Inaugural-Dissertation

submitted to the

Combined Faculty for the
Natural Sciences and Mathematics

of

Heidelberg University, Germany

for the degree of
Doctor of Natural Sciences

put forward by
M. Sc. Lukas Kiefer
born in Malsch

Date of oral examination: ....

Efficient Algorithms for Mumford-Shah and Potts Problems

Advisors: Prof. Dr. Stefania Petra
Prof. Dr. Andreas Weinmann

Zusammenfassung

In dieser Arbeit betrachten wir Mumford-Shah- und Potts-Modelle sowie Verallgemeinerungen höherer Ordnung. Mumford-Shah- und Potts-Modelle gehören zu den bekanntesten variationellen Methoden zum kantenerhaltenden Glätten und Partitionieren von Bildern. Trotz ihrer intuitiven Formulierung ist ihre Anwendung nicht trivial, da es das Lösen schwieriger Minimierungsprobleme, welche insbesondere nicht konvex sind, erfordert. Das Entwickeln neuer algorithmischer Ansätze zum Lösen von Mumford-Shah- und Potts-Modellen ist bis heute ein aktives Forschungsgebiet und bildet den Schwerpunkt dieser Arbeit.

Zunächst betrachten wir die Situation für univariate Signale und stellen fest, dass durch das Anwenden von Mumford-Shah-Modellen höherer Ordnung bekannte Nachteile der klassischen Modelle erster Ordnung beim Verarbeiten von Daten mit steilen Steigungen behoben werden können. Die existierenden Ansätze zum Lösen der Modelle erster Ordnung könnten prinzipiell ebenfalls zum Lösen der Modelle höherer Ordnung genutzt werden, jedoch sind diese langsam oder werden für höhere Ordnungen numerisch instabil. Deshalb entwickeln wir einen neuen Algorithmus für univariate Mumford-Shah- und Potts-Modelle beliebiger Ordnung und zeigen, dass dieser die Modelle stabil in $O(n^2)$ löst.

Dann entwickeln wir Algorithmen für das inverse Potts-Modell, welches einen Ansatz zur gleichzeitigen Rekonstruktion und Partitionierung von Bildern, die nur indirekt in der Form von Messdaten vorliegen, darstellt. Weiterhin führen wir eine Konvergenzanalyse unserer Algorithmen durch. Insbesondere beweisen wir die Konvergenz gegen ein lokales Minimum des zugrunde liegenden NP-schweren Minimierungsproblems. Wir wenden die Algorithmen auf numerische Daten an und illustrieren ihre Vorteile.

Weiter wenden wir den mehrkanaligen Potts-Regularisierer auf das Rekonstruktionsproblem, welches in der multispektralen Computertomographie (CT) auftritt, an. Hierfür entwickeln wir einen neuen “Superiorizaton”-Ansatz. Dieser stört die Iterierten der Methode der konjugierten Gradienten so, dass wir im Vergleich zu diesen bessere Resultate im Hinblick auf den Potts-Regularisierer erhalten. In numerischen Experimenten beleuchten wir das Potential dieses Ansatzes indem wir ihn sowohl mit dem Potts-Modell-Ansatz aus der Literatur als auch mit den vorhandenen totalvariationsartigen Methoden vergleichen.

Als Nächstes betrachten wir das Mumford-Shah-Modell zweiter Ordnung für die kantenerhaltende Glättung von Bildern, welches –ähnlich wie im univariaten Fall– eine Verbesserung gegenüber dem klassischen Mumford-Shah-Modell für Bilder mit linearen Farbverläufen darstellt. Basierend auf Umformulierungen im Sinne von Taylor-Jets, d.h. speziellen Polynomfeldern, leiten wir diskrete Mumford-Shah-Modelle zweiter Ordnung her. Für diese diskreten Modelle entwickeln wir unter Verwendung eines ADMM-Schemas einen effizienten Algorithmus. Wir veranschaulichen das Potenzial der entwickelten Methode durch den Vergleich mit existierenden Ansätzen für das Mumford-Shah-Modell zweiter Ordnung. Ebenfalls veranschaulichen wir die Vorteile der entwickelten Methode im Hinblick auf das Detektieren von Bildkanten.

Schließlich betrachten wir das affin-lineare Potts-Modell für das Bildpartitionierungsproblem. Da viele Bilder innerhalb homogener Regionen lineare Verläufe aufweisen, führt das Anwenden des klassischen Potts-Modells häufig zu einer Übersegmentierung. Das affin-lineare

Potts-Modell berücksichtigt dieses Problem, indem es lineare Verläufe innerhalb der Segmente zulässt. Wir reformulieren das zugrunde liegende Minimierungsproblem im Sinne von Jets und entwickeln wiederum einen ADMM-basierten Algorithmus. In numerischen Experimenten zeigen wir, dass der entwickelte Algorithmus sowohl niedrigere Energiewerte als auch schnellere Laufzeiten erreicht als die Vergleichsmethode, welche auf der iterativen Anwendung des Graph-Cut-Algorithmus (mit α-Expansion-Moves) basiert.

Abstract

In this work, we consider Mumford-Shah and Potts models and their higher order generalizations. Mumford-Shah and Potts models are among the most well-known variational approaches to edge-preserving smoothing and partitioning of images. Though their formulations are intuitive, their application is not straightforward as it corresponds to solving challenging, particularly non-convex, minimization problems. The main focus of this thesis is the development of new algorithmic approaches to Mumford-Shah and Potts models, which is to this day an active field of research.

We start by considering the situation for univariate data. We find that switching to higher order models can overcome known shortcomings of the classical first order models when applied to data with steep slopes. Though the existing approaches to the first order models could be applied in principle, they are slow or become numerically unstable for higher orders. Therefore, we develop a new algorithm for univariate Mumford-Shah and Potts models of any order and show that it solves the models in a stable way in $O(n^2)$.

Furthermore, we develop algorithms for the inverse Potts model. The inverse Potts model can be seen as an approach to jointly reconstructing and partitioning images that are only available indirectly on the basis of measured data. Further, we give a convergence analysis for the proposed algorithms. In particular, we prove the convergence to a local minimum of the underlying NP-hard minimization problem. We apply the proposed algorithms to numerical data to illustrate their benefits.

