linear subspace spanned by tensors. The extension of the algorithms to higher-order multi-way data analysis, such as the spatio-temporal volumetric analysis of moving and deforming objects, is straightforward using higher-order tensors.

The time complexity of PCA for vector is $\mathcal{O}(n^3)$ since the main procedure of PCA is achieved by solving eigenvalue problem. TCPA requires $\mathcal{O}(k \times n^3)$ computation times where k is the iteration times to guaranty numerically convergence of the iteration process. On the other hand the 3D-DCT achieves in $\mathcal{O}(n \log n)$ computation times with numerically acceptable accuracy. Since the DCT matrix is the eigenmatrix of the Laplacian operation, the 3D-DCT method is a harmonic analysis for volumetric data. PCA is a method to derive eigenfunctions from data. The methods based on PCA allow us to operate both shapes and interior textures of volumetric data. This is an advantage of 3D-DCT over Laplace-Beltrami eigenfunctions for shape analysis [3].

Acknowledgements This research was supported by the "Multidisciplinary Computational Anatomy and Its Application to Highly Intelligent Diagnosis and Therapy" project funded by a Grant-in-Aid for Scientific Research on Innovative Areas from MEXT, Japan, and by Grants-in-Aid for Scientific Research funded by the Japan Society for the Promotion of Science.

Appendix

The linear reduction operation R and its dual E are defined as

$$g(\boldsymbol{x}) = Rf(\boldsymbol{y}) = \int_{\mathbf{R}^3} w_3(\boldsymbol{u}) f(2\boldsymbol{x} - \boldsymbol{u}) d\boldsymbol{u}, \tag{8.60}$$

$$Eg(\boldsymbol{x}) = 2^3 \int_{\mathbb{R}^3} w_3(\boldsymbol{u}) g\left(\frac{\boldsymbol{x} - \boldsymbol{u}}{2}\right) d\boldsymbol{u}. \tag{8.61}$$

where $w_3(\boldsymbol{x}) = w(x)w(y)w(z)$ for $\boldsymbol{x} = (x, y, z)^\top$ and

$$w(s) = \begin{cases} \frac{1}{2}(1 - \frac{|s|}{2}), & |s| \le 2 \\ 0, & |s| > 2 \end{cases} \tag{8.62}$$

These processes are achieved by computing a weighted average of the image values in a finite small region, which is called the window for the operation.

Setting $w_{\pm 1} = \frac{1}{4}$ and $w_0 = \frac{1}{2}$, for the two-dimensional sampled function $f_{ijk} = f(\Delta i, \Delta j, \Delta k)$, the transforms of Eqs. (8.60) and (8.61) are respectively described as

$$Rf_{kmn} = \sum_{p,q,r=-1}^{1} w_p w_q w_r f_{2k-p\ 2m-q\ 2n-p}, \tag{8.63}$$

$$Ef_{kmn} = \frac{1}{2^3} \sum_{p,q,r=-2}^{2} w_p w_q w_r f_{\frac{k-p}{2}\ \frac{m-q}{2}\ \frac{n-r}{2}}, \tag{8.64}$$

where the summation is achieved for $\frac{(k-p)}{2}$, $\frac{(m-q)}{2}$ and $\frac{(n-r)}{2}$ being integers. These procedures are called the pyramid transform and extension, respectively. These two operations involve the reduction and expansion of the image sizes. Therefore, image features are extracted in the higher-layer images of the pyramid transform.

Setting

$$\boldsymbol{R} = \frac{1}{4}(\boldsymbol{I} \otimes (0, 1)^\top)(\boldsymbol{D} + 4\boldsymbol{I}) \tag{8.65}$$

for the second-order differential matrix $\boldsymbol{D}$ with the Neumann condition such that

$$\boldsymbol{D} = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix}, \tag{8.66}$$

the three-dimensional pyramid transform

$$g_{pqr} = \sum_{i,j=-1}^{1} w_i w_j w_k f_{2p-i\ 2q-j\ 2r-k,}, \tag{8.67}$$

is redescribed

$$\mathscr{Y} = \mathscr{X} \times_1 \boldsymbol{R} \times_2 \boldsymbol{R} \times_3 \boldsymbol{R} \tag{8.68}$$

using the Tucker-3 decomposition of $\mathscr{X}$.

Since the eigenmatrix of $\boldsymbol{D}$ is the DCT-II matrix the three-dimensional pyramid transform processes the following property.

Property 8.2 Setting $L^N = \mathsf{L}(\{\boldsymbol{\varphi}_k\}_{k=0}^{2^N-1})$, for three-dimensional images, the pyramid transform is a linear transform from $L^N \times L^N \times L^N$ to $L^{\frac{N}{2}} \times L^{\frac{N}{2}} \times L^{\frac{N}{2}}$.

Equation (8.68) directly derives outlines of volumetric shapes by enforcing and inhibiting low- and high-frequency parts, respectively, on the DCT of the volumetric shape. Therefore, the dominant operation in the pyramid transform is the relaxed Karhunen-Loève transform using the DCT.

References

1. A. Andreopoulos, J.K. Tsotsos, Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI. Med. Image Anal. **12**, 335–357 (2008)
2. B. Aubert-Broche, M. Griffin, G.B. Pike, A.C. Evans, D.L. Collins, 20 new digital brain phantoms for creation of validation image data bases. IEEE Trans. Med. Imaging **25**, 1410–1416 (2006)
3. A. Bronstein, M. Bronstein, R. Kimmel, *Numerical Geometry of Non-Rigid Shapes* (Springer, Berlin, 2009)
4. A. Cichocki, R. Zdunek, A.-H. Phan, S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation* (Wiley, Hoboken, 2009)
5. R. Davies, C. Twining, C. Taylor, *Statistical Models of Shape Optimisation and Evaluation* (Springer, Berlin, 2008)
6. U. Grenander, M. Miller, *Pattern Theory: From Representation to Inference* (OUP, New York, 2007)
7. M. Hamidi, J. Pearl, Comparison of the cosine Fourier transform of Markov-1 signals. IEEE ASSP **24**, 428–429 (1976)
8. H. Itoh, A. Imiya, T. Sakai, Mathematical aspects of tensor subspace method, in *Structural, Syntactic, and Statistical Pattern Recognition*. Lecture Notes in Computer Science, vol. 10029 (Springer, Cham, 2016), pp. 37–48
9. T. Iijima, *Pattern Recognition* (Corona-sha, Tokyo, 1974) (in Japanese)
10. A. Imiya, U. Eckhardt, The Euler characteristics of discrete objects and discrete quasi-objects. Comput. Vis. Image Underst. **75**, 307–318 (1999)
11. S. Inagaki, H. Itoh, A. Imiya, Variational multiple warping for cardiac image analysis, in *Computer Analysis of Images and Patterns*. Lecture Notes in Computer Science, vol. 10425 (Springer, Cham, 2015), pp. 749–759
12. S. Inagaki, H. Itoh, A. Imiya, Multiple alignment of spatiotemporal deformable objects for the average-organ computation, in *ECCV2014 Workshops 2014*. Lecture Notes in Computer Science, vol. 8928 (Springer, Cham, 2015), pp. 353–366
13. H. Itoh, T. Sakai, K. Kawamoto, A. Imiya, Topology-preserving dimension-reduction methods for image pattern recognition, in *Image Analysis*. Lecture Notes in Computer Science, vol. 7944 (Springer, Berlin, 2013), pp. 195–204
14. H. Itoh, A. Imiya, T. Sakai, Pattern recognition in multilinear space and its applications: mathematics, computational algorithms and numerical validations. Mach. Vis. Appl. **27**, 1259–1273 (2016)
15. M. Itskov, *Tensor Algebra and Tensor Analysis for Engineers* (Springer, Berlin, 2013)

16. T.G. Kolda, B.W. Bader, Tensor decompositions and applications. SIAM Rev. **51**, 455–500 (2008)
17. P.M. Kroonenberg, *Applied Multiway Data Analysis* (Wiley, Hoboken, 2008)
18. L.D. Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**, 1253–1278 (2000)
19. K. Maeda, From the subspace methods to the mutual subspace method, in *Computer Vision*, ed. by R. Cipolla, S. Battiato, G.M. Farinella. Studies in Computational Intelligence, vol. 285 (Springer, Berlin, 2010), pp. 135–156
20. A. Malcev, *Foundations of Linear Algebra*, Russian edition (1948) (English translation W.H. Freeman and Company, 1963)
21. J.M. Marron, A.M. Alonso, Overview of object oriented data analysis. Biom. J. **56**, 732–753 (2014)
22. M. Mørup, Applications of tensor (multiway array) factorizations and decompositions in data mining. Wiley Interdiscip. Rev. Data Min. Knowl. Disc. **1**, 24–40 (2011)
23. T.M.W. Nye, Principal component analysis in the space of phylogenetic trees. Ann. Stat. **39**, 2716–2739 (2011)
24. E. Oja, *Subspace Methods of Pattern Recognition* (Research Studies Press, Letchworth, 1983)
25. N. Otsu: Mathematical studies on feature extraction in pattern recognition, Researches of the Electrotechnical Laboratory, 818 (1981; in Japanese)
26. T. Sakai, M. Narita, T. Komazaki, H. Nishiguchi, A. Imiya, Image hierarchy in Gaussian scale space, in *Advances in Imaging and Electron Physics*, vol. 165 (Academic, Cambridge, 2013), pp. 175–263
27. G. Strang, T. Nguyen, *Wavelets and Filter Banks* (Wellesley-Cambridge Press, Wellesley, 1996)
28. D.W. Thompson, *On Growth and Form (The Complete Revised Edition)* (Dover, Mineola, 1992)
29. S. Watanabe, N. Pakvasa, Subspace method of pattern recognition, in *Proceedings of the 1st International Joint Conference of Pattern Recognition* (1973), pp. 25–32
30. G.W. Weber, F.L. Bookstein, *Virtual Anthropology: A Guide to a New Interdisciplinary Field* (Springer, Berlin, 2011)
31. L. Younes, *Shapes and Diffeomorphisms* (Springer, Berlin, 2010)
32. M.L. Zelditch, D.L. Swiderski, H.D. Sheets, *Geometric Morphometrics for Biologists: A Primer*, 2nd edn. (Academic, Cambridge, 2012)

Part IV
Machine Learning and Big Data Analysis

Chapter 9
An Incremental Reseeding Strategy for Clustering

Xavier Bresson, Huiyi Hu, Thomas Laurent, Arthur Szlam, and James von Brecht

Abstract We propose an easy-to-implement and highly parallelizable algorithm for multiway graph partitioning. The algorithm proceeds by alternating three simple routines in an iterative fashion: diffusion, thresholding, and random sampling. We demonstrate experimentally that the proper combination of these ingredients leads to an algorithm that achieves state-of-the-art performance in terms of cluster purity on standard benchmark data sets. We also describe a coarsen, cluster and refine approach similar to Dhillon et al. (IEEE Trans Pattern Anal Mach Intell 29(11):1944–1957, 2007) and Karypis and Kumar (SIAM J Sci Comput 20(1):359–392, 1998) that removes an order of magnitude from the runtime of our algorithm while still maintaining competitive accuracy.

X. Bresson (✉)
School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
e-mail: xbresson@ntu.edu.sg

H. Hu
Google Inc, Mountain View, CA, USA
e-mail: clarahu@google.com

T. Laurent
Department of Mathematics, Loyola Marymount University, Los Angeles, CA, USA
e-mail: tlaurent@lmu.edu

A. Szlam
Facebook AI Research, New York, NY, USA
e-mail: aszlam@fb.com

J. von Brecht
Department of Mathematics, California State University, Long Beach, CA, USA
e-mail: james.vonbrecht@csulb.edu

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_9

Introduction

One of the most basic unsupervised learning tasks is to automatically partition data into clusters based on similarity. A standard scenario is that the data are represented as a weighted graph. Data points correspond to vertices on the graph while edges between vertices encode the similarity between data points. Many of the most popular and widely used clustering algorithms, such as spectral clustering, fall into this category. Despite the vast literature on graph-based clustering, the field remains an active area for both theoretical and practical research.

In this work, we propose a resampling-based spectral algorithm for multiway graph partitioning. On graphs that contain reasonably well-balanced clusters of medium scale, the algorithm provides a strong combination of accuracy, efficiency and robustness to noise in the graph construction process. The algorithm is also exceedingly simple, intuitive and trivial to implement. A MATLAB code consists of fewer than 40 lines, for instance (the code is provided in the Appendix). It also parallelizes trivially, and can therefore scale gracefully to large numbers of clusters as well as to graphs with large numbers of vertices.

We validate these claims via an extensive experimental evaluation of the algorithm. We also provide a detailed algorithmic comparison using recent clustering algorithms that claim state-of-the-art results. These experiments demonstrate that our algorithm achieves state-of-the-art performance in terms of cluster purity while running faster than the other highly accurate clustering methods (e.g. [4, 16]) that we compare against. We also provide experiments to demonstrate the robustness of the algorithm with respect to noise and perturbations in the underlying graph. While many highly accurate algorithms exhibit a sharp decrease in accuracy if the input graph is corrupted by noise, our algorithm remains stable: the accuracy of our algorithm decays slowly and gracefully with increasing levels of noise. These results, when taken together, lead to an algorithm with a quite appealing combination of simplicity, performance and ease in out-of-the-box usage.

Description of the Algorithm

The main idea behind our algorithm arises from a well-known and widely used property of the random walk on a graph. Specifically, a random walker started in a low conductance cluster is unlikely to leave that cluster quickly [11]. This fact provides the basis for transductive label propagation methods [18] as well as for "local" clustering methods [14]. In label propagation, for instance, an oracle provides a set of labeled vertices that are propagated along the graph using a random walk matrix or a diffusion matrix. Each unlabeled vertex is then associated to the label which, after being propagated, best represents the given unlabeled vertex.

Our algorithm simply iterates upon this basic idea. Assume that the graph has R well-defined clusters of comparable size and low conductance. If we knew these

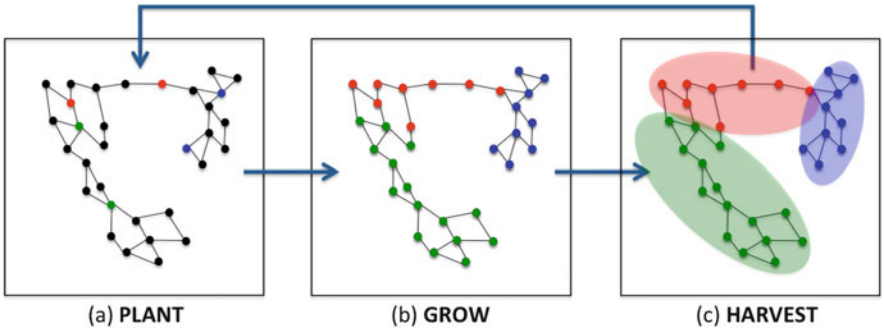


Fig. 9.1 Illustration of the Incremental Reseeding (INCRES) Algorithm for $R = 3$ clusters. The various colors red, blue, and green identify the clusters. (**a**) At this stage of the algorithm, $s = 2$ seeds are randomly planted in the clusters computed from the previous iteration. (**b**) The seeds grow with the random walk operator. (**c**) A new partition of the graph is obtained and used to plant $s + ds$ seeds into these clusters at the next iteration

clusters in advance, we could then select a handful of "seed" vertices in the center of each cluster. We would then expect to obtain good results from a transductive label propagation by using these seeds as labels. In an unsupervised context we cannot, of course, *a-priori* place seeds in the center of each cluster. To overcome this, we instead place a handful of seeds at random. We then apply a random walk matrix or diffusion matrix a few times to propagate these seeds. We finally obtain a temporary clustering by assigning each vertex to the seed which, after propagation, best represents the vertex. We then choose new seeds from these temporary clusters and iterate the process. If the clusters improve then the seeds will likely improve, and vice-versa. This incites a feed-back loop and we get a virtuous cycle. We can then excite the speed and improve the quality of this cycle by gradually drawing more and more seeds throughout the process. We refer to this idea as an *incremental reseeding strategy*. Figure 9.1 depicts this cyclic process graphically.