Next, we apply the multi-channel Potts prior to the reconstruction problem in multi-spectral computed tomography (CT). To this end, we propose a new superiorization approach, which perturbs the iterates of the conjugate gradient method towards better results with respect to the Potts prior. In numerical experiments, we illustrate the benefits of the proposed approach by comparing it to the existing Potts model approach from the literature as well as to the existing total variation type methods.

Hereafter, we consider the second order Mumford-Shah model for edge-preserving smoothing of images which –similarly to the univariate case– improves upon the classical Mumford-Shah model for images with linear color gradients. Based on reformulations in terms of Taylor jets, i.e. specific fields of polynomials, we derive discrete second order Mumford-Shah models for which we develop an efficient algorithm using an ADMM scheme. We illustrate the potential of the proposed method by comparing it with existing methods for the second order Mumford-Shah model. Further, we illustrate its benefits in connection with edge detection.

Finally, we consider the affine-linear Potts model for the image partitioning problem. As many images possess linear trends within homogeneous regions, the classical Potts model frequently leads to oversegmentation. The affine-linear Potts model accounts for that problem by allowing for linear trends within segments. We lift the corresponding minimization problem to the jet space and develop again an ADMM approach. In numerical experiments, we show that the proposed algorithm achieves lower energy values as well as faster runtimes than the method of comparison, which is based on the iterative application of the graph cut algorithm (with α-expansion moves).

Acknowledgements

First and foremost, I would like to thank my advisors Prof. Stefania Petra, Faculty of Mathematics and Computer Science of Heidelberg University, and Prof. Andreas Weinmann, Department of Mathematics and Natural Sciences, Hochschule Darmstadt. I am grateful for their valuable mathematical insights, criticism and support during my PhD.

Furthermore, special thanks go to Prof. Martin Storath who always had a sympathetic ear for me and my questions particularly in the beginning of my PhD and introduced me to the art of crafting numerical experiments.

I am very grateful for the time I was able to spend with my colleagues resulting in great memories and fruitful discussions both in and outside of work. In particular, I would like to thank Jan Plier, Francesco Silvestri, Artjom Zern, Matthias Zisler, Alexander Zeilmann, Fabrizio Savarino, Ruben Hühnerbein and Jennifer Bachmann for our time working together and the nice atmosphere they helped to shape.

For proofreading parts of this thesis I thank Jan Plier, Marius Leonhardt, Martin Storath and Lena Kiefer, whose comments helped a lot to improve this thesis, and to ease my mind.

I thank my parents Susanne and Thomas and my sister Lena for their patience, support and always encouraging me in pursuing my goals.

Finally, I would like to acknowledge the financial support of the German Research Foundation (DFG) via the grant WE5886/4-1 and the funding received through the doctoral scholarship of the Hochschule Darmstadt during the writing of this thesis.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

$\mathbb{N}$	natural numbers
$\mathbb{Z}$	integer numbers
$\mathbb{R}$	real numbers
$\mathbb{R}^{m \times n}$	real $(m \times n)$-matrices
$\mathbb{R}^{m \times n \times K}$	real $(m \times n \times K)$-tensors
∂P	boundary of a set P
$\mathrm{length}(\Gamma)$	length of a set $\Gamma \subset \mathbb{R}^n$ in the sense of the $(n-1)$-dimensional Hausdorff measure
$l : r$	discrete “interval” $\{l, l+1, \ldots, r\}$ with boundaries $l, r \in \mathbb{N},\ l \leq r$
$\ker(A)$	kernel or nullspace of a linear operator A
$\mathrm{spectrum}(A)$	spectrum of a quadratic matrix A (the set of its eigenvalues)
$\mathrm{sign}(u)$	vector of entrywise signs of $u \in \mathbb{R}^n$
$\langle x, y \rangle$	standard inner product in $\mathbb{R}^n$
e_1, e_2	compass/standard unit vectors in $\mathbb{R}^2$
$\|u\|_0$	number of nonzero entries of $u \in \mathbb{R}^n$
$\nabla^k u$	k-th order finite differences of $u \in \mathbb{R}^n$
$\nabla_d^k u$	k-th order finite differences of $u \in \mathbb{R}^{m \times n}$ for direction $d \in \mathbb{Z}^2$
$J^k u_x$	k-th order Taylor jet of the function u at point x
$J u_x$	first order Taylor jet of the function u at point x
$\mathcal{O}$	big O notation for time complexity (also called Bachmann-Landau notation)
$\mathcal{E}^I$	optimal approximation error on a discrete interval I
$\mathrm{MSSIM}(u, f)$	mean structural similarity between images u and f

List of Publications

1. L. Kiefer and S. Petra. *Performance bounds for cosparse multichannel signal recovery via collaborative-TV.* International Conference on Sampling Theory and Applications. IEEE, 2017.
2. M. Storath, L. Kiefer, and A. Weinmann. *Smoothing for signals with discontinuities using higher order Mumford–Shah models.* Numerische Mathematik, 143(2):423–460, 2019.
3. L. Kiefer, S. Petra, M. Storath, and A. Weinmann. *Direct MRI segmentation from k-space data by iterative Potts minimization.* International Conference on Scale Space and Variational Methods in Computer Vision. Springer, 2019.
4. L. Kiefer, M. Storath, and A. Weinmann. *Iterative Potts minimization for the recovery of signals with discontinuities from indirect measurements: the multivariate case.* Foundations of Computational Mathematics, 2020.
5. L. Kiefer, M. Storath, and A. Weinmann. *An efficient algorithm for the piecewise affine-linear Mumford-Shah model based on a Taylor jet splitting.* IEEE Transactions on Image Processing, 29:921–933, 2020.
6. L. Kiefer, M. Storath, and A. Weinmann. *PALMS image partitioning – a new parallel algorithm for the piecewise affine-linear Mumford-Shah model.* Image Processing On Line, accepted, 2020.
7. L. Kiefer, M. Storath, and A. Weinmann. *An algorithm for second order Mumford-Shah models based on a Taylor jet formulation.* Preprint, 2020.
8. L. Kiefer, S. Petra, M. Storath, and A. Weinmann. *Multi-channel Potts-based reconstruction for multi-spectral computed tomography.* Preprint, 2020.