The Basic Algorithm

To formalize these ideas, let $G = (V, W)$ denote a weighted, connected graph on N vertices $V = \{v_1, \ldots, v_N\}$ with edge weights $W = \{W_{ij}\}_{i,j=1}^{N}$ that encode a measure of similarity between each pair (i, j) of vertices. Let D denote the diagonal matrix of (weighted) vertex degrees. The algorithm starts from a random partition $\mathscr{P} = (\mathscr{C}_1, \ldots, \mathscr{C}_R), \mathscr{C}_1 \cup \cdots \cup \mathscr{C}_R = V, \mathscr{C}_r \cap \mathscr{C}_q = \emptyset\ (r \neq q)$ of the vertices. In other words, each v_i is assigned to one of the R clusters uniformly at random. Let $s = 1$ denote the initial number of seeds. At each of the successive iteration, we update the current partition $\mathscr{P} = (\mathscr{C}_1, \ldots, \mathscr{C}_R)$ according to the steps outlined in Algorithm 9.1. At the beginning of each iteration, the routine PLANT($\mathscr{P}$,s) will sample s seeds from each of the R clusters $\mathscr{C}_r$ in the current partition $\mathscr{P}$ uniformly

Algorithm 9.1: INCRES algorithm

Input: Similarity matrix W, seed increment ds, number of clusters R.
Initialization: $s = 1$, random partition $\mathscr{P}$.
repeat
  $F \leftarrow \text{PLANT}(\mathscr{P}, s)$
  $U \leftarrow \text{GROW}(F, W)$
  $\mathscr{P} \leftarrow \text{HARVEST}(U)$
  $s \leftarrow s + ds$
until $\mathscr{P}$ *converges*;
Output: $\mathscr{P}$

at random. These Rs seeds furnish the temporary labels that the GROW routine then propagates along the graph using a random walk matrix. We initialize GROW with an $N \times R$ matrix U, where $U_{i,r} = 1$ if vertex v_i was drawn from cluster $\mathscr{C}_r$ and $U_{i,r} = 0$ otherwise. We then iteratively apply the random walk transition matrix WD^{-1} to U until each vertex has a nonzero probability of being visited by a random walker, i.e. until each entry $U_{i,r}$ is nonzero. Finally, the routine HARVEST simply assigns each vertex v_i to its most likely cluster, or in other words the cluster for which $U_{i,r}$ is maximal. This produces a new partition $\mathscr{P}$ of the vertices into R clusters. We then increment s to $s + ds$, and use this partition and number of seeds s to initialize PLANT at the beginning of the next iteration. We refer to this overall procedure as the Incremental Reseeding Algorithm (INCRES).

function PLANT$(\mathscr{P}, s)$
  Initialize F as an N-by-R matrix of zeros.
  for $r = 1$ to R **do**
    for $k = 1$ to s **do**
      Draw at random a vertex i in cluster $\mathscr{C}_r$.
      $F_{i,r} \leftarrow F_{i,r} + 1$
    end for
  end for
  return F.
end function

function GROW(F, W)
  Initialize U as an N-by-R matrix equal to F.
  while $\min_i \min_r U_{i,r} = 0$ **do**
    $U \leftarrow \left(WD^{-1}\right) U$
  end while
  return U.
end function

function HARVEST(U)
  for $r = 1$ to R **do**
    $\mathscr{C}_r = \{i : U_{i,r} \geq U_{i,q} \text{ for all } q\}$
  end for
  return $\mathscr{P} = (\mathscr{C}_1, \ldots, \mathscr{C}_R)$
end function

The overall routine has only a single parameter ds that controls the linear rate at which the number of seeds drawn at each iteration increases. In practice, we select

$$ds = \textbf{speed} \times 10^{-4} \times \frac{N}{R} \tag{9.1}$$

for some proportionality constant **speed** between one and ten. By rescaling ds in this way, a constant of proportionality **speed** $= 1$ corresponds to a total of $s = 0.1N/R$ seeds planted in each cluster after 1000 iterations. Assuming well-balanced clusters of roughly equal size, approximately one-tenth of each cluster is sampled after 1000 iterations. Drawing a significant fraction of each cluster will cause the subsequent clustering to stabilize, leading to eventual convergence of the algorithm. Using around 10% of labels in each cluster is a typical level at which INCRES will stabilize.

The parameter ds therefore represents a "timestep" for INCRES, and the overall algorithm behaves well with respect to this parameter. In general, small increments ds will lead to slower convergence at higher accuracy while larger increments ds will lead to faster convergence but potentially less accurate solutions. Our experiments show that **speed** $= 1$ works remarkably well for a large variety of data sets. We also provide results for **speed** $= 5$, which yields faster stabilization with slightly less accuracy, to show that the algorithm is indeed robust and predictable with respect to the choice of this parameter.

The general INCRES framework also proves robust to implementation choices for the three main routines. For instance, in addition to the random walk matrix WD^{-1}, there exists a variety of alternative means to propagate labels along a graph. By-and-large, the overall INCRES strategy does not depend heavily upon the particular implementation of GROW, so long as it realizes the basic idea of label propagation in one form or another. For instance, we have found that replacing the random walk step $U \leftarrow WD^{-1}U$ with a diffusion step $U \leftarrow D^{-1}WU$ or $U \leftarrow D^{-1/2}WD^{-1/2}U$ will give similar results in many circumstances. Occasionally, we have found that utilizing a "personalized Page-Rank" step

$$U \leftarrow \alpha WD^{-1}U + (1-\alpha)F \tag{9.2}$$

can give better performance on small data sets that contain a large (relative to the size of the data set) number of clusters. Here the parameter $0 < \alpha < 1$ denotes a length-scale that controls the extent of diffusion and F denotes the input to the GROW routine. A propagation step of the form (9.2) is also used in Pagerank-NIBBLE [1] and NMFR [16], up to replacing WD^{-1} with $D^{-1/2}WD^{-1/2}$ in the latter case. As another example, choosing to sample with or without replacement in the PLANT routine leads to essentially no significant difference in the resultant clusterings.

Relation with Other Work

Our methodology relies upon and incorporates number of ideas from transductive learning. In particular, we leverage the notion of label propagation [18]. In the standard label propagation framework, an oracle provides a set of labeled points or vertices. These labeled vertices form either nonzero initial conditions or heat sources for a discrete heat equation on the graph. The second step of the INCRES algorithm (the GROW routine) precisely corresponds with a label propagation of the random labels returned from the first step of the algorithm (the PLANT routine).

The proposed algorithm has a quite intuitive and appealing motivation based on a first principle approach to graph partitioning, in the sense that INCRES combines a simple iterative application of label propagation and thresholding to handle unsupervised learning. At a deeper level, we may also view INCRES as a leading-order approximation of the algorithm from [4] that optimizes the product cut (PCUT) objective. The PCUT algorithm optimizes the "likelihood"

$$\mathscr{L}(\mathscr{P}) := \prod_{r=1}^{R} \prod_{v_j \in \mathscr{C}_r} \mathrm{prob}_{\mathscr{C}_r}(v_j)$$

of a partition, where the "probability distribution" $\mathrm{prob}_{\mathscr{C}_r}(v_j)$ of a cluster $\mathscr{C}_r \subsetneq V$ results from iterating the personalized Page-Rank step (9.2) until convergence. The subsequent optimization of $\mathscr{L}(\mathscr{P})$ from [4] then proceeds using a sequence of three routines analogous to the PLANT, GROW and HARVEST routines of INCRES. While the PLANT and HARVEST strategies from [4] barely differ from the INCRES algorithm, the GROW routine

---

function GROWPCUT(F, W)

$\widehat{F} = F\ \mathrm{diag}(1^T F)^{-1}$
Solve $\quad \left(\mathrm{Id} - \alpha W D^{-1}\right) \widehat{U} = (1-\alpha)\widehat{F} \quad$ for $\widehat{U}$

$\tilde{F}_{ik} = F_{ik}/\widehat{U}_{ik}$
Solve $\quad \left(\mathrm{Id} - \alpha D^{-1} W\right) \tilde{U} = (1-\alpha)\tilde{F} \quad$ for $\tilde{U}$

return $U = \tilde{U} + \log \widehat{U}$.

end function

---

for PCUT requires two label propagation steps performed in series. After normalizing F to make $\widehat{F}$ column-stochastic, the solution $\widehat{U}$ of the first system comes from iterating

$$U \leftarrow \alpha W D^{-1} U + (1-\alpha)\widehat{F} \tag{9.3}$$

until convergence. The solution $\tilde{U}$ of the second system similarly comes from an iterative diffusive process

$$U \leftarrow \alpha D^{-1} W U + (1 - \alpha)\tilde{F}. \tag{9.4}$$

In other words, the GROW function for the PCUT algorithm also performs a type of label propagation on randomly planted seeds, but it does so in quite an expensive way. The dual diffusions result in twice the complexity of INCRES, and moreover, the second PCUT diffusion depends on the first; they cannot be performed in parallel. While performing GROW in this way leads to a rigorous strategy for optimizing the likelihood $\mathscr{L}(\mathscr{P})$, in practice the second diffusion proves unnecessary. At the outset of the algorithm $\log \widehat{U}$ dominates $\tilde{U}$ by an order of magnitude, and approximating the PCUT algorithm by neglecting this term essentially results in INCRES. Making this approximation allows us to perform label propagation in a more efficient way, and as a consequence, we obtain an algorithm that runs more than twice as fast as the PCUT algorithm. The INCRES algorithm has a heuristic dynamic process rather than a rigorous energetic framework as its foundation, but as we show in the experimental section, the two approaches achieve comparable results in terms of accuracy.

The NIBBLE algorithm and its relatives [1, 11, 13, 14] also relate to INCRES in the sense that they obtain unsupervised clusterings from label propagation by planting random seeds. These works cluster the entire graph in a sequential manner: at each step a single random vertex is drawn and propagated. Then a sweep is performed to extract a small cluster around this vertex. These algorithms function well for problems aimed at extracting many small clusters from graphs with fine structure. In contrast, we perform multiway partitioning directly instead of recursively. We also aim at medium scale clusters instead of small scale clusters. We also utilize a significantly different random seeding strategy.

The INCRES algorithm alternates between label propagation (GROW) and thresholding (HARVEST). The idea of iteratively alternating between a few steps of label propagation and subsequent thresholding has also appeared in a transductive learning context [7], although the presence of labeled information results in a different implementation of the propagation step. The non-negative matrix factorization method [16] also incorporates random walk information in a manner that resembles the GROW routine, but otherwise the underlying principles of the algorithms differ substantially.

Finally, the algorithms GRACLUS [6] and METIS [8] directly inspired the multigrid version of our algorithm. We use essentially the same coarsening algorithm, but rely upon a different clustering on the coarsest scale (INCRES vs. kernelized k-means or pure spectral clustering). Our refinement technique also differs substantially. The INCRES algorithm relates to the kernelized k-means procedure used in GRACLUS even in the single level case: we can essentially interpret the GROW routine as the "maximization" step in an alternating minimization for a kernelized k-means. However, the kernel is a power of the normalized weights and the power

may depend on the cluster, so it is not exactly the same. The "expectation" step in our algorithm is replaced by sampling, and instead of having a single representative for a class, the number of representatives increases as the algorithm progresses. Using power iterations of the weight matrix W directly for clustering has appeared in [9, 10]. These works utilize the power iterations to generate an embedding of the vertices of the graph, which is then clustered using k-means. These methods can also be considered as kernelized k-means methods, with a power of the weights providing the kernel.

Because the GROW function we use iterates the random walk on the graph, our algorithm is a form of spectral clustering. However, our main contribution to the clustering problem, and the primary novelty in our algorithm, is the *incremental reseeding process*. This process is not fundamentally tied to the INCRES algorithm presented here—it seems to be quite universal and can be adapted to other clustering methods. However, combining reseeding with the random walk method offers an excellent combination of accuracy, speed, and robustness.

A Multigrid Speedup

The main computational burden of the basic INCRES algorithm lies in the GROW routine. Its computational cost scales like $O(R \times E \times \text{diam}(G))$ in the worst case, where E denotes the number of edges in the graph and $\text{diam}(G)$ denotes its diameter. In practice, the *expected* diameter of the graph effectively determines the cost of each step in the algorithm. For graphs commonly used in machine learning, such as k-nearest neighbor graphs, the number of matrix multiplications required in each call to GROW is generally quite small as a result.

However, the computational burden of the GROW routine still causes the straightforward implementation of our algorithm (in serial) to run two orders of magnitude slower than popular multiscale coarsen-and-refine algorithms such as GRACLUS [6] and METIS [8]. As we now show, pursuing a similar coarsen-and-refine strategy allows us remove an additional order of magnitude from the runtime of our algorithm. In many cases, this multiscale version of INCRES still maintains the consistently high level of accuracy obtained by the basic version.

Coarsening Phase We follow the same coarsening procedure used by GRACLUS and its relatives [6, 8] in our multilevel approach. We construct an agglomerative hierarchy of weighted graphs in a recursive fashion, beginning from the original weighted graph. We therefore set $G^1 := G = (V, W)$ and then successively transform the vertex set $V^1 = V$ into a sequence of smaller weighted graphs $G^2, G^3, \ldots, G^L$ in such a way that the size of each corresponding vertex set decreases $|V^1| > |V^2| > \ldots > |V^L|$ in a geometric fashion. The procedure terminates once the number of vertices $|V^L|$ in the current graph falls below some number n_0, which we take as $n_0 = 20R$ in our experiments.

The transition between two successive levels G^l and G^{l+1} proceeds as follows. Each vertex of G^l begins unmarked. We then visit the vertices in V^l one-by-one according to their degree, from smallest to largest. When visiting a given vertex v_i, we merge it with its neighboring unmarked vertex v_j that maximizes

$$\frac{W_{ij}}{d_i} + \frac{W_{ij}}{d_j}.$$

The two merged vertices v_i and v_j are then marked, and the process continues. If at any stage a vertex v_i has no unmarked neighbors, we simply mark v_i and leave it as a singleton. The vertices v_i^{l+1} in G^{l+1} therefore correspond to pairs of vertices $\{v_{i_1}^l, v_{i_2}^l\}$ from G^l, so that the number of vertices $|V^l|$ roughly halves at each stage. Given two new vertices $v_i^{l+1} = \{v_{i_1}^l, v_{i_2}^l\}$ and $v_j^{l+1} = \{v_{j_1}^l, v_{j_2}^l\}$, we define the weights $W_{i,j}^{(l+1)}$ between these two vertices according to the relations

$$W_{ij}^{(l+1)} := W_{i_1 j_1}^{(l)} + W_{i_1 j_2}^{(l)} + W_{i_2 j_1}^{(l)} + W_{i_2 j_2}^{(l)},$$
$$W_{ii}^{(l+1)} := W_{i_1,i_1}^{(l)} + 2W_{i_1 i_2}^{(l)} + W_{i_2 i_2}^{(l)},$$

i.e. simply by summing the weights between all possible pairs of vertices in the two merged pairs.

Base Clustering and Refinement The clustering phase begins with a base clustering of the coarsest graph G^L in the hierarchy. We simply use the basic version of INCRES applied to G^L to obtain this initial clustering. We then extrapolate this clustering to the next level G^{L-1} in the hierarchy, refine the clustering of G^{L-1} using INCRES again, and repeat until we obtain a clustering of the original graph at the finest level.

The extrapolation procedure we use is straightforward: a clustering of G^{l+1} defines a corresponding clustering of G^l by simply assigning each vertex $v_{i_1}^l$ and $v_{i_2}^l$ in the pair $v_i^{l+1} = \{v_{i_1}^l, v_{i_2}^l\}$ to the cluster of its parent. We subsequently refine this clustering using a slightly modified version the original INCRES procedure. Let $N^l := |V^l|$ denote the number of vertices in the current graph. We set the initial number of seeds to $s_i = .2N^l$, the final number of seeds to $s_f = .5N^l$, and we specify a set number of iterations I^l of INCRES to perform at the current level. We then select the seed increment parameter ds so that the number of seeds s transitions from s_i to s_f in exactly I^l iterations. We perform $I^L = 200$ steps of INCRES at the coarsest level and $I^1 = 1$ step of INCRES at the finest level. We then select the number of iterations I^l to perform at the lth level as a geometrically decreasing progression between these two endpoints.

Experiments

We now provide the results of our extensive experimental evaluation of the algorithm. This section shows that our algorithm achieves state-of-the-art performance in terms of cluster purity on a variety of real word data sets while running faster than the other comparably accurate clustering methods. We also show that INCRES is very robust to perturbations in the input graph.