1. Introduction

1.1. Motivation

Smoothing and partitioning are important tasks in signal and image processing [174, 90]. Smoothing corresponds to the task of removing undesired or unimportant small scale variations in the data, e.g., noise, while preserving the important information. As a result, smoothing is often used as the first low-level processing step. Subsequent processing steps can then be applied to simplified data which still hold the important information. Partitioning describes the task of subdividing the domain of a signal or image into regions of some similar characteristics. In particular, smoothing and partitioning can serve as a basis for a segmentation pipeline [8]. Furthermore, smoothing and partitioning techniques are frequently used in the context of inverse problems, i.e., the task of reconstructing an unknown signal or image from data/measurements which are indirectly given and often incomplete. Corresponding examples are computed tomography, magnetic resonance imaging or microscopy [37, 101, 161, 35, 71, 131, 220].

Classical smoothing methods typically smooth a signal across its entire domain (e.g., smoothing splines, low-pass filters or Gaussian filters). As a result, if the signal possesses discontinuities or edges they will be typically smoothed out. However, these discontinuities often carry important information about the considered signal. For instance, they indicate abrupt changes of time series or boundaries of objects in images. In view of this, it is a natural objective to preserve discontinuities throughout the smoothing process [207]. In Figure 1.1, we give a visual comparison of classical non-discontinuity preserving smoothing and discontinuity-preserving smoothing of a noisy image.

First approaches to discontinuity preserving reconstruction and smoothing, respectively, were motivated by models from statistical mechanics. The Ising model [106] models the transformation of a magnetic unsorted material into a ferro magnet at high temperatures. Here, the spin at each atom has a binary value ±1 and neighboring atoms (in the sense of a discrete grid) with different spins increase the energy by a constant interaction energy. Potts [164] generalized the Ising model to $q \geq 2$ states – also in the context of statistical mechanics and in a fully discrete setting. With respect to this, the model is frequently called the *Potts model.* Geman and Geman [83] adopted the Potts model for the edge-preserving reconstruction of piecewise constant images. They motivated the model from a statistical viewpoint as the result can be seen as a maximum a posteriori estimate for the unknown image. In his technical report [145], Marroquin extended this model to continuous intensity values and piecewise smooth solutions. In [21], Blake and Zisserman proposed a model for the piecewise smooth reconstruction on discrete domains which they named weak string model for one-dimensional signals and weak membrane model for images. Further, they already proposed extensions to second order

(a) Noisy image

(b) Classical Gaussian filtering

(c) Edge-preserving smoothing

Figure 1.1.: Comparison of classical smoothing by Gaussian filtering and edge-preserving smoothing. While classical Gaussian filtering smooths over the sharp contours, edge-preserving smoothing preserves them (see, e.g., the eye of the parrot).

smoothness penalties named weak rod and weak plate model, respectively. In their seminal works [148, 149], Mumford and Shah studied joint reconstruction and partitioning by imposing piecewise smoothness and piecewise constancy on images in a spatially continuous framework. Subsequently, the model and its variants are commonly called *Mumford-Shah models*. Mumford-Shah models are frequently used for edge-preserving smoothing and the partitioning of images [8, 10, 185, 186, 5, 215] and time series [206, 79, 121].

Rudin, Osher and Fatemi proposed minimizing the total variation (see Definition A.3 in the appendix) of the image for edge-preserving smoothing [169]. Consequently, the corresponding model is typically called the ROF model. In comparison to Mumford-Shah models, the ROF model is easier to access both theoretically and algorithmically as it does not explicitly model the edge set and corresponds to a convex problem. The concept was later extended to involving higher order derivatives, that is, total generalized variation minimization [32].

Another class of methods is based on partial differential equations. It is a classical result that the convolution of an image with a Gaussian kernel corresponds to the solution of a linear diffusion equation –the heat equation– at a specific time (with the image as initial condition). In [160], Perona and Malik proposed a nonlinear diffusion based model for edge-preserving image smoothing which has been frequently called the Perona-Malik model. The basic idea of the Perona-Malik model is to preserve edges by using a diffusion tensor which slows down the diffusion at the locations of (potential) edges. A shortcoming of the method is the weak smoothing along edges. This problem was addressed in [199] by means of an anisotropic diffusion model. In particular, the idea is to incorporate the structure tensor [76, 18] to obtain a diffusion across edges as in the Perona-Malik model and a linear diffusion along edges.

The preservation of discontinuities is an important topic in other fields than signal and image smoothing as well. An example corresponds to finding numerical solutions of differential equations and in particular the (numerical) approximation of hyperbolic conservation laws in fluid dynamics. More precisely, shock-waves in the flow will result in sharp, discontinuous changes in flow variables. In this situation, applying standard methods results in undesired effects such as spurious oscillations near discontinuities. Therefore, so-called shock capturing methods have been developed. Prominent shock-capturing methods are essentially non-oscillatory (ENO) schemes [93] and corresponding weighted (WENO) schemes [140].

In this thesis, we concentrate on Mumford-Shah and Potts models. In particular, the main focus is the development of new algorithmic approaches to Mumford-Shah and Potts models as well as to higher order generalizations of them. Because of their instructive formulation and the appealing practical results, Mumford-Shah and Potts models are frequently employed in data and image processing. Therefore, and because of the algorithmic challenges that come with the corresponding minimization problems, numerical approaches to Mumford-Shah and Potts models are actively researched to this day [147, 216, 180, 215, 158, 74, 112, 110]. Please note that further related work will be discussed in the respective chapters.