The Algorithms We compare our method against five clustering algorithms that rely on variety of different principles. We select algorithms that, like our algorithm, partition the graph in a direct, non-recursive manner. The PCut algorithm [4] shares many of the features and motivations of the INCRES algorithm, but has twice the complexity per step. The NCut algorithm [17] is a widely used spectral algorithm that relies on a post-processing of the eigenvectors of the graph Laplacian to optimize the normalized cut energy. The NMFR algorithm [16] uses non-negative matrix factorization and graph-based random walk principles in order to factorize and regularize the original input similarity matrix. The LSD algorithm [2] provides another non-negative matrix factorization algorithm. It aims at finding a left-stochastic decomposition of the similarity matrix. The MTV algorithm from [3] provides a total-variation based algorithm that attempts to find an optimal multiway Cheeger cut of the graph by using ℓ^1 optimization techniques. The last three algorithms (NMFR, LSD and MTV) all use NCut in order to obtain an initial partition. By contrast, we initialize our algorithm with a random partition. We use the code available from [17] for NCut, the code available from [16] to test the two non-negative matrix factorization algorithms (NMFR and LSD) and the code available from [3] for the MTV algorithm.

The Data Sets We provide experimental results on four text data sets (20NEWS, RCV1, WEBKB4, CITESEER) and four data sets containing images of handwritten digits (MNIST, PENDIGITS, USPS, OPTDIGITS). We processed the text data sets by removing a list of stop words as well as by removing all words with fewer than twenty occurrences (for 20NEWS) and fewer than five occurrences (for all others) across the corpus. We then construct a 5-NN graph based on the cosine similarity between tf-idf features. For variety, we include some weighted graphs (RCV1 and CITESEER) as well as some unweighted graphs (20NEWS and WEBKB4). For MNIST, PENDIGITS and OPTDIGITS we use the similarity matrices constructed by [16], where the authors first extract scattering features [5] for images before calculating an unweighted 10-NN graph. For USPS we constructed a weighted 10-NN graph from the raw data without any preprocessing. We provide the source for these data sets and more details on their construction in the supplementary material.

Accuracy Comparisons In Table 9.1 we report the accuracy obtained by the selected algorithms NCUT LSD, NMFR, MTV, PCUT and INCRES (for two values of the timestep parameter, **speed** $= 1$ and **speed** $= 5$) on the various data sets. We use cluster purity to quantify the quality of the calculated partition, defined

Table 9.1 Algorithmic comparison via cluster purity

Data	Size	R	RND	NCUT	LSD	NMFR	MTV	PCUT *(speed 1)*	INCRES *(speed 1)*	INCRES *(speed 5)*
20NEWS	20K	20	6%	27%	34%	**61%**	36%	**61%**	**61%**	**61%**
RCV1	9.6K	4	30%	38%	38%	43%	43%	53%	**55%**	51%
WEBKB4	4.2K	4	39%	40%	46%	**58%**	45%	**58%**	57%	57%
CITESEER	3.3K	6	22%	23%	53%	**63%**	43%	**63%**	62%	62%
MNIST	70K	10	11%	77%	76%	**97%**	96%	**97%**	96%	94%
PENDIGIT	11K	10	12%	80%	86%	87%	87%	87%	**89%**	86%
USPS	9.3K	10	17%	72%	70%	86%	85%	**89%**	88%	87%
OPTDIGIT	5.6K	10	12%	91%	91%	**98%**	95%	**98%**	97%	95%

according to the relation

$$\text{Purity} = \frac{\text{number of "successes"}}{N} = \frac{1}{N}\sum_{r=1}^{R} \max_{1<i<R} n_{r,i}\,.$$

Here $n_{r,i}$ denotes the number of data points in the rth cluster that belong to the ith ground-truth class. In other words, given a computed cluster we count a data point as a success if it belongs to the ground truth class that best represents the cluster. We allowed each iterative algorithm a total of 10,000 iterations to reach convergence. Both INCRES, PCUT and MTV rely on randomization, so for these algorithm we report the average purity achieved over 1000 different runs. The fourth column of the table (RND) provides a base-line purity for reference, i.e. the purity obtained by assigning each data point to a class from 1 to R uniformly at random. The boldface numbers in the table indicate the highest purity score achieved on each data set.

Overall, INCRES, PCUT and NMFR significantly outperform the other algorithms. This is especially true for text data sets. These three algorithms utilize a random walk strategy to help "smooth" irregular graphs, such as the similarity matrices obtained from text data sets. This strategy also contributes to the robustness of these algorithms and to their solid performance across the full range of data sets. However, the INCRES algorithm typically runs at least one order of magnitude faster than the NMFR algorithm and more than twice as fast as the PCUT algorithm. As that INCRES and PCUT obtain very similar purity scores, these results provide evidence of the fact that the sophisticated GROW routine of the PCUT algorithm does not lead to substantially better results than the simple and more efficient GROW routine of the INCRES algorithm. The additional mathematical rigor of PCUT does not translate into better results in practice, and it comes with a non-trivial increase in computational cost.

Finally, note that the INCRES algorithm performs comparably when **speed** $= 1$ and **speed** $= 5$, demonstrating that the algorithm is robust with respect to the choice of the seed increment parameter ds.

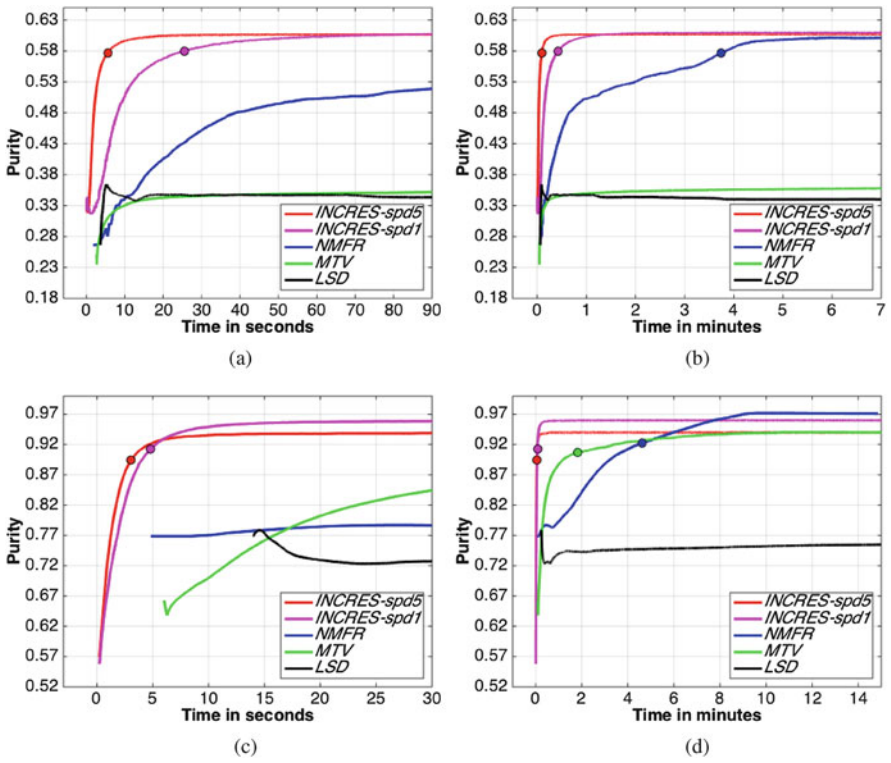


Fig. 9.2 Purity curves for the four algorithms considered on two benchmark data sets (20NEWS and MNIST). We plot purity against time for each algorithm over two different time windows. The circular marks on each curve indicate the point at which the curve reaches 95% of its limiting value. The corresponding times at which this happens are reported in Table 9.2. (**a**) 20NEWS (first 90 s). (**b**) 20NEWS (first 7 min). (**c**) MNIST (first 30 s). (**d**) MNIST (first 15 min)

Speed Comparisons Figure 9.2 illustrates the speed at which, LSD, MTV, NMFR and INCRES converge toward their respective solutions. We ran each algorithm for a total of 7 min on 20NEWS and for 15 min on MNIST. We report the purity obtained by the algorithm at each iteration. For the randomized algorithm (INCRES and MTV) the purity curves were obtained by averaging the results over 240 runs. The overwhelming computational burden for all of these algorithms arises from the sparse-matrix times full-matrix multiplications required at each step. The PCUT algorithm (not shown) runs about two times slower than the INCRES algorithm but otherwise achieves similar results. Each algorithm is implemented in a fair and consistent way, and the experiments were all performed on the same architecture.

In order to give an indication of the speed/accuracy trade-off for each algorithm, in Table 9.2 we record the time it took for the purity obtained by each algorithm to reach 95% of its limiting value on both 20NEWS and on MNIST. Overall, the simple INCRES algorithm provides accuracy comparable to the state-of-the-art NMF algorithm [16], yet runs an order of magnitude faster. Timing results on

Table 9.2 Computational time

Data	NMFR	MTV	INC. *(spd1)*	INC. *(spd5)*
20NEWS	3.7 min	–	25.4 s	5.6 s
	(57.7%)	–	(57.7%)	(58%)
MNIST	4.6 min	1.8 min	4.8 s	3.1 s
	(92.2%)	(90.7%)	(91.2%)	(89.3%)

Table 9.3 Robustness comparisons

Noise	NCUT	LSD	MTV	GRACLUS *(multilevel)*	INCRES-ml *(multilevel)*	INCRES *(speed1)*
20NEWS						
+0% edges	27%	34%	36%	42%	59%	**61%**
+50% edges	21%	27%	20%	15%	19%	**52%**
+100% edges	18%	22%	11%	13%	12%	**44%**
+150% edges	15%	20%	10%	12%	10%	**34%**
+200% edges	14%	18%	9%	11%	9%	**27%**
MNIST						
+0% edges	77%	76%	96%	**97%**	**97%**	96%
+50% edges	87%	94%	55%	93%	**97%**	**97%**
+100% edges	84%	93%	25%	80%	90%	**97%**
+150% edges	74%	87%	18%	63%	74%	**97%**
+200% edges	67%	82%	16%	52%	53%	**96%**

Bold values mean highest values—this is standard to use in this field

the data sets from Table 9.1 are consistent with those obtained for 20NEWS and MNIST, in the sense that INCRES typically runs one order of magnitude faster than NMFR on these data sets as well.

Robustness Experiments Table 9.3 reports accuracy results of various algorithms on graphs that we corrupted by adding different levels of noise. We began with the original 20NEWS graph used in Table 9.1 and added additional edges to the graph uniformly at random. The original graph had $e = 144,632$ edges. For the experiment, we added $0.5e$, e, $1.5e$ and $2e$ additional noise edges. For each of these four levels of noise, we randomly generated 144 separate perturbed graphs. The table reports, for each level of noise, the average purity obtained by each algorithm on the 144 randomly generated matrices. We then proceeded to perturb the original MNIST graph in a similar fashion. The original graph has $e = 1,027,412$ edges, and we randomly generated 120 graphs at each level of noise. This gives a total of 1056 randomly generated graphs for this set of experiments. We provide experimental results for all algorithms other than NMFR (it is far too slow to run to convergence on all 1056 adjacency matrices) and PCUT (to avoid redundancy).

The results clearly elucidate the robustness of the INCRES algorithm with respect to noise in the graph construction process. On the 20NEWS data set, for example, all other algorithms experience a sharp decrease in accuracy as soon as noise is added. In contrast, the purity of the INCRES algorithm slowly decreases in

Table 9.4 Accuracy comparison for multilevel algorithms

Data	Size	METIS	GRACLUS		INCRES-ml	
20NEWS	20K	42.4%	42.4%	(0.05s)	**58.6%**	(0.85s/0.07s)
RCV1	9.6K	34.1%	42.4%	(0.01s)	**47.9%**	(0.15s/0.03s)
WEBKB4	4.2K	37.9%	49.0%	(0.01s)	**53.9%**	(0.10s/0.02s)
CITESEER	3.3K	45.2%	53.5%	(0.01s)	**61.5%**	(0.11s/0.02s)
MNIST	70K	86.0%	**96.9%**	(0.17s)	**96.9%**	(1.99s/0.52s)
PENDIGIT	11K	67.3%	84.7%	(0.02s)	**88.8%**	(0.46s/0.05s)
USPS	9.3K	75.1%	86.9%	(0.02s)	**87.2%**	(0.34s/0.05s)
OPTIDIGIT	5.6K	83.0%	94.2%	(0.01s)	**95.0%**	(0.26s/0.03s)

Bold values mean highest values—this is standard to use in this field

a stable fashion. On the MNIST data sets, the results obtained by INCRES remain essentially unchanged across all noise levels. The competing algorithms do not exhibit this behavior. Interestingly, NCut and LSD actually obtain *better* results at the 50% and 100% noise levels. Given that LSD relies on NCut for initialization, it comes as no surprise that gains for NCut produce subsequent gains for LSD as well. This pathological behavior still indicates a lack of robustness, in the sense that both algorithms exhibit a high degree of sensitivity to changes in the underlying graph.

Multigrid Experiments Table 9.4 reports the accuracies and run times of the coarsen and refine algorithms METIS [8], GRACLUS [6] and the multilevel version of our reseeding algorithm. We refer to the multilevel version of INCRES as INCRES-ml to distinguish it from the basic version. The reported purity values correspond to the average accuracy obtained over 500 trials of each routine. By and large, INCRES-ml obtains clustering of higher quality than the two other multilevel algorithms. This comes at a cost of one order of magnitude in computational time.

The computational cost that INCRES-ml incurs at level l of the graph hierarchy scales as

$$O(R \times E_l \times \text{diam}(G_l) \times I^l),$$

where E_l denotes the number of edges in the graph G^l and I^l once again denotes the number of iterations performed at this level. In comparison, the computational burden that GRACLUS incurs at level l scales as

$$O(R \times E_l \times I^l),$$

where I^l has the same meaning as before, but obviously refers to the number of iterations performed by GRACLUS. The extra term $\text{diam}(G_l)$ in the computational cost of INCRES-ml partly explains the difference between the computational times. However, the diameters of the graphs considered in our experiments are typically smaller than ten. This is especially true for the coarsest levels in the hierarchy. The other major source of the difference in computational time comes from implementation: while GRACLUS is a heavily optimized C code, our multilevel

algorithm has a naive MATLAB implementation. Finally, we remark that all of these multilevel procedures suffer from a severe sensitivity to noise, see Table 9.3. This fact motivates the use of the basic version of INCRES in situations where robustness to the graph construction process is highly desirable.

Acknowledgements XB is supported by NRF Fellowship NRFF2017-10.