The Mumford-Shah model. Mumford-Shah models are among the most well-known instances of *free-discontinuity problems*. Free-discontinuity problems are variational problems which are not formulated in terms of a function only, but in terms of a pair (u, Γ), where $\Gamma \subset \Omega$ is a closed set and u is a function on the domain Ω which is (sufficiently) smooth on $\Omega \setminus \Gamma$; see [61]. As a consequence, free-discontinuity problems provide an explicit modeling of the discontinuity set Γ.

We give a formulation of the classical Mumford-Shah model. To this end, we assume the data image $f : \Omega \to \mathbb{R}$ to be defined on the open bounded domain $\Omega \subset \mathbb{R}^2$. Then, the classical Mumford-Shah model for the data image f corresponds to solving the variational problem

$$\operatorname*{argmin}_{u,\Gamma} \int_\Omega |u(x) - f(x)|^2 \, dx + \beta^2 \int_{\Omega\setminus\Gamma} \|\nabla u(x)\|^2 \, dx + \gamma \, \mathrm{length}(\Gamma). \tag{1.1}$$

The minimum is taken with respect to the target variables u and Γ, where $u : \Omega \to \mathbb{R}$ is a piecewise smooth function and Γ is the corresponding discontinuity set. The existence of a minimizer was proven for a corresponding weak problem, where the discontinuity set Γ is not explicitly modeled anymore [1]. Rather, it is implicitly modeled via u which is in the space of special functions of bounded variation (see Definition A.4 in the appendix). We discuss the summands in the functional of (1.1) which correspond to a data penalty, a variation penalty, and a length penalty, respectively. More precisely, the ℓ^2-distance between u and f ensures closeness of the image u to the data f. The ℓ^2-norm of the gradient of u penalizes the variation on u on the complement of Γ *only*. Therefore, (1.1) promotes smoothness on u outside of Γ, while allowing for rapid changes at Γ, i.e., u is continuously differentiable on $\Omega \setminus \Gamma$. Concerning the length penalty, $\mathrm{length}(\Gamma)$ is defined as the one-dimensional Hausdorff measure of Γ (see

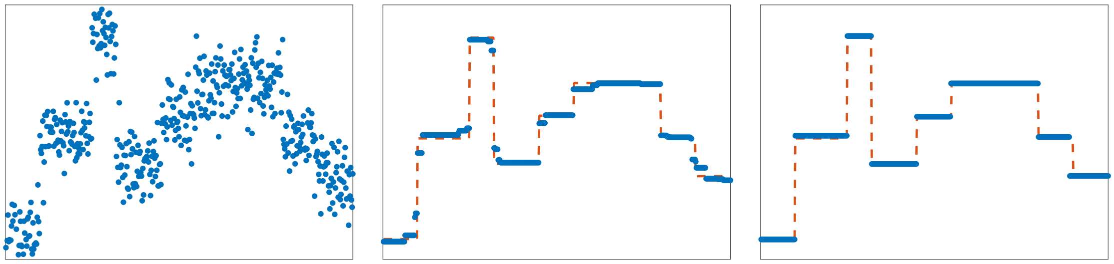

(a) Noisy data (b) ROF model (c) Potts model

Figure 1.2.: Recovery of a piecewise constant signal from noisy data with the convex ROF model and the Potts model. The ROF model tends to produce transitional points at the jumps of the signal. The Potts model recovers the piecewise constant signal almost perfectly. (The red dashed lines correspond to the ground truth signal.)

Definition A.1 in the appendix). To fix ideas, if Γ is smooth, this amounts to the length of the corresponding curve in the sense of differential geometry. In particular, when dropping the length penalty in (1.1), we would simply reproduce the data, i.e., $u = f$ and $\Gamma = \Omega$. Finally, the model parameters $\beta > 0$ and $\gamma > 0$ balance the variation penalty and the edge length, respectively.

The Potts model. The solutions of the Mumford-Shah model (1.1) are piecewise smooth. Mumford and Shah also studied the further restriction to *piecewise constant* solutions which corresponds to the limit case of (1.1) for $\beta \to \infty$. In this situation, any deviation of the gradient $\nabla u(x)$ from zero is infinitely penalized for all $x \notin \Gamma$. As a consequence, u is enforced to be piecewise constant. Further, the discontinuity set Γ cannot exhibit open ends (so-called "crack tips") anymore. Hence, Γ corresponds to the union of segment boundaries of a partition of the domain Ω. We call the resulting model the *Potts model* throughout the thesis at hand. (In the literature, it may also be referred to as the piecewise constant Mumford-Shah model.) It is given by

$$\begin{aligned} &\operatorname*{argmin}_{u,\mathcal{P}} \sum_{P\in\mathcal{P}} \int_P |u(x) - f(x)|^2 dx + \frac{\gamma}{2}\text{length}(\partial P) \\ &\text{subject to } u|_P \text{ is constant for all } P \in \mathcal{P}. \end{aligned} \tag{1.2}$$

The minimization in (1.2) takes place with respect to the partitions $\mathcal{P}$ of the function domain Ω into disjoint connected open sets P and the functions $u : \Omega \to \mathbb{R}$ which are constant on each segment $P \in \mathcal{P}$. The remaining parameter $\gamma > 0$ balances the data fidelity and the length penalty term. Note that the factor $\frac{1}{2}$ in (1.2) compensates for the double counting of boundaries, when using this formulation. In Figure 1.2, we illustrate the benefits of the Potts model compared to the ROF model (which may be seen as a convex relaxation of the Potts model) for the recovery of a piecewise constant signal from noisy data.

1.2. Contribution

This thesis contributes to the further development of algorithmic approaches to Mumford-Shah and Potts models. In particular, the contributions are given as follows.

(i) We study *higher order Mumford-Shah and Potts models* for univariate discrete signals and develop a new efficient solver which is based on dynamic programming and recurrence relations based on Givens rotations. Compared with the classical (first order) Mumford-Shah and Potts models, higher order Mumford-Shah and Potts models have the advantage that they can recover large gradients in the signal without introducing spurious segments ("gradient-limit effect").

   It is well-known that global minimizers of (univariate) Mumford-Shah and Potts functionals can be found via dynamic programming. However, using standard techniques for solving the emerging subproblems yields algorithms with cubic time complexity which become very slow for large data. For the classical first order models, there exists an efficient approach to these subproblems. On the flipside, generalizing this approach to higher orders becomes numerically unstable and the produced results become unsatisfying.