Appendix

Matlab Code Used in the Experimental Section

Below is the exact MATLAB code that we used in the experimental section of the paper.

```
function C=INCRES( W , R , speed , maxiter)
D = sum(W,2);
RW=W*spdiags(1./D,0,n,n);
s=1;
ds=speed*10^(-4)*n/R;
C=randi(R,n,1);
for iter=1:maxiter
    [F,R] = PLANT(C, R, round(s));
    while ( min( min(F) ) < eps )
        F= RW * F;
    end
    [~,C] = max(F,[],2);
    s=s+ds;
end
end

function [F,R] = PLANT(C,R,s)
n=length(C);
F=zeros(n,R);
EmptyClass=[];
for k=1:R
    idx= find(C==k);
    ClassSize=length(idx);
    if (ClassSize== 0)
        EmptyClass=[EmptyClass,k];
    else
        idxSeeds = idx( randi(ClassSize,s,1) );
        F(:,k)= full(sparse(idxSeeds,ones(1,s),1,n,1));
    end
```

```
end
F(:,EmptyClass)=[];
R=size(F,2);
end
```

Datasets

- 20NEWS (unweighted similarity matrix): The word count from the raw documents was computed using the Rainbow library [12] with a default list of stop words. Words appearing less than 20 times were also removed. The similarity matrix was then obtained by 5 nearest neighbors using cosine similarity between tf-idf features. Source: http://www.cs.cmu.edu/~mccallum/bow/rainbow/
- RCV1 (weighted similarity matrix): This dataset was obtained in preprocessed format from http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html with the tf-idf features were already computed. We then simply used cosine similarity and 5-NN.
- WEBKB4 (unweighted similarity matrix): The word count from the raw documents was done with the Rainbow library [12]. A list of stop word was removed. Words appearing less than five times were removed. The similarity matrix was then obtained by five nearest neighbors using cosine similarity between tf-idf features. Source: http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/
- CITESEER (weighted similarity matrix): This dataset was obtained in preprocessed format from http://linqs.cs.umd.edu/projects//projects/lbc/index.html where each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. We then simply used cosine similarity and 5-NN.
- MNIST, PENDIGITS, OPTDIGITS (unweighted similarity matrix): The similarity matrices were obtained from [16], where the authors first extracted scattering features using [15] for images before calculating the 10-NN graph. Source: http://users.ics.aalto.fi/rozyang/nmfr/index.shtml
- USPS (weighted similarity matrix): We computed a 10-NN graph using standard Euclidean distance between the raw images. Each edge in the 10-NN graph was given the weight

$$w_{ij} = \mathrm{e}^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

where each $\mathbf{x}_i$ denotes a vector containing the raw pixel data. The parameter σ was chosen as the mean distance between each vertex and its 10^{th} nearest neighbor. Source: http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

References

1. R. Andersen, F. Chung, K. Lang, Local graph partitioning using pagerank vectors, in *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS '06)*, pp. 475–486 (2006)
2. R. Arora, M. Gupta, A. Kapila, M. Fazel, Clustering by left-stochastic matrix factorization, in *International Conference on Machine Learning (ICML)* (2011), pp. 761–768
3. X. Bresson, T. Laurent, D. Uminsky, J. von Brecht, Multiclass total variation clustering, in *Advances in Neural Information Processing Systems (NIPS)* (2013)
4. X. Bresson, T. Laurent, A. Szlam, J.H. von Brecht, The product cut, in *Advances in Neural Information Processing Systems (NIPS)* (2016)
5. J. Bruna, S. Mallat, Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1872–1886 (2013)
6. I.S. Dhillon, Y. Guan, B. Kulis, Weighted graph cuts without eigenvectors: a multilevel approach. IEEE Trans. Pattern Anal. Mach. Intell. **29**(11), 1944–1957 (2007)
7. C. Garcia-Cardona, E. Merkurjev, A.L. Bertozzi, A. Flenner, A.G. Percus, Multiclass data segmentation using diffuse interface methods on graphs. IEEE Trans. Pattern Anal. Mach. Intell. **99**, 1 (2014)
8. G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359–392 (1998)
9. S. Lafon, A.B. Lee, Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. IEEE Trans. Pattern Anal. Mach. Intell. **28**(9), 1393–1403 (2006)
10. F. Lin, W.W. Cohen, Power iteration clustering, in *ICML* (2010), pp. 655–662
11. L. Lovász, M. Simonovits, Random walks in a convex body and an improved volume algorithm. Random Struct. Algorithms **4**(4), 359–412 (1993)
12. A.K. McCallum, Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering (1996). http://www.cs.cmu.edu/~mccallum/bow
13. D.A. Spielman, S.-H. Teng, Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems, in *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing* (2004), pp. 81–90
14. D.A. Spielman, S.-H. Teng, A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. SIAM J. Comput. **42**(1), 1–26 (2013)
15. M. Stephane, Group invariant scattering. Commun. Pure Appl. Math. **65**(10), 1331–1398 (2012)
16. Z. Yang, T. Hao, O. Dikmen, X. Chen, E. Oja, Clustering by nonnegative matrix factorization using graph random walk, in *Advances in Neural Information Processing Systems (NIPS)* (2012), pp. 1088–1096
17. S.X. Yu, J. Shi, Multiclass spectral clustering. in international conference on computer vision, in *International Conference on Computer Vision* (2003)
18. X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions. in *IN ICML*, pp. 912–919 (2003), pp. 912–919

Chapter 10
Ego-Motion Classification for Body-Worn Videos

Zhaoyi Meng, Javier Sánchez, Jean-Michel Morel, Andrea L. Bertozzi, and P. Jeffrey Brantingham

Abstract Portable cameras record dynamic first-person video footage and these videos contain information on the motion of the individual to whom the camera is mounted, defined as ego. We address the task of discovering ego-motion from the video itself, without other external calibration information. We investigate the use of similarity transformations between successive video frames to extract signals reflecting ego-motions and their frequencies. We use novel graph-based unsupervised and semi-supervised learning algorithms to segment the video frames into different ego-motion categories. Our results show very accurate results on both choreographed test videos and ego-motion videos provided by the Los Angeles Police Department.

Introduction

Affordable high-quality cameras for recording the first-person point-of-view experience, such as GoPro, are an increasingly common item in many aspects of people's lives. In this paper, we present a novel approach for segmenting or indexing body-worn videos to different ego-motion categories.

Prior work on vision-based first-person human action analysis has focused a lot on indoor activities, such as object recognition [24], hand gesture recognition [18] [32], sign language recognition [29], context aware gesture recognition [28], hand tracking [31] and detecting daily life activities [23]. Work with body-worn

Z. Meng · A. L. Bertozzi (✉) · P. J. Brantingham
University of California, Los Angeles, Los Angeles, CA, USA
e-mail: mzhy@ucla.edu; bertozzi@ucla.edu; branting@ucla.edu

J. Sánchez
Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain
e-mail: jsanchez@ulpgc.es

J.-M. Morel
Ecole Normale Supérieure de Cachan, Cachan, France

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_10

sensors has also been shown to be effective for categorizing human actions and activities [11] [27]. An unsupervised ego-action learning method was proposed in [14] for sports videos. The basis of video indexing is to model the transformation between successive frames in the video. For the purpose of video indexing, several studies have examined parametric models of frame transformation [6, 13]. Parametric models can be also used for video stabilization [26], and panorama construction [12].

In this paper, we propose an approach to classify different ego-motion categories. We know that human motion observed from a first-person point-of-view can be captured by the global displacement between successive frames. This means that we should be able to aggregate global motion and marginalize out local outlier motion. We also know that motion involving the human gait has an inherent frequency component. Therefore we can expect that frequency analysis can be used as an important feature for ego-action categorization. We propose the use of a parametric model for calculating the simple global representation of motion. This approach produces a low dimensional representation of the motion of the ego. We then classify the ego-motion using novel graph-based semi-supervised and unsupervised learning algorithms. The algorithms are motivated by PDE-based image segmentation methods and achieve high performance in both accuracy and efficiency for different discrete data sets.

We consider the ego-motion classification problem with both benchmark and real-world data. Working with both types of data is critical because of the stark differences in the degree of difficulty in the analysis of video data collected under controlled and uncontrolled or "wild" conditions. Benchmark datasets with known ground truth are developed under experimental conditions controlled by the researcher. Such datasets attempt to simulate the types of behaviors that are of most interest. Simulations may favor positive outcomes because they seek not only to limit sources of error linked to video image quality, but also enhance target behaviors of interest. For example, experimental protocols that seek to enhance camera stability, ensure adequate lighting conditions, avoid obstructions may all assist in the algorithmic task. Ensuring that experimental participants enact well-defined or discrete transitions between different types of behavior, or exaggerate the differences between behavioral modes may favor accurate segmentation. We draw on choreographed video collected under controlled circumstances to develop our approach.

Videos not collected under controlled conditions may nevertheless be hand-labeled by the researcher to produce a ground truth. Such videos may be subject to many more quality challenges than simulated scenes. Actual behavior and conditions as they exist on the ground are unforgiving. People in real-world settings may not act in discrete, linear sequences, nor are they necessarily inclined to exaggerate their different actions for easy detection. Ego-motions may also proceed so quickly that they defy discrete recognition. We may also lack sufficient semantic categories to capture the diversity of real-world behavior. Real-world video systems may also not be state-of-the-art and therefore suffer from poor camera stability, low frame rate, low resolution, poor color saturation and data collection errors (both

human and mechanical). All of these effects can drastically impact the ability of the researcher to label video for ground truth, which introduces errors into algorithmic methods. We draw on police body-worn video (BWV) to evaluate how our methods perform under challenging real-world conditions. Police BWV is typically shaky, contains noise from low light conditions, poor color saturation and occlusions, and represents diverse and often mixed motion routines.

The paper is organized as follows. In section "Motion Features", we describe the method for motion feature extraction for two successive frames. In section "Classification Method", we investigate the semi-supervised and unsupervised graph-based MBO algorithms for classification. In section "Experimental Results", we elaborate on our experimental results using choreographed test video and real-world video data. Section "Conclusion" concludes the paper.

Motion Features

We characterize motion in a video sequence using a set of features. The features represent the relative movements of ego, the individual on whom the video camera is mounted. The features depend on the estimation of parametric models between successive frames and on the analysis of periodic signals of the motion through characteristic frequencies. We illustrate our method of constructing the motion features in Fig. 10.1. In section "Transformations Between Two Successive Frames", we discuss how to use the inverse compositional algorithm to estimate the similarity transformation between successive frames. This transformation is represented by four parameters t_x, t_y, a and b. In section "Movement Signal", we construct four of the features to be used for the video segmentation—horizontal displacement (x), vertical displacement (y), angle of rotation (r) and zoom (z) using the similarity transformation. In addition, the characteristic frequencies of these four signals are computed using the method discussed in section "Frequency Signal". In section "Equalization of Variance", we combine the four movement features and four frequency features to obtain the eight-dimensional feature vector

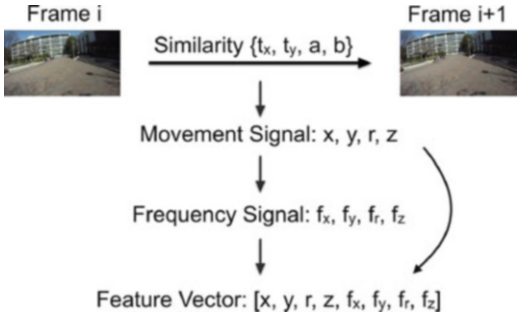


Fig. 10.1 The process of constructing the motion features for each two successive frames

for each transformation between two successive frames. It is this feature vector that will be used for the graph-based machine learning method.

Transformations Between Two Successive Frames

To compute the motion of the video sequence, we estimate the similarity transformations between consecutive frames using the inverse compositional algorithm [1, 2]. It is possible to use more general parametric motions, such as affinities or homographies. However, their calculation is more prone to errors in the presence of camera shake. In any case, we find that the four parameters of the similarity are sufficient to characterize motion.

The inverse compositional algorithm is an improvement of the Lucas-Kanade method [2, 15] for image registration. Its implementation in [25] includes the use of robust error functions and estimates the correct transformation even in the presence of occlusions or multiple motions. Let $\mathbf{I}_1(\mathbf{x})$ and $\mathbf{I}_2(\mathbf{x})$ be two images, with $\mathbf{x} = (x, y)$. Let $\mathbf{p}$ be the global displacement vector between the two images and $\Delta\mathbf{p}$ be the incremental displacement vector at each iteration. Let $\mathbf{x}'(\mathbf{x}; \mathbf{p}, \Delta\mathbf{p})$ be the correspondence map from the left to the right image, or equivalently two frames in a video sequence, parameterized by $\mathbf{p}$ and the incremental refinement $\Delta\mathbf{p}$. The energy model is given by

$$E(\Delta\mathbf{p}) = \sum_{\mathbf{x}} \rho\left(\left|\mathbf{I}_2(\mathbf{x}'(\mathbf{x};\mathbf{p})) - \mathbf{I}_1(\mathbf{x}'(\mathbf{x};\Delta\mathbf{p}))\right|_2^2; \lambda\right), \tag{10.1}$$

where $\rho(\cdot)$ is a function that gives less weight to large values of the argument, where the difference in image intensities is big (e.g., $\rho(s^2, \lambda) = 0.5s^2/(s^2 + \lambda^2)$).

Minimizing the energy with respect to $\Delta\mathbf{p}$ yields:

$$\Delta\mathbf{p} = H_\delta^{-1} \sum_{\mathbf{x}} \rho' \cdot (\nabla\mathbf{I}_1(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \left(\mathbf{I}_2(\mathbf{x}'(\mathbf{x};\mathbf{p})) - \mathbf{I}_1(\mathbf{x})\right), \tag{10.2}$$

with

$$\begin{aligned} H_\delta &= \sum_{\mathbf{x}} \rho' \cdot (\nabla\mathbf{I}_1(\mathbf{x})\mathbf{J}(\mathbf{x}))^T \nabla\mathbf{I}_1(\mathbf{x})\mathbf{J}(\mathbf{x}) \\ &= \begin{pmatrix} \sum_{\mathbf{x}} \rho' \cdot \left(\mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x})\right)^T \mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x}) & \sum_{\mathbf{x}} \rho' \cdot \left(\mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x})\right)^T \mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x}) \\ \sum_{\mathbf{x}} \rho' \cdot \left(\mathbf{I}_{1,x}(\mathbf{x})\mathbf{J}(\mathbf{x})\right)^T \mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x}) & \sum_{\mathbf{x}} \rho' \cdot \left(\mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x})\right)^T \mathbf{I}_{1,y}(\mathbf{x})\mathbf{J}(\mathbf{x}) \end{pmatrix}, \end{aligned} \tag{10.3}$$

Table 10.1 Similarity transformation and its Jacobian

Transform	Parameters—$\mathbf{p}$	Matrix—$\mathbf{H}(\mathbf{p})$	Jacobian—$\mathbf{J}(\mathbf{x};\mathbf{p})$
Similarity	(t_x, t_y, a, b)	$\begin{pmatrix} 1+a & -b & t_x \\ b & 1+a & t_y \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & x & -y \\ 0 & 1 & y & x \end{pmatrix}$

and $\rho' := \rho'\left(\left|\mathbf{I}_2(\mathbf{x}'(\mathbf{x};\mathbf{p})) - \mathbf{I}_1(\mathbf{x})\right|_2^2; \lambda\right)$. $\mathbf{J}(\mathbf{x};\mathbf{p}) = \frac{\partial \mathbf{x}'(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}}$ is the Jacobian of the transformation. Table 10.1 lists the similarity transformation and its Jacobian using the parametrization proposed in [33].

The minimum of this energy provides the parameters of the transformation. To reach a highly accurate solution, the algorithm uses an iterative process. It also includes a coarse-to-fine strategy for estimating large displacements. See [25] for further details.

Movement Signal

Simple motions, such as horizontal (x) and vertical (y) movements, zoom (z) and rotation (r) information can be computed given the similarity. The procedure for calculating the displacement of the central pixel is shown in Algorithm 10.1.

Since the similarity includes the composition of a zoom and rotation matrices, it is easy to obtain these coefficients from the parametrization of Table 10.1. In this case, the rotation r and zoom factor z are calculated as

$$r = \arctan\left(\frac{b}{1+a}\right), \quad z = \sqrt{(1+a)^2 + b^2}. \tag{10.4}$$

The signals from raw video footage may have abnormally large values. We filter out these values in preprocessing. We replace the signal value by its mean μ, if the signal value is outside the $(\mu-3\sigma, \mu+3\sigma)$ region where σ is the standard derivation.

Algorithm 10.1: Calculate the displacement of the central pixel

Input : The similarity $\mathbf{H}$, size of the frame n_x and n_y
Output: x,y
1: $\mathbf{p_m} \leftarrow (n_x/2, n_y/2, 1)^T$ {the center of the frame}
2: $(p_1, p_2, p_3)^T \leftarrow \mathbf{H} \cdot \mathbf{p_m}$ {project the center point using the similarity }
3: $(p_1, p_2, p_3)^T \leftarrow (p_1, p_2, p_3)^T / p_3$ {normalize by the third component}
4: $x \leftarrow p_1 - n_x/2$ {the horizontal movement}
5: $y \leftarrow p_2 - n_y/2$ {the vertical movement}
6: **return** x, y

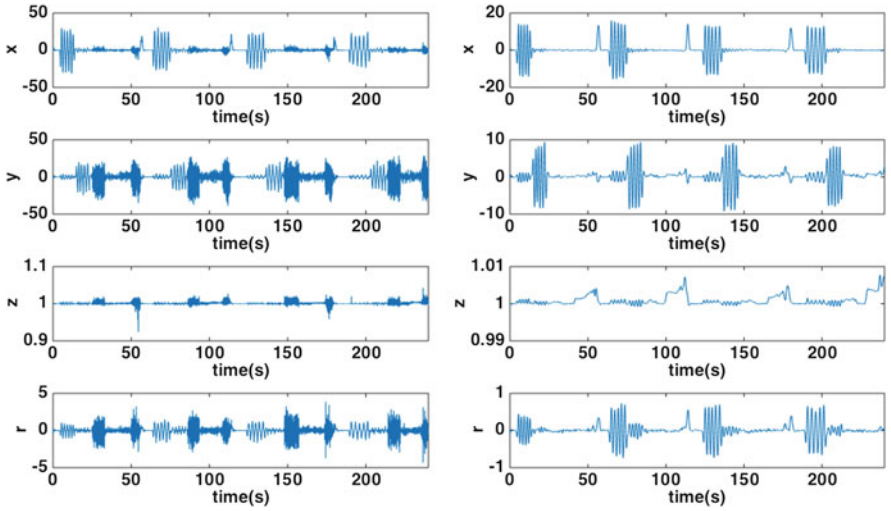


Fig. 10.2 The x, y, r and z signals. On the left, the original signals and, on the right, the corresponding filtered and smoothed data

The filtered signals can still be very noisy. We use convolutions with a Gaussian function to smooth these signals, which is the basic idea in video stabilization [26].