   We develop a new fast and stable approach to these subproblems. We prove that the resulting overall algorithm has quadratic time complexity and obtain stability results. The practical runtime is improved by incorporating pruning techniques.

   We study Mumford-Shah and Potts models of various orders in numerical experiments. Furthermore, we illustrate the effectiveness and stability of the proposed algorithm.

   Related material has been published in [179].

(ii) We propose new algorithmic approaches to the *inverse Potts model*. Compared to (1.2), the data f does not necessarily correspond to a (noisy) data image anymore, but may be given in terms of a linear imaging operator A. Thus, the inverse Potts model allows for joint reconstruction and partitioning of images which are only indirectly given in the form of (often incomplete) measured data.

   We start with a quadratic penalty relaxation of the problem. We approach this relaxed problem by solving a sequence of surrogate minimization problems which we see to decompose into univariate Potts problems. These are solved with solvers proposed in (i). We further propose a projection procedure which constructs a feasible solution (w.r.t. the original problem) from the result of the quadratic penalty relaxation. The thereby obtained result is close to a local minimum of the original problem for lower bounded imaging operators.

   The second algorithm approaches the original problem by iterating between running the first algorithm and updating the relaxation parameters according to a certain rule.

   We give a convergence analysis for the proposed schemes. In particular, we prove the convergence of the first algorithm to a local minimum of the relaxed problem and the convergence of the second algorithm to a local minimum of the original problem (on subsequences).

The applicability of both algorithms is illustrated in several experiments including image reconstruction from Radon data, partitioning of blurred images and classical image partitioning.

A corresponding paper has been published as [118].

(iii) We apply the (multi-channel) Potts prior to the reconstruction problem in multi-spectral computed tomography (CT).

We begin with considering the Potts prior in terms of the Potts model, which is algorithmically approached by an alternating direction method of multipliers (ADMM) scheme. Then we study the Potts prior in connection with the superiorization methodology. The main contribution of this chapter is the development of a superiorized algorithm that employs Potts-based target function reduction steps. As our prior has no calculable partial derivatives or subgradients, our approach contributes to derivative-free superiorization applied in a challenging non-convex setting.

In numerical experiments, we compare the Potts ADMM approach and the proposed Potts superiorization approach to the existing TV-type approaches by applying them to simulated multi-spectral CT data, which illustrates the benefits of employing the multi-channel Potts prior in multi-spectral CT reconstruction.

(iv) We consider *second and higher order Mumford-Shah models* for edge-preserving smoothing of images. Similarly to the univariate case, using the second order model can circumvent gradient limit effects.

We derive a new formulation of the corresponding variational problems in terms of Taylor jets which allows us to explicitly model the partial derivatives of the image. Using the jet formulation, we obtain a family of new jet-based discretizations.

We develop an algorithm based on the ADMM in a way that the subproblems decompose into one-dimensional segmented jet estimation problems. We propose a fast and stable solver for these subproblems based on our approach to higher order univariate Mumford-Shah problems.

We illustrate the benefits of the proposed approach by comparing it with existing approaches to the second order Mumford-Shah model qualitatively as well as quantitatively. Further, we derive an edge detection method based on the proposed algorithm. We compare it with classical Canny edge detection on the Berkley data test set and obtain improved results.

(v) We consider the *affine-linear Potts model* for the image partitioning problem. As many types of images posses linear trends within their segments (e.g., the sky in a landscape image), the classical (piecewise constant) Potts model (1.2) might lead to oversegmentation similarly as in the univariate case. A natural account for that is the extension to piecewise affine-linear solutions.

We reformulate the corresponding variational problem in terms of Taylor jets so that the solutions are piecewise constant Taylor jets. We approach the problem via the ADMM such that the emerging subproblems decompose into linewise piecewise constant jet estimation problems. For these, we obtain a fast and stable algorithm based on our approach to higher order univariate Potts problems.

We illustrate the benefits of the algorithmic approach in numerical experiments. In particular, the proposed approach has lower mean computation times and lower mean functional values than the state-of-the-art approach.

Related material has been published in [117]. Further, the resulting algorithm is made publicly available as part of the IPOL publication [119].[1]

1.3. Organization

In Chapter 2, we give mathematical preliminaries which are needed later on repeatedly. In Chapter 3, we consider higher order Mumford-Shah and Potts models for univariate signals. In particular, we derive an efficient and stable algorithm to solve them. In Chapter 4, we derive new algorithms for the inverse Potts model for images and data which are given indirectly. We give a convergence analysis and illustrate the proposed methods in numerical experiments. In Chapter 5, we consider image reconstruction from multi-spectral CT data with the multi-channel Potts model as well as a strategy based on superiorizing the conjugate gradients method towards Potts superiorized solutions. In numerical experiments, we demonstrate the benefits of the Potts-based approaches by comparing them to TV-based approaches. In Chapter 6, we consider the second Mumford-Shah model. We derive a new algorithm based on Taylor jets and illustrate our findings in qualitative and quantitative experiments. Chapter 7 deals with the affine-linear Potts model for image partitioning for which we develop an algorithm based on Taylor jets. We underpin our findings by a quantitative comparison of the proposed algorithm with the state-of-the-art approach. Finally, in Chapter 8, we discuss our findings and give suggestions for potential directions of future work.

---

[1] https://www.ipol.im/pub/pre/295/

2. Preliminaries

This chapter recalls some concepts which are used regularly throughout the thesis. In particular, we recall the augmented direction method of multipliers, give basics on the superiorization methodology and on Taylor jets and give a brief description of the mean structural similarity index. Proofs are omitted in general. Instead, we refer the interested reader to the given references.

2.1. Alternating direction method of multipliers (ADMM)

In this section, we give basics on the *alternating direction method of multipliers (ADMM)* which is a classical general framework for solving optimization problems with linear constraints [86, 81]; see also [25] for a review of the ADMM. In the context of imaging problems –which is of primary interest in this thesis– it is well-known that the ADMM performs particularly well [88, 153, 177, 25, 213]. The ADMM approaches the constrained minimization problem by solving a sequence of subproblems. Naturally, one requires that these subproblems are easier to solve than the original problem.