We use the QUAD video data set[1] to examine ego-motion signals. We discuss the details of this data set in section "Experimental Results".

The motion signals we calculate using Algorithm 10.1 and Eq. (10.4) are shown in Fig. 10.2. The left column gives the raw data x, y, z and r and the right column the corresponding filtered and smoothed data.

The periodic pattern correlates with the periodic actions in the QUAD video. The large oscillation of x corresponds to ego turning left and right repeatedly. The large oscillation of y corresponds to ego repeatedly looking up and looking down. The four peaks in z correspond to ego walking and running, since the frames zoom fast when the person is walking or running. The large oscillations of rotation r also correlate with the movements of turning left, turning right, looking up and looking down.

Frequency Signal

Some ego-motions are periodic, such as jumping, walking and running. Periodic motions have different characteristic frequencies. This observation leads us to investigate the frequencies of x, y, z and r using Fourier analysis. We use the

[1]The data set can be found at: http://www.cs.cmu.edu/$\sim$kkitani/datasets/.

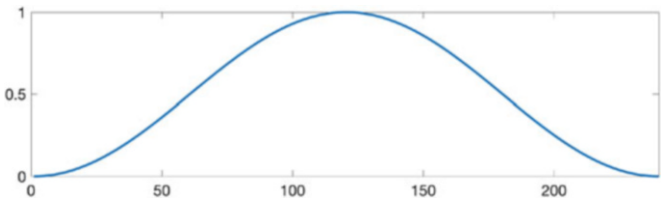


Fig. 10.3 The Hann window

short-time Fourier transform (STFT) to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. In practice, the procedure for computing STFTs is to use a sliding window of fixed length and compute the Fourier transform as the window slides over the whole signal. We use the Hann window here:

$$w(n) = 0.5\left(1 - cos(\frac{2\pi n}{N-1})\right). \tag{10.5}$$

As shown in Fig. 10.3, the Hann window is zero at the boundaries which reduces the artifacts at the boundary. The STFT is defined by:

$$STFTx[n](m,\omega) = X(m,\omega) = \sum_{n=0}^{N} x[n]w[n-m]e^{-j\omega n}, \tag{10.6}$$

where the length of the window is N and m indicates the window sampling rate.

The magnitude squared of the STFT yields the spectrogram of the function:

$$spectrogram\{x[n]\}(m,\omega) = |X(m,\omega)|^2. \tag{10.7}$$

We use a five-second window in our experiments. We show the spectrogram of six different motions of the y signal in Fig. 10.4. The frequency is very small when the ego repeatedly turns left and right. The 2 s period is almost the same as when ego repeatedly looks up and down. Looking up and down causes a frequency at 0.6 Hz. The spectrogram of small steps and walking are very similar. The largest frequency is at 7.8 Hz. When ego walks at 0.5 s per step, the frequency is 2 Hz. However, because the GoPro camera is head-mounted, the camera also has an oscillation when ego is walking. This camera oscillation causes these observed high frequencies. For jumping and running, the spectrogram gives accurate frequencies at 2 Hz and 3.4 Hz, respectively.

We select the characteristic frequency of the window, which is defined as:

$$f_w = \begin{cases} f_{max}, & \text{if } f_{max} > 3\delta \\ 0, & \text{otherwise} \end{cases}, \tag{10.8}$$

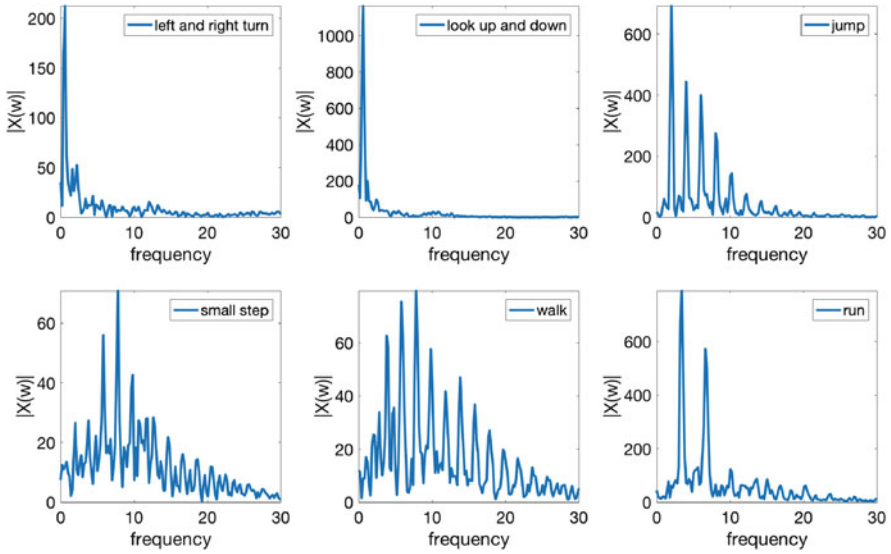


Fig. 10.4 Spectrogram of six kinds of motions

where f_{max} is the frequency corresponding to the largest value in the spectrogram and δ is the standard deviation of the spectrogram. The condition of being larger than 3δ guarantees that the frequency picked is unlikely to be caused by noise.

In practice, we choose N to be 300 frames (5 s) and let the window move 60 frames (1 s) each time. In this case, at each frame, there are 5 f_ws. We choose the median of these f_ws to be the final frequency at the frame.

We apply this procedure with the four movement signals x, y, r, and z and get four frequency signals f_x, f_y, f_r and f_z. In other words, in addition to four movement signals, each frame transition is also associated with four characteristic frequencies. We compute these frequencies for the QUAD video and show their values in Fig. 10.5. We can observe four periods in the frequencies which correlate with the action periods in the video.

Equalization of Variance

We always force the variance of each signal to be 1 by forcing x to be $\bar{x} + \frac{x-\bar{x}}{\sigma(x)}$, where $\bar{x}$ is the mean and $\sigma(x)$ is the standard variation. In this way, each signal gives equal contribution to the combined feature vector. Different weights can be considered to be applied on different signals based on the importance of the signals.

After equalizing the variance of the 8 signals, we combine them into a final motion feature f_{motion}. It is an $N \times 8$ matrix, where N is the number of frames in the video. Each row represents the eight-dimensional feature vector of one frame

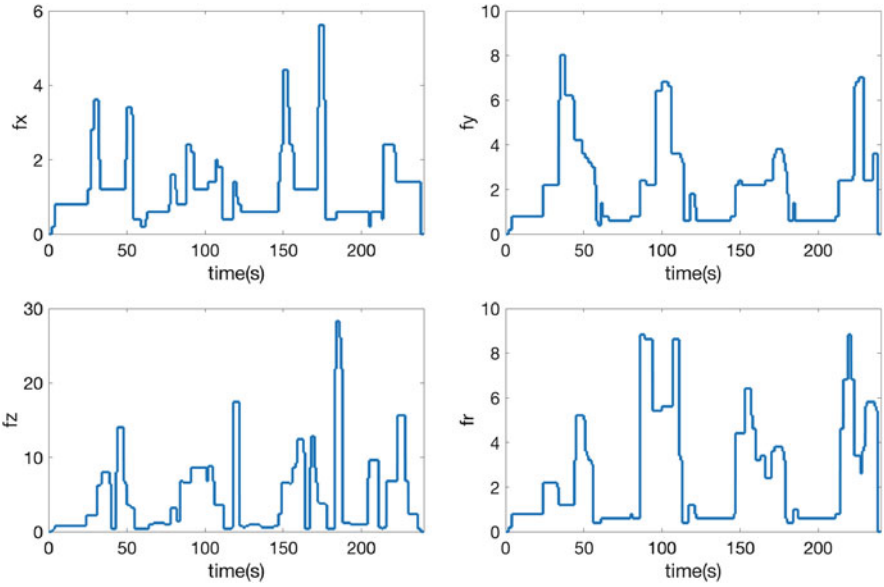


Fig. 10.5 The four characteristic frequencies fx, fy, fz, fr of the QUAD video

and we denote the feature vector of the ith frame to be F_i. In this way, we code the video frames by their feature matrix f_{motion}:

$$f_{motion} = [x, y, r, z, f_x, f_y, f_r, f_z]. \tag{10.9}$$

Classification Method

Once we have built the features f_{motion} of the video, we would like to infer a number of ego-motion categories from the data. In this section, we explore graph-based semi-supervised and unsupervised algorithms for video segmentation. We consider each transformation between two successive frames as a node in a weighted graph and classify them in different motion classes.

Recently, novel classification algorithms have been proposed [20] that are motivated by PDE-based image segmentation methods and are modified to apply to discrete data sets. These algorithms improve both accuracy of the solution and efficiency of the computation and can be potentially faster in parallel than various classification algorithms such as spectral clustering with K-means [17, 36]. The OpenMP parallelization and optimization of the algorithms are discussed in [19] with online demo and codes.

The classification algorithms consider each data point as a node in a weighted graph. The similarity (weight) between two nodes i and j is given by formula:

$$w_{ij} = exp(-||F_i - F_j||_2^2/\tau), \tag{10.10}$$

where F_i and F_j are feature vectors of nodes i and j according to (10.9), and τ is a parameter to be determined [7, 36]. We use the Euclidean distance here. To determine the value of τ, we try different values and run the experiments on the validation data to choose the τ with the best accuracy. We use $\tau = 40$ in this paper. More about how to choose τ can be found in [4].

The classification problem is approached using ideas from graph-cuts [30]. Given a weighted undirected graph, the goal is to find the minimum cut (measured by a summation of the weights along the graph cut) for this problem. This is equivalent to assigning a scalar or vector value u_i to each ith data point and minimizing the graph total variation (TV) $\sum_{ij} |u_i - u_j| w_{ij}$ [34]. Instead of directly solving a graph-TV minimization problem, the graph TV can be transformed to a graph-based Ginzburg-Laudau (GL) functional [4]:

$$E(u) = \epsilon < L_s u, u > + \frac{1}{\epsilon} \sum_i (W(u_i)), \tag{10.11}$$

where $W(u)$ is a double well potential, for example $W(u) = \frac{1}{4}(u^2 - 1)^2$ in a binary partitioning and multi-well potential in k dimensions (same as the number of classes). L_s is the normalized symmetric graph Laplacian which is defined as $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where D is a diagonal matrix with diagonal elements $d_i = \sum_{j \in V} w(i, j)$.

In the vanishing ϵ limit, a variant of the GL energy Gamma-converges to the graph TV functional [35]. Different fidelity terms are added to the GL functional for semi-supervised and unsupervised learning respectively. The GL energy for semi-supervised learning is:

$$E(u) = \epsilon \langle L_s u, u \rangle + \frac{1}{\epsilon} \sum_i W(u_i) + \sum_i \frac{\mu}{2} \lambda(x_i) ||u_i - \widehat{u_i}||_{L_2}^2. \tag{10.12}$$

The last term of Eq. (10.12) is the regular L_2 fit to known data with some constant μ, while $\lambda(x)$ takes the value of 1 on fidelity nodes, and 0 otherwise. The variable $\widehat{u}$ is the initial value for u with randomly chosen labels for non-fidelity data points and the "ground truth" for the fidelity points.

The GL energy for unsupervised learning is:

$$E(u, c_r) = \epsilon \langle L_s u, u \rangle + \frac{1}{\epsilon} \sum_i W(u_i) + \mu \sum_{r=1}^{\widehat{n}} \langle ||f - c_r||^2, u_{\star,r} \rangle. \tag{10.13}$$

In (10.13), the term $||f - c_r||^2$ denotes an $N \times 1$ vector $(||f(x_1) - c_r||^2, \ldots, ||f(x_N) - c_r||^2)^T$ and the x_i $(i = 1, \ldots N)$ are the N pixels of the data set. In addition, the term $u_{\star,r}$ indicates the rth column of u; the vector $u_{\star,r}$ is a $N \times 1$ vector which contains the probabilities of every node belonging to class r. The term $\widehat{n}$ is the number of classes and is to be provided to the algorithm in advance. This problem is essentially equivalent to the K-means method when μ approaches $+\infty$.

The GL functional is minimized using gradient descent [16]. An alternative is to directly minimize the GL functional using the MBO scheme [9, 21], or a direct compressed sensing method [22]. We use the multiclass MBO scheme [9] in this paper in which one alternates between solving the heat (diffusion) equation for u and thresholding to maintain distinct class structure. Computation of the entire graph Laplacian is prohibitive for large data sets so we use the Nyström extension to randomly sample the graph and compute a modest number of leading eigenvalues and eigenfunctions of the graph Laplacian [8]. By projecting all vectors onto this sub-eigenspace, the iteration step reduces to a simple coefficient update.

Semi-supervised and Unsupervised Algorithms

We outline here the semi-supervised and the unsupervised algorithms. For the semi-supervised algorithm, the fidelity data (a small amount of "ground truth") is known and the remaining data needs to be classified according to the categories of the fidelity. For the unsupervised algorithm, there is no prior knowledge of the data labels. We use the Nyström extension algorithm beforehand for both algorithms to calculate the eigenvalues and eigenvectors as the inputs. In practice, these two algorithms converge very fast and give accurate classification results.

Algorithm 10.2: Semi-supervised Graph MBO algorithm [21]

Data: Eigenvectors matrix Φ, eigenvalues $\{\lambda_k\}_{k=1}^M$ and fidelity.
Result: u
Initialize u^0, $d^0 = \mathbf{0}$, $a^0 = \Phi^T \cdot u_0$;
while $\frac{||u^{n+1}-u^n||_2^2}{||u^{n+1}||_2^2} < \alpha = 0.0000001$ **do**
  a. Heat equation;
    1). $a_k^{n+1} = a_k^n \cdot (1 - dt \cdot \lambda_k) - dt \cdot d_k^n$;
    2). $y = \Phi \cdot a^{n+1}$;
    3). $d^{n+1} = \Phi^T \cdot \mu(y - u^0)$,;
  b. Thresholding;
    $u_i^{n+1} = e_r, r = \arg\max_j y_i$;
  c. Updating a;
    $a^{n+1} = \Phi^T \cdot u^{n+1}$

Algorithm 10.3: Unsupervised graph MBO algorithm [10]

Data: data matrix f, eigenvector matrix Φ, eigenvalues $\{\lambda_k\}_{k=1}^{N}$
Result: u
Initialize u^0, $a^0 = \Phi^T \cdot u^0$;
while $\frac{||u^{n+1}-u^n||_2^2}{||u^{n+1}||_2^2} < \alpha = 0.0000001$ **do**
a. Updating c;
$c_k^{n+1} = \frac{<f, u_k^{n+1}>}{\sum_{i=1}^{N} u_{ki}}$;
b. Heat equation;
1. $a_k^{n+\frac{1}{2}} = a_k^n \cdot (1 - dt \cdot \lambda_k)$;
2. Calculating matrix P, where $P_{i,j} = ||f_i - c_j||_2^2$;
3. $y = \Phi \cdot a_k^{n+\frac{1}{2}} - dt \cdot \mu P$;
c. Thresholding;
$u_i^{n+1} = e_r, r = \arg\max_j y_i$;
d. Updating a;
$a^{n+1} = \Phi^T \cdot u^{n+1}$;

The K-means algorithm [17] for finding K clusters proceeds iteratively by first choosing K centroids and then assigning each point to the cluster of the nearest centroid. The centroid of each cluster is then recalculated and the iterations continue until there is little change from one iteration to the next.

In both semi-supervised and unsupervised algorithms, we calculate the leading eigenvalues and eigenvectors of the graph Laplacian using the Nyström method [8] to accelerate the computation. This is achieved by calculating an eigendecomposition on a smaller system of size $M << N$ and then expanding the results back up to N dimensions. The computational complexity is almost $O(N)$. We can set $M << N$ without any significant decrease in the accuracy of the solution.

Suppose $Z = \{Z_k\}_{k=1}^{N}$ is the whole set of nodes on the graph. By randomly selecting a small subset X, we can partition Z as $Z = X \bigcup Y$, where X and Y are two disjoint sets, $X = \{Z_i\}_{i=1}^{M}$ and $Y = \{Z_j\}_{j=1}^{N-M}$ and $M << N$. The weight matrix W can be written as

$$W = \begin{bmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{bmatrix},$$

where W_{XX} denotes the weights of nodes in set X, W_{XY} denotes the weights between set X and set Y, $W_{YX} = W_{XY}^T$ and W_{YY} denotes the weights of nodes in set Y. It can be shown that the large matrix W_{YY} can be approximated by $W_{YY} \approx W_{YX} W_{XX}^{-1} W_{XY}$, and the error is determined by how many of the rows of W_{XY} span the rows of W_{YY}. We only need to compute W_{XX}, $W_{XY} = W_{YX}^T$, and it requires only $(|X| \cdot (|X| + |Y|))$ computations versus $(|X| + |Y|)^2$ when the whole matrix is used. Recently this algorithm has been further developed to run on parallel architectures [19, 20].

Experimental Results

To evaluate the performance of our method we need both choreographed video sequences to run controlled experiments and real-world videos to observe performance of our method in naturalistic settings. It is easy to define the ground truth for the choreographed videos since the motions of the person who takes the video are both discrete, and well-defined. For example, looking left and right never coincides with running. However, real-world body-worn video usually contains a combination of different motions with noise and it is therefore harder to define a ground truth.

Choreographed Video

The first video we use is QUAD [14]. We show one frame of the QUAD video in Fig. 10.6. This video is 4 min and 10 s in length and has 60 frames per second for a total of 15,000 frames. It contains nine ego-motions (stand still, turn left, turn right, look up, look down, jump, step in place, walk and run). The Ego wore a head-mounted GoPro camera. The nine actions were performed in order and repeated four times. The ground truth is shown in the first row of Fig. 10.7. The horizontal axis represents time and colors represent different ego-motion categories. The order of the movements are standing still, turning left and turning right repeatedly, looking up and looking down repeatedly, jumping, stepping, walking, running, turning left and then start the same series of motions again for another three times. We compute the feature vector for each two successive frames as described in section "Motion Features". Then we use K-means, the unsupervised graph MBO algorithm and the semi-supervised graph MBO algorithm for the ego-motion classification. We use

Fig. 10.6 One frame of the QUAD video

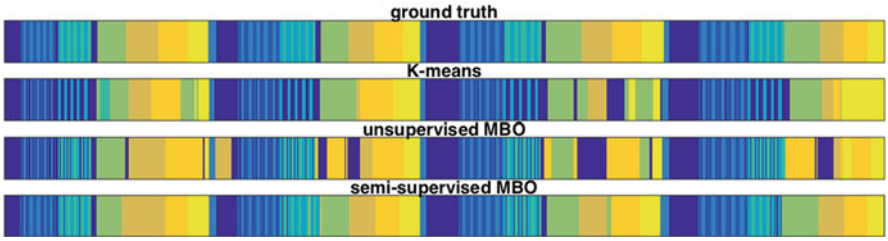


Fig. 10.7 Ego-motion classification results of the QUAD video. The 9 colors represent 9 different ego-motion classes: standing still (dark blue), turning left (moderate blue), turning right (light blue), looking up (dark green) and looking down (light green), jumping (bud green), stepping (aztec gold), walking (orange), runing (yellow)

Table 10.2 Accuracy summary of the QUAD data set

Accuracy	Overall	Average	1. Stand still	2. Turn left	3. Turn right	4. Look up
K-means	**64.84%**	**61.79%**	95.82%	72.26%	77.28%	73.24%
Unsupervised MBO	**66.62%**	**67.59%**	79.99%	76.82%	83.37%	69.41%
Semi-supervised MBO	**89.14%**	**88.74%**	87.90%	89.43%	92.80%	80.36%
Accuracy	5. Look down	6. Jump	7. Step	8. Walk	9. Run	
K-means	0	83.29%	49.29%	36.66%	68.25%	
Unsupervised MBO	77.82%	39.38%	43.54%	83.27%	54.68%	
Semi-supervised MBO	84.59%	92.71%	93.98%	84.52%	92.38%	

10% known labels (evenly sampled) in the semi-supervised graph MBO algorithm. The classification results of these three algorithms are shown in the 2nd, 3rd and 4th rows of Fig. 10.7. For the K-means and the unsupervised MBO algorithm, we ran the experiments several times and pick the best results here. Depending on the initialization, these two algorithms can converge to different local minima, which is common for most non-convex variational methods. The K-means algorithm gives relatively good results, except that it does not recognize the category of looking down and misclassifies some parts of running, jumping, small steps and walking. The unsupervised graph MBO algorithm gives results similar to K-means. The semi-supervised graph MBO algorithm with 10% known labels gives very accurate results. The accuracy summary of these three algorithms is shown in Table 10.2.

Real-World Body-Worn Video

We also investigated real-world body-worn videos. We use a data set from the Los Angeles Police Department. The videos are from police wearing chest-mounted cameras while patrolling areas of Los Angeles on foot. The videos record a wide array of police activities from basic patrol through foot chases and arrest. Our

ego-motion classification results may be used in modeling the routine activities of police and their interactions with the public.

Police BWV is not collected under controlled circumstances. Ego-motions may evolve rapidly without clear or discrete transitions. Much body worn video is collected at night impacting light and color saturation. The videos also have distortion due to the use of a fish-eye lens. Since there has been very little formal analysis of police BWV, there is a lack of appreciation for the diversity of police behavior likely to be encountered (i.e., very limited semantic dictionaries). The ground-truth is labeled by us without input from the police.

We show here the video segmentation result of one clip of police video. The video is 8 min and 16 s in length, with 14991 frames in total. This particular video was chosen for its diversity of ego-activities thereby presenting a challenging problem for ego motion classification. In the video, police arrive at an apartment building, talk with some people in front of the building, go upstairs, wait outside a room, enter and search the room, leave the room, walk downstairs, and talk to several people outside the building. We define four ego-motion categories in this video—standing (or very slow motions not easy to define), walking, going upstairs, and going downstairs. The ground truth classification of this video is shown in the first row of Fig. 10.8. The dark blue segments represent the category of standing or slow movements when the officer talks with others in front of the building. It also contains actions when the officer enters the room. The video of this period is very shaky and not easily defined as one motion category. The light blue segment corresponds to the walking category. The green segment corresponds to the police going upstairs and the yellow part is going downstairs.

We consider the semi-supervised algorithm for the police body-worn video. We are not using the unsupervised graph MBO algorithm because the result is not consistent. The results are shown in Fig. 10.8. K-means is included as a baseline method since it captures the difference between going upstairs and downstairs. However, K-means frequently misclassifies walking and going downstairs. Some standing frames are classified as other motion categories. This later result is reasonable since standing in this video combines some other movements. The semi-supervised graph MBO algorithm includes 10% known labels on this piece of video. The segmentation results are shown in the third row of Fig. 10.8. The

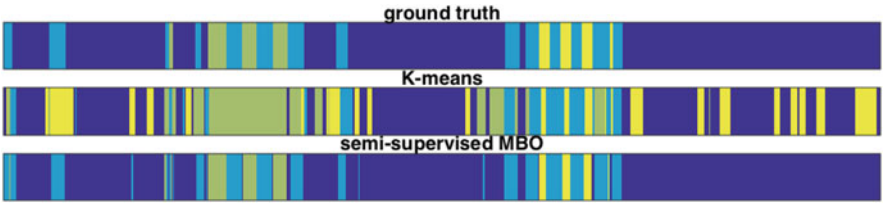


Fig. 10.8 Ego-motion classification results of the police video. The 4 colors represent 4 different ego-motion classes: standing or very slow motions and motions not easy to define (dark blue), walking (light blue), going upstairs (green) and going downstairs (yellow)

Table 10.3 Accuracy summary of the police body-worn video data set

Accuracy	Overall	Average	1.Stand	2.Walk	3.Upstairs	4.Downstairs
K-means	**63.62%**	**63.77%**	68.91%	37.78%	91.84%	56.53%
Semi-supervised MBO	**90.17%**	**74.09%**	96.10%	82.12%	83.45%	34.71%

semi-supervised graph MBO method is much better than K-means, and the four categories are all captured almost correctly. The accuracy summary is shown in Table 10.3. The overall accuracy of the semi-supervised graph MBO algorithm with 10% known labels is 90.17%.

Conclusion

In this paper, we investigate the task of discovering ego-motion categories from first-person videos. We deal with this problem in two steps. The first step is comparing two successive frames using the inverse compositional algorithm to extract signals containing motion and motion frequency information. Then we use unsupervised and semi-supervised clustering algorithms for classification. The semi-supervised graph based methods are particularly accurate using only 10% training data. We show promising results on both choreographed and real-world video data.

The potential for future advances in this area are significant particularly in relation to police body-worn video. At full deployment of body-worn video in 2018, the Los Angeles Police Department is projected to collect 3.2 million individual videos totaling more than 200K h of total video feed per year. This represents both a vast resource and a significant analytical challenge. The amount of data suggests that the full array of ego-motions practiced by police might eventually be discovered and subject to classification, moving us towards a realistic picture of the diversity of police activities. There will clearly be no lack of training data with which to tackle this problem. The same surfeit of video data is proving to be true in other domains outside of policing. Recognition of the diversity of ego-motion in policing activity may also lead to novel extensions of the methods into dyadic- and n-person motion models. In the dyadic-motion case there is much to be learned. It is well known that relative motion of individuals with respect to one another encodes fundamental social information [3]. For example, an individual running away from ego may encode avoidance or fear, while an individual running directly towards ego may encode attraction and threat. More complex social interactions may be captured in n-person motion models.

The challenges to achieving such outcomes with real-world video are also significant. In the police body-worn video case, semi-supervised classification clearly outperforms the unsupervised approach. Yet even a small fraction of fidelity points (10% in the current method) is probably infeasible given the volumes of video arriving each day. Semi-supervised methods will therefore need to rely on

as few fidelity points as possible. However another approach is video labeling where activities segmented in one video might be used as labels for semi-supervised segmentation in another video. This was demonstrated in [4, 5] for image labelling. It will also be necessary to consider how generalizable methods are across real-world video examples. Ideally, a handful of videos might be exhaustively labeled for ground-truth and these would then work across the growing set of videos. This is an empirical questions that we can start addressing now with the recognition that new methods may be needed to account for the variability of real-world video.

Finally, we also point out that body-worn video is but one sensor platform in what is increasingly a multi-sensor world. It is worth investigating whether there is an advantage to doing more with single sensors, or whether it is better to integrate the signals from many independent sensors. For example, we can imagine doing both ego-motion and scene topic classification from the same video sequence, or as an alternative use accelerometers to capture ego-motion and matching these data to scene classification from video. Importantly, the issues are not strictly technological. Police body-worn video is treated as evidence and therefore is subject to all of the evidence handling rules required by law. Each sensor implies a different packet of physical of evidence that must be maintained and handled appropriately. Future work will need to examine these sorts of tradeoffs in detail.

Acknowledgements The work was supported by the ONR grant N00014-16-1-2119, NSF grant DMS-1737770, NSF grant DMS-1417674, FUI project Plein Phare by BPI-France and NIJ Grant 2014-R2-CX-0101.

References

1. S. Baker, I. Matthews, Equivalence and efficiency of image alignment algorithms, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001)
2. S. Baker, I. Matthews, Lucas-Kanade 20 years on: a unifying framework. Int. J. Comput. Vis. **56**, 221–255 (2004)
3. H.C. Barrett, P.M. Todd, G.F. Miller, P.W. Blythe, Accurate judgments of intention from motion cues alone: a cross-cultural study. Evol. Hum. Behav. **26**(4), 313–331 (2005)
4. A.L. Bertozzi, A. Flenner, Diffuse interface models on graphs for classification of high dimensional data. Multiscale Model. Simul. **10**(3), 1090–1118 (2012)
5. A.L. Bertozzi, A. Flenner, Diffuse interface models on graphs for classification of high dimensional data. SIAM Rev. **58**(2), 293–328 (2016)
6. P. Bouthemy, M. Gelgon, F. Ganansia, A unified approach to shot change detection and camera motion characterization. IEEE Trans. Circuits Syst. Video Technol. **9**(7), 1030–1044 (1999)
7. F. Chung, *Spectral Graph Theory*, vol. 92 (American Mathematical Society, Providence, 1997)
8. C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nyström method. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 214–225 (2004)
9. C. Garcia-Cardona, E. Merkurjev, A.L. Bertozzi, A. Flenner, A.G. Percus, Multiclass data segmentation using diffuse interface methods on graphs. IEEE Trans. Pattern Anal. Mach. Int. **36**(8), 1600–1613 (2014)
10. H. Hu, J. Sunu, A.L. Bertozzi, Multi-class graph Mumford-Shah model for plume detection using the MBO scheme, in *Energy Minimization Methods in Computer Vision and Pattern Recognition* (Springer International Publishing, Berlin, 2015)

11. T. Huynh, M. Fritz, B. Schiele, Discovery of activity patterns using topic models, in *Proceedings of the 10th International Conference on Ubiquitous Computing* (2008), pp. 10–19
12. V. Kiani, H.R. Pourreza, Robust GME in encoded mpeg video, in *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia* (2011), pp. 147–154
13. J. Kim, H.S. Chang, J. Kim, H. Kim, Efficient camera motion characterization for MPEG video indexing. IEEE Int. Conf. Multimed. Expo **2**, 1171–1174 (2000)
14. K.M. Kitani, T. Okabe, Y. Sato, A. Sugimoto, Fast unsupervised ego-action learning for first-person sports videos, in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), pp. 3241–3248
15. B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in *Proceedings of the 7th International Joint Conference on Artificial intelligence (IJCAI)* (1981)
16. X. Luo, A.L. Bertozzi, Convergence analysis of the graph Allen-Cahn scheme. J. Stat. Phys. **167**(3), 934–958 (2017)
17. J. MacQueen, Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967)
18. W.W. Mayol, D.W. Murray, Wearable hand activity recognition for event summarization, in *International Symposium on Searable Computers* (2005), pp. 122–129
19. Z. Meng, A. Koniges, H. Yun, S. Williams, T. Kurth, B. Cook, J. Deslippe, A.L. Bertozzi: OpenMP parallelization and optimization of graph-based machine learning algorithms, in *OpenMP: Memory, Devices, and Tasks*, ed. by N. Maruyama, B. de Supinski, M. Wahib. Lecture Notes in Computer Science, vol. 9903 (Springer, Berlin, 2016). IWOMP
20. Z. Meng, E. Merkurjev, A. Koniges, A.L. Bertozzi, Hyperspectral image classification using graph clustering methods. Image Process. Line **7**, 218–245 (2017)
21. E. Merkurjev, T. Kostic, A.L. Bertozzi, An MBO scheme on graphs for classification and image processing. SIAM J. Imag. Sci. **6**(4), 1903–1930 (2013)
22. E. Merkurjev, E. Bae, A.L. Bertozzi, X.C. Tai, Global binary optimization on graphs for classification of high-dimensional data. J. Math. Imag. Vision **52**(3), 414–435 (2015)
23. H. Pirsiavash, D. Ramanan, Detecting activities of daily living in first-person camera views, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012), pp. 2847–2854
24. X. Ren, C. Gu, Figure-ground segmentation improves handled object recognition in egocentric video. CVPR **2**(3), 6 (2010)
25. J. Sánchez, The inverse compositional algorithm for parametric registration. Image Process. Line **6**, 212–232 (2016)
26. J. Sánchez, J.-M. Morel, Motion smoothing strategies for 2D video stabilization. SIAM J. Imag. Sci. **11**(1), 219–251 (2018)
27. E.H. Spriggs, F. de la Torre, M. Hebert, Temporal segmentation and activity classification from first-person sensing, in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2009), pp. 17–24
28. T. Starner, B. Schiele, A. Pentland, *International Symposium on Wearable Computers* (1998), pp. 50–57
29. T. Starner, J. Weaver, A. Pentland, Real-time american sign language recognition using desk and wearable computer based video. IEEE Trans. Pattern Anal. Mach. Intell. **20**(12), 1371–1375 (1998)
30. M. Stoer, F. Wagner, A simple min-cut algorithm. J. ACM **44**(4), 585–591 (1997)
31. L. Sun, U. Klank, M. Beetz, EYEWATCHME—3D hand and object tracking for inside out activity analysis, in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2009), pp. 9–16
32. S. Sundaram, W. Cuevas, High level activity recognition using low resolution wearable vision, in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2009), pp. 25–32

33. R. Szeliski, *Computer Vision: Algorithms and Applications* (Springer Science & Business Media, Berlin, 2010)
34. A. Szlam, X. Bresson, A total variation-based graph clustering algorithm for cheeger ratio cuts. UCLA CAM Report: 09-68 (2009)
35. Y. Van Gennip, A.L. Bertozzi, *Gamma*-convergence of graph Ginzburg-Landau functionals. Adv. Differ. Equ. **17**(11/12), 1115–1180 (2012)
36. U. Von Luxburg, A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)

Chapter 11
Synchronized Recovery Method for Multi-Rank Symmetric Tensor Decomposition

Haixia Liu, Lizhang Miao, and Yang Wang

Abstract Symmetric tensor decomposition is of great importance in applications. In this paper, we design a synchronized multi-rank symmetric Tensor Decomposition alternating minimization method. In this algorithm, we start from a careful initialization for the non-convex symmetric tensor decomposition and then perform an alternating minimization algorithm. Our contributions are as follows: (1) Our method is synchronized and there is no need for a greedy algorithm to get the multi-rank tensor decomposition. (2) Initialization is an important part in our proposed method. With a careful initialization, our proposed algorithm can converge to the global minimizer of the non-convex objective function. (3) The designed alternating minimization algorithm can give a highly accurate result. In numerical results, our proposed algorithm is much better than the simple gradient descent method from the same initialization. Moreover, our results show that with eigenvectors of random projection as initialization, we can quickly get the global solution by using simple alternating minimization algorithm, though finding the global minimum of this non-convex minimization problem is NP-hard.