In the Euclidean setting, the ADMM approaches problems of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^m,\, y\in\mathbb{R}^n} f(x) + h(y) \\ &\text{subject to } Ax + By = b, \end{aligned} \tag{2.1}$$

where $f : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$, $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ and $A \in \mathbb{R}^{p\times m}, B \in \mathbb{R}^{p\times n}, b \in \mathbb{R}^p$. The *augmented Lagrangian* of this problem is given by

$$\mathcal{L}_\mu(x, y, \lambda) = f(x) + h(y) + \langle \lambda, Ax + By - b\rangle + \tfrac{\mu}{2}\|Ax + By - b\|_2^2, \tag{2.2}$$

where $\lambda \in \mathbb{R}^p$ is a Lagrange multiplier corresponding to the constraint in (2.1) and $\mu > 0$ is a corresponding penalty parameter. ($\mathcal{L}_\mu$ is called augmented Lagrangian since it amounts to the usual Lagrangian together with the quadratic penalty weighted by $\mu/2$.) The basic ADMM scheme is then given by

$$x^{j+1} \in \operatorname*{argmin}_x \mathcal{L}_\mu(x, y^j, \lambda^j) \tag{2.3}$$

$$y^{j+1} \in \operatorname*{argmin}_y \mathcal{L}_\mu(x^{j+1}, y, \lambda^j) \tag{2.4}$$

$$\lambda^{j+1} = \lambda^j + \mu(Ax^{j+1} + By^{j+1} - b). \tag{2.5}$$

Hence, in the course of the iterations the ADMM synchronizes the solutions of the local subproblems (2.3), (2.4) to a solution of the global problem (2.1). In this context, the variable λ can be interpreted as the weighted running sum of the constraint errors. Therefore, the ADMM is particularly useful if (2.3) and (2.4) can be solved efficiently.

Concerning convergence, first assume that the functionals f and h are closed, proper and convex. In this situation and if the (non-augmented) Lagrangian $\mathcal{L}_0$ has a saddle point, the iterations (2.3)-(2.5) converge to a feasible point (x^*, y^*), that is, $Ax^* + By^* = b$, the objective converges to the optimal value and λ^* is an optimal dual point [25].

A standard extension to the ADMM is to include a varying penalty parameter μ_j to improve practical convergence. A basic strategy is to initialize μ_0 by a small value and increase it after each iteration by a constant factor $\varphi > 1$, that is, $\mu_{j+1} = \varphi\mu_j$. Intuitively, in view of the penalty parameter, this allows the iterates x^j and y^j to evolve rather independently in the beginning until the penalty parameter μ^j becomes large which enforces the constraints in (2.1).

The minimization problems we deal with in this thesis are non-convex. Yet, originally formulated for convex problems, the ADMM has been successfully applied to non-convex problems as well [211, 54, 101, 214, 203]. In this context, the non-convex ADMM has gained increased theoretical attention in recent years [197, 102, 136, 194, 195].

2.2. Basics on the superiorization methodology

In this section, we briefly introduce the concepts of the superiorization methodology [60, 97] as we need them later in Chapter 5. Further details and examples can be found in [45, 47] and the references therein. For a comparison of the superiorization methodology to (convex) optimization, we refer to [47]. For an extensive overview, we refer the interested reader to the continuously updated bibliography on superiorization in [44].

The general framework of the superiorization methodology is to take an iterative algorithm, the *basic algorithm*, and *perturb* its iterates in a way such that the final result becomes more desirable in terms of an a priori chosen *target function*. More precisely, the perturbations are used to lower the value of the target function in a way that, in principle, does not compromise the convergence or, alternatively, the termination of the basic algorithm. Thereby, the target function values of the iterates are decreased so that the final result should become more desirable in terms of the target function. Thus, the perturbed algorithm is called the *superiorized* version (w.r.t. the target function) of the original algorithm. Often, the superiorization methodology requires only minor modifications of the basic algorithm. In particular, the existing code may only need to be slightly modified.

We note that both energy minimization methods and superiorization methods typically result in two steps within an algorithmic scheme: a forward step followed by a regularizing step. However, the interpretation of these steps is different.

We fix some notation and give the basic definitions. First, we denote by $\mathcal{S}_{\mathcal{T}} \subset \mathbb{R}^n$ the solution set of a given problem $\mathcal{T}$. For a start, we let $\mathcal{A}$ be an algorithmic operator, which generates sequences $\mathcal{X}_{\mathcal{A}} = (x^k)_{k\in\mathbb{N}_0}$ of the form

$$x^{k+1} = \mathcal{A}(x^k) \quad \text{for all } k \geq 0 \tag{2.6}$$

that converge to points in $\mathcal{S}_{\mathcal{T}}$ for any initial $x^0 \in \mathbb{R}^n$. To fix ideas, we give the following basic example.

Example 2.1. *A prototypical example for a problem $\mathcal{T}$ is given by the least squares problem*

$$\min_{x\in\mathbb{R}^n} \varphi(x) = \min_x \tfrac{1}{2}\|Ax - b\|_2^2, \tag{2.7}$$

where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. Consequently, $\mathcal{S}_{\mathcal{T}}$ corresponds to the set of minimizers of (2.7). *Here, $\mathcal{A}$ may be given by gradient descent steps, i.e., the Landweber iteration which is well-known to converge to a solution of* (2.6), *[72]. The iterates are given by*

$$x^{k+1} = \mathcal{A}(x_k) = x^k - \alpha\nabla\varphi(x^k), \tag{2.8}$$

for the gradient $\nabla\varphi(x) = A^TAx - A^Tb$, step-size $0 < \alpha < \frac{1}{\|A\|_2}$ and any initial $x^0 \in \mathbb{R}^n$.

The following notion of perturbation resilience [47] provides a criterion for an iterative algorithm $\mathcal{A}$ to be superiorizable by bounded perturbations.

Definition 2.2 (Bounded perturbation resilience)**.** Let $\mathcal{A}$ be an algorithmic operator which creates sequences $x^{k+1} = \mathcal{A}(x^k)$ that converge to points in the solution set $\mathcal{S}_{\mathcal{T}}$ of a given problem $\mathcal{T}$ for all starting points. Then $\mathcal{A}$ is *bounded perturbation resilient* if any sequence $y^{k+1} = \mathcal{A}(y^k + \beta_k v^k)$ also converges to a point in $\mathcal{S}_{\mathcal{T}}$ provided all $\beta_k v^k$ are bounded perturbations, i.e., $(v^k)_{k\in\mathbb{N}}$ is bounded, $\beta_k \geq 0$ for all k and $\sum_{k=0}^{\infty}\beta_k < \infty$.

Definition 2.2 is useful only for problems which do not involve noise[1] and have a nonempty solution set and basic algorithms whose theoretical convergence is guaranteed [97]. To this end, the notion of bounded perturbation resilience was introduced in [97]. It ensures that an iterative algorithm still terminates (rather than converges) when its iterates are (appropriately) perturbed. To give a precise definition of strong perturbation resilience, we need some preparations.

Definition 2.3 (Proximity function, ε-compatibility, proximity sets)**.** Given a problem $\mathcal{T}$, a *proximity function* is a function $\mathcal{P}r_{\mathcal{T}}(x) : \mathbb{R}^n \to [0,\infty)$ that measures how incompatible x is with $\mathcal{T}$. For any $\varepsilon > 0$ and any proximity function, we define that x is *ε-compatible* with $\mathcal{T}$ if $\mathcal{P}r_{\mathcal{T}}(x) \leq \varepsilon$. We call the sets

$$\Gamma_\varepsilon = \left\{x \in \mathbb{R}^n : \mathcal{P}r_{\mathcal{T}}(x) \leq \varepsilon\right\} \tag{2.9}$$

proximity sets.

For least squares problems $\mathcal{T}$ as in Example 2.1, the proximity function is typically given by the residual sum of squares $\mathcal{P}r_{\mathcal{T}}(x) = \varphi(x) = \frac{1}{2}\|Ax - b\|_2^2$ and ε is chosen according to the assumed level of noise.

Definition 2.4 (ε-output of a sequence with respect to Γ_ε)**.** For some $\varepsilon > 0$, a nonempty proximity set Γ_ε and a sequence $\mathcal{X} = (x^k)$ of points, the *ε-output* $O(\Gamma_\varepsilon, \mathcal{X})$ of $\mathcal{X}$ w.r.t. Γ_ε is defined to be the element x^k with smallest $k \in \mathbb{N}$ such that $x^k \in \Gamma_\varepsilon$.

We can now specify strong perturbation resilience of an algorithmic operator.

---

[1] For example, if $\mathcal{T}$ is a least squares problem, then the exact solutions $\mathcal{S}_{\mathcal{T}}$ are desirable if no noise is involved.

Definition 2.5 (Strong perturbation resilience)**.** Let Γ_ε denote the proximity set and $\mathcal{A}$ be an algorithmic operator. The algorithmic scheme (2.6) is *strongly perturbation resilient* if

(i) there is an $\varepsilon > 0$ such that the ε-output $O(\Gamma_\varepsilon, \mathcal{X}_{\mathcal{A}})$ of the sequence $\mathcal{X}_{\mathcal{A}}$ exists for every $x^0 \in \mathbb{R}^n$;

(ii) for all $\varepsilon \geq 0$ such that $O(\Gamma_\varepsilon, \mathcal{X}_{\mathcal{A}})$ is defined for every $x^0 \in \mathbb{R}^n$, we also have that $O(\Gamma_{\varepsilon'}, \mathcal{Y}_{\mathcal{A}})$ is defined for every $\varepsilon' \geq \varepsilon$ and for every sequence $\mathcal{Y}_{\mathcal{A}} = (y^k)$ generated by

$$y^{k+1} = \mathcal{A}(y^k + \beta_k v^k), \qquad \text{for all } k \geq 0, \tag{2.10}$$

where the terms $\beta_k v^k$ are bounded perturbations as defined in Definition 2.2.

We note that bounded perturbation resilience implies strong perturbation resilience given $\mathcal{S}_{\mathcal{T}} \neq \emptyset$ and $\mathcal{S}_{\mathcal{T}} \subset \Gamma_\varepsilon$ for some $\varepsilon > 0$.

We discuss the choice of the basic algorithm. The Landweber iteration (2.8) converges to solutions of least squares problems (2.7), but is known to have slow convergence for ill-conditioned problems. The conjugate gradient algorithm (CG) converges to solutions of (2.7) as well and at the same time copes better with ill-conditioned problems [192]. This motivates to consider the CG algorithm. In [218], the authors studied several variants of the CG method and identified a particular one which is strongly perturbation resilient. The following pseudocode describes this variant of the CG method.

Algorithm 2.1 (The conjugate gradient algorithm for least squares problems (2.7))**.**

```
• Set $k = 0$, pick an arbitrary initial point $x^0 \in \mathbb{R}^n$ and $\varepsilon > 0$.
• Set $p^0 = -A^T(Ax^0 - b)$ and $h^0 = A^T A p^0$.
• Iterate until $\|Ax^k - b\|_2^2 \leq \varepsilon$:
    1. $g = A^T(Ax^k - b)$
    2. $\alpha = \langle g, h^k \rangle / \langle p, h^k \rangle$
    3. $p^{k+1} = -g + \alpha p^k$
    4. $h^{k+1} = A^T A p$
    5. $\kappa = -\langle g, p^{k+1} \rangle / \langle p^{k+1}, h^{k+1} \rangle$
    6. $x^{k+1} = x^k + \kappa p$
    7. $k \leftarrow k + 1$
```

We record the strong perturbation resilience of the presented variant of the CG algorithm (Algorithm 2.1), which has been proven in [218, Theorem A.1].