Introduction

Tensor decomposition [5, 9, 14, 18, 27–30] plays an important role in data analysis, machine learning and dimension reduction. The problem of tensor decomposition is an extension of the singular value decomposition (SVD) of a symmetric matrix, which is an important tool in numerical linear algebra and its applications. There

H. Liu
School of Science, Harbin Institute of Technology (Shenzhen), Shenzhen, China
e-mail: liuhaixia@hit.edu.cn

L. Miao (✉) · Y. Wang
Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
e-mail: lmiao@ust.hk; yangwang@ust.hk

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5_11

is a particularly important class of tensors, called symmetric tensor. Symmetric tensors appear mainly as higher order derivatives or moments and cumulants of random vectors, which is applied for speech, mobile communications, machine learning, factor analysis of k-way arrays, biomedical engineering, psychometrics and chemometrics [4, 6, 7, 10, 12, 21, 26]. For a tensor $\mathscr{T} = (\mathscr{T}_{i_1 i_2 \cdots i_n}) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$, we call it symmetric if $n_1 = n_2 = \cdots = n_m$ and $\mathscr{T}_{i_1 i_2 \cdots i_m}$ is invariant under any permutation of the index $(i_1 i_2 \cdots i_m)$. The canonical polyadic (CP) decomposition of the symmetric tensor $\mathscr{T}$ is of the form

$$\mathscr{T} = \sum_{i=1}^{r} \lambda_i \underbrace{\mathbf{v}_i \otimes \cdots \otimes \mathbf{v}_i}_{m}, \tag{11.1}$$

where each $\lambda_i \underbrace{\mathbf{v}_i \otimes \cdots \otimes \mathbf{v}_i}_{m}, i = 1, \cdots, r$ is called a rank-1 component and $\otimes$ is the outer product between $\mathbf{x}$

$$(\underbrace{\mathbf{x} \otimes \mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{m})_{i_1 i_2 \cdots i_m} = \prod_{k=1}^{m} \mathbf{x}_{i_k},$$

m is the order of $\mathscr{T}$ and the CP rank of $\mathscr{T}$ is defined as the smallest r such that Eq. (11.1) holds. Finding the decomposition of a symmetric tensor $\mathscr{T}$ with fixed rank r is to solve the following optimization problem

$$\underset{\{\lambda_i, \|\mathbf{v}_i\|=1\}_{i=1}^{r}}{\arg\min} \left\| \mathscr{T} - \sum_{i=1}^{r} \lambda_i \underbrace{\mathbf{v}_i \otimes \cdots \otimes \mathbf{v}_i}_{m} \right\|^2. \tag{11.2}$$

is equivalent to

$$\underset{\{\mathbf{u}_i\}_{i=1}^{r}}{\arg\min} \left\| \mathscr{T} - \sum_{i=1}^{r} \operatorname{sign}(\lambda_i) \underbrace{\mathbf{u}_i \otimes \cdots \otimes \mathbf{u}_i}_{m} \right\|^2,$$

where $\mathbf{u}_i = \sqrt[m]{|\lambda_i|}\mathbf{v}_i$. There are many works for the decomposition of the symmetric tensor. Comon et al. [8] study various properties of symmetric tensors in relation to a decomposition of symmetric tensor and show that the rank and symmetric rank are equal for the symmetric tensors in a number of cases. To solve the model (11.2), many works are based on greedy algorithm, that is to find each rank-one component by iteration. To get each component of the decomposition for a symmetric tensor $\mathscr{T}$, the problem can be formulated as

$$\min \quad \frac{1}{2} \| \mathscr{T} - \underbrace{\mathbf{x} \otimes \mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{m} \|^2 \tag{11.3}$$

which is equivalent to

$$\begin{aligned} \max \quad & \langle \mathscr{T}, \underbrace{\mathbf{x} \otimes \mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{m} \rangle \\ \text{s.t.} \quad & \|\mathbf{x}\| = 1, \end{aligned} \tag{11.4}$$

Problem (11.4) is an optimization problem to find the best rank one approximation of the tensor $\mathscr{T}$ and known as the maximum Z-eigenvalue problem, which is NP-hard [15]. A variety of methods are introduced to solve the maximum Z-eigenvalue problems [15, 16, 18, 19, 22–24]. In 2009, Qi et al. [24] proposed Z-eigenvalue methods for solving the problem. Brachat et al. [3] reformulate Sylvester's method from the dual point of view and propose a method for symmetric tensor decomposition by computations with Hankel matrices. Kofidis and Regalia [16] consider the high-order power method (HOPM) and explain the condition under which the method is convergent for even-order symmetric tensors. Both Regalia and Kofidis [25] and Erdogan [13] introduce the Shifted variants of the power method (SVPM) is in the context of independent component analysis (ICA) and prove that they are monotonically convergent. In 2011, a shifted symmetric higher-order power method (SS-HOPM) [19] is introduced by Kolda et al. for computing tensor eigenpairs. In 2014, Anandkumar et al. [1] give a robust power method for tensor decompositions for learning latent variable models. They analyze robustness in theory for orthogonal decomposable tensors, although orthogonal symmetric tensor decomposition does not exist in general. Anandkumar et al. show that some symmetric tensor decompositions can be transformed to orthogonal decomposition by *whitening*. More recently, Kolda [17] points out there is a transformation from non-orthogonal tensor decomposition to orthogonal tensor decomposition when the matrix is with full column rank. Jiang et al. [15] solved the problem by matrix optimization under a rank-one constraint, which is hard for the large dimensional tensor. Recently, an iterative eigendecomposition algorithm [2] and a nonlinear model order reduction method [11] are introduced for symmetric tensor decomposition. Although there are so many works presented, the problem for symmetric tensor decomposition is still unsolved.

In this paper, we will introduce a synchronized multi-rank symmetric tensor decomposition method to calculate symmetric tensor decomposition. We consider the symmetric tensor decomposition problem as a non-convex optimization problem. Although there may be many stationary points, our proposed algorithm can converge to the global minimizer with a careful initialization. Numerical results show that our method is much better than the standard gradient descent method with the same initialization. Moreover, a careful initialization is very important for our non-convex tensor decomposition problem, this algorithm may not converge to the global minimization for a random initialization.

Before introducing our method, we will give some notations which will be used throughout the paper. A m-th order tensor $\mathscr{T} = (\mathscr{T}_{i_1 i_2 \cdots i_m}) \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$, which is in calligraphic letter. The Frobenius norm of $\mathscr{T}$ is

$$\|\mathscr{T}\| = \sqrt{\sum_{i_1, i_2, \cdots, i_m} \mathscr{T}^2_{i_1 i_2, \cdots i_m}}.$$

$\otimes$ denotes the outer product for tensors, that is, for $\mathscr{T}_1 \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ and $\mathscr{T}_2 \in \mathbb{R}^{n_{m+1} \times \cdots \times n_{m+l}}$, then $\mathscr{T}_1 \otimes \mathscr{T}_2 \in \mathbb{R}^{n_1 \times \cdots \times n_m \times n_{n+1} \times \cdots \times n_{m+l}}$ and

$$(\mathscr{A}_1 \otimes \mathscr{T}_2)_{i_1 \cdots i_{m+l}} = (\mathscr{T}_1)_{i_1 \cdots i_m} (\mathscr{T}_2)_{i_{m+1} \cdots i_{m+l}}.$$

$\otimes^m$ denotes the multiplication of the outer product for tensors, that is, for $\mathscr{T} \in \mathbb{R}^{n \times \cdots \times n}$,

$$\mathscr{T}^{\otimes^m} = \underbrace{\mathscr{T} \otimes \cdots \otimes \mathscr{T}}_{m}.$$

$\otimes_j$ denotes a tensor times a vector in mode j, that is,

$$(\mathscr{T} \otimes_j \mathbf{x})_{i_1 \cdots i_{j-1} i_{j+1} \cdots i_m} = \sum_{l=1}^{n_j} \mathscr{T}_{i_1 \cdots i_{j-1} l i_{j+1} \cdots i_m} \mathbf{x}_l.$$

The inner product of two tensors with the same size $\mathscr{T}_1 \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ and $\mathscr{T}_2 \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ is

$$\langle \mathscr{T}_1, \mathscr{A}_2 \rangle = \sum_{i_1, \cdots, i_m} (\mathscr{T}_1)_{i_1 \cdots i_m} (\mathscr{T}_2)_{i_1 \cdots i_m}.$$

$\mathscr{T}$ is a rank-one tensor if $\mathscr{T}$ can be represented as

$$\mathscr{T} = \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \cdots \otimes \mathbf{x}_m,$$

where $\mathbf{x}_i \in \mathbb{R}^n, i = 1, \cdots, m$. If $\mathbf{x}_1 = \cdots = \mathbf{x}_m$, then we say $\mathscr{T}$ is a symmetric rank-one tensor.

Theorem 11.1 *For a symmetric tensor* $\mathscr{T} \in \mathbb{R}^{\overbrace{n \times \cdots \times n}^{m}}$ *and* $\mathbf{x} \in \mathbb{R}^n$, *we have*

$$\mathscr{T} \otimes_1 \mathbf{x} \cdots \otimes_{i-1} \mathbf{x} \otimes_{i+1} \mathbf{x} \cdots \otimes_{j-1} \mathbf{x} \otimes_{j+1} \mathbf{x} \cdots \otimes_m \mathbf{x} = \mathscr{T} \otimes_1 \mathbf{x} \cdots \otimes_{m-2} \mathbf{x}.$$

The rest of this paper is organized as follows. In section "Synchronized Multi-Rank Symmetric Tensor Decomposition", we propose our synchronized multi-rank

symmetric tensor decomposition method to calculate symmetric tensor decomposition. Numerical results are illustrated in section "Numerical Implements". Finally, we conclude in section "Conclusion".

Synchronized Multi-Rank Symmetric Tensor Decomposition

Minimization of a non-convex objective function is known as a NP-hard problem and there may be very many stationary points. Tensor decomposition is a much challenging problem, even for the rank one approximation due to the objective function is not convex. In this section, we propose a heuristic algorithm of synchronized recovery for multi-rank symmetric tensor decomposition based on non-convex optimization. In section "Minimization of a Non-convex Objective Function", a minimization of a non-convex objective function algorithm is proposed, which is followed by a careful initialization via a matrix SVD (or eigenvalue decomposition) technique in section "Initialization".

Minimization of a Non-convex Objective Function

Finding the solution of a tensor decomposition (11.2) is a minimization of a non-convex problem. In order to solve (11.2), one simple way is to use gradient descent algorithm with a careful initialization (section "Initialization"). That is, iterate by the following scheme for $k = 0, 1, 2, \cdots$

$$U_{k+1} = U_k - \alpha_k \nabla_U \tilde{f}(U_k),$$

where $\tilde{f}(U) = \|\mathscr{T} - \sum_{i=1}^{r} \mathbf{u}_i^3\|$, $U = [\mathbf{u}_1, \cdots, \mathbf{u}_r]$, U_k is the solution in the k-th step and α_k is the step size. In order to obtain more accurate results, we propose a very useful algorithm based on the proximal gradient descent algorithm. Our alternating minimization algorithm uses the following iterations

1. **Update** $V = [\mathbf{v}_1, \cdots, \mathbf{v}_r]$, **with** $\Lambda = \left[\lambda_1^{(k)}, \cdots, \lambda_r^{(k)}\right]$ **fixed.**

$$\begin{aligned} &\underset{\{\|\mathbf{v}_i\|=1\}_{i=1}^{r}}{\arg\min} \left\| \mathscr{T} - \sum_{i=1}^{r} \lambda_i^{(k)} \mathbf{v}_i^{\otimes m} \right\|^2 \\ &= \underset{\{\mathbf{u}_i\}_{i=1}^{r}}{\arg\min} \left\| \mathscr{T} - \sum_{i=1}^{r} \operatorname{sign}(\lambda_i) \mathbf{u}_i^{\otimes m} \right\|^2 \triangleq f(\mathbf{u}_1, \cdots, \mathbf{u}_r) \end{aligned} \tag{11.5}$$

with $\mathbf{u}_i = \sqrt[m]{\left|\lambda_i^{(k+1)}\right|}\mathbf{v}_i$.

In order to solve (11.5), we can use proximal gradient descent:

(a)

$$\left[\tilde{\mathbf{u}}_1^{(k+1)}, \cdots, \tilde{\mathbf{u}}_r^{(k+1)}\right] = \left[\mathbf{u}_1^{(k)}, \cdots, \mathbf{u}_r^{(k)}\right] - \alpha_k \nabla_U f\left(\mathbf{u}_1^{(k)}, \cdots, \mathbf{u}_r^{(k)}\right).$$

(b) Project each $\mathbf{u}_i^{(k+1)}, i = 1, \cdots, r$ to the subspace $\{\mathbf{x} : \|\mathbf{x}\| = 1\}$. That is,

$$\mathbf{v}_i^{(k+1)} = \frac{\mathbf{u}_i^{(k+1)}}{\|\mathbf{u}_i^{(k+1)}\|}, i = 1, \cdots, r.$$

2. **Update Λ with V fixed.**

$$\arg\min_{\{\lambda_i\}_{i=1}^r} \left\| \mathscr{T} - \sum_{i=1}^r \lambda_i \left(\mathbf{v}_i^{(k+1)}\right)^{\otimes m} \right\|^2. \tag{11.6}$$

(11.5) is a non-convex problem about $\mathbf{v}_1, \cdots, \mathbf{v}_r$, we use gradient descent to solve it and then do a projection to the unit ball. For the fixed $\mathbf{v}_i^{(k+1)}, i = 1, \cdots, r$, (11.6) is a quadratic equation about $\lambda_1, \cdots, \lambda_r$, which can be solved by the least-square method. Obviously, $\mathbf{u}_i^{(k+1)} = \sqrt[m]{\left|\lambda_i^{(k+1)}\right|}\mathbf{v}_i^{(k+1)}$. The whole method is summarized in Algorithm 11.1.

Initialization

Initialization is an important part in the whole algorithm for non-convex optimization. The algorithm will converge to the global solution with a good initialization. In the following, we will introduce our initialization algorithm. As we all know, the problem of tensor decomposition is an extension of the singular value decomposition (SVD) of a symmetric matrix, which is an important tool in numerical linear algebra. One way to connect high-order tensor and matrix is projection which projects a tensor into a matrix along some vectors [18]. Formally, for

Algorithm 11.1 Rank-r approximation for symmetric tensor

Input: A symmetric tensor $\mathscr{T}$, the rank number r, stepsize α_0 and threshold ϵ.
Initialization: $U = [\mathbf{u}_1, \cdots, \mathbf{u}_r]$.
while $\left\|\mathscr{T} - \sum_{i=1}^r \mathbf{u}_i^{\otimes m}\right\|_F^2 > \epsilon$ **do**
$\left[\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_r\right] = [\mathbf{u}_1, \cdots, \mathbf{u}_r] - \alpha_k \nabla_U f(\mathbf{u}_1, \cdots, \mathbf{u}_r)$ with $f(\mathbf{u}_1, \cdots, \mathbf{u}_r) = \left\|\mathscr{T} - \sum_{i=1}^r \text{sign}(\lambda_i)\mathbf{u}_i^{\otimes m}\right\|_F^2$ $\mathbf{v}_i = \frac{\tilde{\mathbf{u}}_i}{\|\tilde{\mathbf{u}}_i\|}, i = 1, \cdots, r$ $[\lambda_1, \cdots, \lambda_r] = \arg\min_{\{\lambda_i\}_{i=1}^r} \left\|\mathscr{T} - \sum_{i=1}^r \lambda_i \mathbf{v}_i^{\otimes m}\right\|^2$ $\mathbf{u}_i^{(k+1)} = \sqrt[m]{|\lambda_i|}\mathbf{v}_i, i = 1, \cdots, r$
Output: $U = [\mathbf{u}_1, \cdots, \mathbf{u}_r]$.

$\mathscr{T} = \sum_{i=1}^{r} \mathbf{u}_i^{\otimes m} \in \mathbb{R}^{n^m}$, the projection along a vector $\mathbf{w}$ is

$$\begin{aligned}\mathscr{T}(I, I, \mathbf{w}, \cdots, \mathbf{w}) &= \mathscr{T} \otimes_1 I \otimes_2 I \otimes_3 \mathbf{w} \cdots \otimes_m \mathbf{w} \\ &= \sum_{i_3, \cdots, i_m = 1}^{n} \Pi_{j=3}^{m} w_{i_j} \mathscr{T}(:, :, i_3, \cdots, i_m)\end{aligned}$$

Due to the symmetry of symmetric tensor, projection in different mode will get the same matrix. For simplicity, we denote T as the projection of the tensor $\mathscr{T}$ in this article. In order to get the initialization, we do the following steps:

1. In order to get the matrix by projection, we generate the vector $\mathbf{w}$ randomly, where $T = \mathscr{T}(I, I, \mathbf{w}, \cdots, \mathbf{w})$.
2. SVD is performed on T to get the eigenvalues and eigenvectors.
3. The initialization is the eigenvectors corresponding to the largest r absolute values of eigenvalues.

Numerical Implements

In this section, we evaluate our proposed method for rank-r tensor decomposition. We use Signal-to-Noise Ratio (SNR) for evaluation, where SNR value of the given symmetric tensor $\mathscr{T}$ is

$$\text{SNR} = 20 \log \left(\frac{\|\mathscr{T}\|}{\|\mathscr{T} - \sum_{i=1}^{r} \mathbf{u}_i^{\otimes m}\|} \right),$$

with $\mathbf{u}_i, i = 1, \cdots, r$ calculated by Algorithm 11.1. In our numerical experiments, we mainly focus on the 3-order (m=3) symmetric tensor decomposition and the step size is a function of iteration number

$$\alpha_t = \min(1 - e^{-t/t_0}, \alpha_0),$$

where t is the iteration number and the values of $t_0 = 330$ and of $\alpha_0 = 0.3$. In the following, we give the numerical results for the exact rank-r tensor.

Example 11.1 (Rank-r Tensor Decomposition) We generate a vector $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \cdots, r$ randomly obeying Gaussian distribution, and create the symmetric rank-r 3-th order tensor with Gaussian noise

$$\mathscr{T} = \sum_{i=1}^{r} \mathbf{x}_i^{\otimes 3} + \delta \mathscr{T}_0,$$

where $\mathscr{T}_0$ is a symmetric Gaussian tensor.

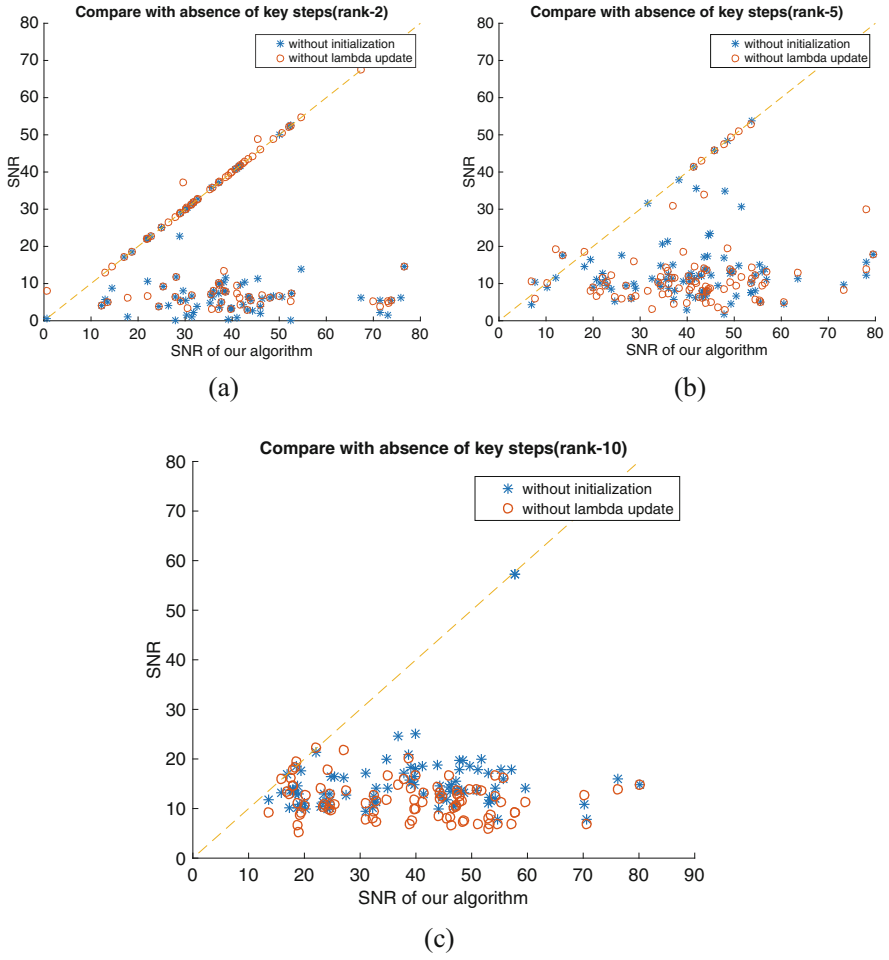


Fig. 11.1 The plot of comparison results. (**a**) Rank 2. (**b**) Rank 5. (**c**) Rank 10

In order to verify the values of our proposed algorithm, we perform 3 numerical experiments with $r \in \{2, 5, 10\}$. For each r, There are 90 tensors for test in total: we generate randomly 10 tensors for each $\delta \in \{10^{-3}, 10^{-2}, 2 \times 10^{-2}, 3 \times 10^{-2}, 4 \times 10^{-2}, 5 \times 10^{-2}, 0.1, 0.2, 0.5\}$, respectively. For each numerical experiment, we run the following three algorithms on each group of data:

1. Algorithm without a careful initialization, that is, we start with a random initialization.
2. Algorithm with a careful initialization in section "Initialization" plus gradient descent.
3. Our algorithm.

The SNR results of algorithms (1) versus (3) (in blue $*$) and (2) versus (3) (in red circle) for $r = 2, 5, 10$ in Fig. 11.1a, b and c, respectively. From these

Table 11.1 Stability of our algorithm

Rank	SNR(without noise)	SNR(with noise)	Rank	SNR(without noise)	SNR(with noise)
1	301.79	69.61	6	51.89	65.11
	299.26	65.91		119.74	75.92
	281.82	62.04		114.90	76.38
	291.09	74.48		119.01	74.66
	285.45	65.34		97.22	57.30
2	81.86	112.97	7	48.97	48.57
	112.97	67.75		77.17	16.95
	109.10	65.75		62.32	40.29
	121.88	74.08		120.89	76.10
	119.24	70.50		70.64	53.22
	111.71	70.24		107.32	67.4
	119.62	74.81		102.47	75.43
3	113.75	69.45	8	93.61	54.85
	102.22	74.33		79.77	77.28
	117.97	71.14		119.20	76.62
	125.98	76.67		43.53	36.33
	93.38	67.73		122.42	81.33
4	124.02	76.37	9	91.17	61.40
	120.17	73.51		140.75	73.17
	95.36	71.50		73.55	70.01
	110.71	74.22		68.48	38.29
	27.25	34.86		72.45	16.70
5	82.92	73.96	10	53.67	81.84
	86.37	77.85		86.41	59.53
	79.97	73.80		100.35	69.13

figures, we know that any part of our proposed algorithms is very important for this tensor decomposition problem. Though finding the global minimum of this non-convex minimization problem is NP-hard, our results show with eigenvectors of random projection as initialization, we can fast get global solution by using simple alternating minimization algorithm. We also give some stability analysis for our proposed algorithm. For each rank, we repeat the experiments five times. Table 11.1 shows the average of the SNR in five experiments for the data with or without noise for rank from 1 to 10. From the results in the table, our method is rather stable.

Beside, we also compare with the power method, which is a classical method for tensor decomposition [19, 20]. In the numerical experiments, we choose $m = 3$, $n = 15$ and $r \in \{2, 3, 4, 5\}$. For the two algorithms, the maximum iteration number is 1000 and the tolerance is 10^{-6}. Table 11.2 gives the comparison results, including SNR, iteration numbers and computational time. From the results stated in Table 11.2, the SNR by our method is very large, we know our method converges to the global minimizer.

Table 11.2 Performance for the symmetric tensor decomposition by Power method and our proposed method, Iterative num and Comp

$m = 3,\ n = 15$				
r	Algo.	SNR	Iterative num	Comp. time (s)
2	Pow	10.1767	19	0.305112
	Our	120.8453	296	0.186037
3	Pow	11.8510	31	0.347772
	Our	122.2619	369	0.274274
4	Pow	24.7118	8	0.295911
	Our	118.8325	417	0.379962
5	Pow	16.1120	12	0.368514
	our	117.9566	423	0.451466

Time are iterative numbers and the computational time (second)

Conclusion

In this paper, we design an alternating minimization method with a careful initialization for non-convex symmetric tensor decomposition. Our contributions are as follows: (1) Initialization is an important part in our proposed method. With a careful initialization, our proposed algorithm can converge to the global minimizer of the non-convex objective function. (2) The designed alternating minimization algorithm can give a highly accurate result. In numerical results, our proposed algorithm is much better than the simple gradient descent method. Moreover, our results show that with eigenvectors of random projection as initialization, we can quickly get the global solution by using simple alternating minimization algorithm, though finding the global minimum of this non-convex minimization problem is NP-hard. Stability analysis indicates that our algorithm is rather stable for the symmetric tensor decomposition problem.

References

1. A. Anandkumar, R. Ge, D. Hsu, S.M. Kakade, M. Telgarsky. Tensor decompositions for learning latent variable models. J. Mach. Learn. Res. **15**(1), 2773–2832 (2014)
2. K. Batselier, N. Wong, Symmetric tensor decomposition by an iterative eigen decomposition algorithm. J. Comput. Appl. Math. **308**, 69–82 (2016)
3. J. Brachat, P. Comon, B. Mourrain, E. Tsigaridas, Symmetric tensor decomposition, in *Signal Processing Conference, 2009 17th European* (IEEE, New York, 2009), pp. 525–529
4. J.-F. Cardoso, Blind signal separation: statistical principles. Proc. IEEE **86**(10), 2009–2025 (1998)
5. P. Chevalier, Optimal separation of independent narrow-band sources: concept and performance. Signal Process. **73**(1), 27–47 (1999)
6. A. Cichocki, S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, vol. 1 (Wiley, New York, 2002)
7. P. Comon, M. Rajih, Blind identification of under-determined mixtures based on the characteristic function. Signal Process. **86**(9), 2271–2281 (2006)

8. P. Comon, G. Golub, L.-H. Lim, B. Mourrain, Symmetric tensors and symmetric tensor rank. SIAM J. Matrix Anal. Appl. **30**(3), 1254–1279 (2008)
9. L. De Lathauwer, J. Castaing, Tensor-based techniques for the blind separation of DS-CDMA signals. Signal Process. **87**(2), 322–336 (2007)
10. L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
11. J. Deng, H. Liu, K. Batselier, Y.-K. Kwok, N. Wong, STORM: a nonlinear model order reduction method via symmetric tensor decomposition, in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)* (IEEE, New York, 2016), pp. 557–562
12. D.L. Donoho, X. Huo, Uncertainty principles and ideal atomic decomposition. IEEE Trans. Inf. Theory **47**(7), 2845–2862 (2001)
13. A.T. Erdogan, On the convergence of ICA algorithms with symmetric orthogonalization. IEEE Trans. Signal Process. **57**(6), 2209–2221 (2009)
14. A. Ferreol, P. Chevalier, On the behavior of current second and higher order blind source separation methods for cyclostationary sources. IEEE Trans. Signal Process. **48**(6), 1712–1725 (2000)
15. B. Jiang, S. Ma, S. Zhang. Tensor principal component analysis via convex optimization. Math. Programm. **150**(2), 423–457 (2015)
16. E. Kofidis, P.A. Regalia, On the best rank-1 approximation of higher-order supersymmetric tensors. SIAM J. Matrix Anal. Appl. **23**(3), 863–884 (2002)
17. T.G. Kolda, Symmetric orthogonal tensor decomposition is trivial (2015). arXiv preprint arXiv:1503.01375
18. T.G. Kolda, B.W. Bader, Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
19. T.G. Kolda, J.R. Mayo, Shifted power method for computing tensor eigenpairs. SIAM J. Matrix Anal. Appl. **32**(4), 1095–1124 (2011)
20. T.G. Kolda, J.R. Mayo, An adaptive shifted power method for computing generalized tensor eigenpairs. SIAM J. Matrix Anal. Appl. **35**(4), 1563–1581 (2014)
21. J.B. Kruskal, Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Linear Algebra Appl. **18**(2), 95–138 (1977)
22. L.-H. Lim, Singular values and eigenvalues of tensors: a variational approach (2006). arXiv preprint math/0607648
23. L. Qi, Eigenvalues of a real supersymmetric tensor. J. Symb. Comput. **40**(6), 1302–1324 (2005)
24. L. Qi, F. Wang, Y. Wang, Z-eigenvalue methods for a global polynomial optimization problem. Math. Programm. **118**(2), 301–316 (2009)
25. P.A. Regalia, E. Kofidis, Monotonic convergence of fixed-point algorithms for ICA. IEEE Trans. Neural Netw. **14**(4), 943–949 (2003)
26. N.D. Sidiropoulos, R. Bro, G.B. Giannakis, Parallel factor analysis in sensor array processing. IEEE Trans. Signal Process. **48**(8), 2377–2388 (2000)
27. N.D. Sidiropoulos, G.B. Giannakis, R. Bro. Blind PARAFAC receivers for DS-CDMA systems. IEEE Trans. Signal Process. **48**(3), 810–823 (2000)
28. L.I. Smith, A tutorial on principal components analysis. Cornell University, USA, 51:52 (2002)
29. A. Swami, G. Giannakis, S. Shamsunder, Multichannel ARMA processes. IEEE Trans. Signal Process. **42**(4), 898–913 (1994)
30. A.-J. Van Der Veen, A. Paulraj, An analytical constant modulus algorithm. IEEE Trans. Signal Process. **44**(5), 1136–1155 (1996)

Index

X.-C. Tai et al. (eds.), *Imaging, Vision and Learning Based on Optimization and PDEs*, Mathematics and Visualization,
https://doi.org/10.1007/978-3-319-91274-5