Theorem 2.6. *The CG method as recorded in Algorithm 2.1 is strongly perturbation resilient.*

We saw that if the sequence (2.6) created by the basic algorithm $\mathcal{A}$ converges to some $x^* \in \mathcal{S}_\mathcal{T}$ and $\mathcal{A}$ is bounded perturbation resilient, then bounded perturbations may be used to find another $y^* \in \mathcal{S}_\mathcal{T}$ without altering the convergence of the basic algorithm. Alternatively, if $\mathcal{A}$ is strongly perturbation resilient, then one obtains some $y^* \in \Gamma_\varepsilon$ in a finite number of iterations, when perturbing its iterates by bounded perturbations.

Towards superiorized solutions, one perturbs the iterates of a basic algorithm by means of an a priori chosen target function $\phi : \mathbb{R}^n \to \mathbb{R}$. Typically, in the context of superiorization, the perturbations correspond to adding *non-ascending directions* w.r.t. the target function with appropriate step-sizes as in Definitions 2.2 and 2.5.

Definition 2.7 (Non-ascending direction)**.** Given a function $\phi : \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, the vector $v \in \mathbb{R}^n$ is a *non-ascending direction* for ϕ at x, if $\|v\| \leq 1$ and if there is $\bar{t} > 0$ such that

$$\phi(x + tv) \leq \phi(x) \qquad \text{for all } t \in (0, \bar{t}]. \tag{2.11}$$

The computation of non-ascending directions should be cheap, typically partial derivatives or subgradients of ϕ are used. In [47], the authors used proximal points, which was motivated by the common practice of applying multiple perturbation steps (towards the proximal mapping) before the next step of the basic algorithm.

The basic superiorization scheme in terms of non-ascending directions is given by

$$y^{k+1} = \mathcal{A}(\mathcal{P}(y^k)) \qquad \text{for all } k \geq 0, \tag{2.12}$$

where the perturbation procedure $\mathcal{P}(y^k) = y^k + \beta_k v^k$ adds a non-ascending direction v^k w.r.t. ϕ at y^k to y^k with appropriate step-size β_k in the sense of Definitions 2.2 and 2.5, respectively. In practice, the sequence $(\beta_k)_{k \in \mathbb{N}_0}$ is realized via $\beta_k = \beta_0 a^k$ for a positive β_0 and $0 < a < 1$ which, together with $\|v_k\| \leq 1$, ensure the summability of the perturbations as required in Definitions 2.5 and 2.7. Finally, the iteration (2.12) is typically stopped when the proximity satisfies $\mathcal{P}r_\mathcal{T}(y^k) \leq \varepsilon'$, where ε' depends on the (estimated) noise-level.

2.3. Basics on Taylor jets

In this section, we give basics on Taylor jets as we need them in Chapter 6 and Chapter 7. We also fix further notation regarding differentials. Throughout this section, we assume $d \geq 2$, $\Omega \subset \mathbb{R}^d$ to be open and $u : \Omega \to \mathbb{R}$ being (sufficiently) smooth.

Jets. We first informally recall the notion of a jet. A jet J of order k is a smooth mapping from Ω to the space of k-th order polynomials Π_k, i.e., a k-th order polynomial J_x is assigned to each point $x \in \Omega$. In our context, we mean by the attribute "smooth" that the mapping is sufficiently smooth, i.e., the mapping is C^k, meaning it is k times continuously differentiable where the concrete k depends on the situation and is suitably chosen such that all appearing differentials are indeed well-defined (in the classical sense).

Taylor jets. The *Taylor jet* $J^k u$ of order k of u is a field of truncated Taylor polynomials of order k. Formally, the Taylor jet is a smooth mapping from Ω to the space of k-th order polynomials Π_k, that is, $J^k u : \Omega \to \Pi_k$ and the evaluation of $J^k u$ in a point x is pointwise given by

$$J^k u_x(z) = u(x) + \sum_{l=1}^{k} \frac{1}{l!} D^l u_x(\underbrace{z - x, \ldots, z - x}_{l \text{ times}}) \tag{2.13}$$

for all $z \in \mathbb{R}^d$. (Note that (2.13) is well-defined as u is sufficiently smooth, i.e., u is at least k times continuously differentiable and, in particular, the partial derivatives of u up to order k exist.) We denote by $D^l u_x$ the l-th order differential of u. Recall that $D^l u_x$ corresponds to the symmetric l-linear form which assigns the l-th directional derivative of u at x along $z - x$. Its action on arguments $h_1, \ldots, h_l \in \mathbb{R}^d$ in terms of partial derivatives is given by

$$D^l u_x(h_1, \ldots, h_l) = \sum_{i_1, \ldots, i_l = 1}^{d} \partial_{i_1} \cdots \partial_{i_l} u(x) \cdot h_{1, i_1} \cdots h_{l, i_l}. \tag{2.14}$$

Note that h_{l, i_l}, is the i_l-th component of h_l and $\partial_{i_1} \cdots \partial_{i_l} u(x)$ denotes the l-th partial derivative of a function u for the coordinate directions $i_1, \ldots, i_l$, evaluated at the point x.

Example 2.8. *For a domain $\Omega \subset \mathbb{R}^2$ and order $k = 1$, the first order jet of a twice continuously differentiable function $u : \Omega \to \mathbb{R}$ at $x \in \Omega$ is the first order Taylor polynomial of u at x, i.e., for $z \in \mathbb{R}^2$ we have*

$$J^1 u_x(z) = u(x) + \partial_1 u(x)(z_1 - x_1) + \partial_2 u(x)(z_2 - x_2).$$

Representations of Taylor jets. The number of l-th order partial derivatives of u at a point x is $N_l = \binom{d+l-1}{l}$ and, in view of (2.14), the space of l-th order differentials has dimension N_l. More precisely, the identification of the l-th order differential of u at x with the vector of all l-th order partial derivatives is an isomorphism. By using multi-index notation this vector is given by $(\partial_\alpha u(x))_{|\alpha|=l} \in \mathbb{R}^{N_l}$, where $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ denotes a multi-index, $|\alpha| := \alpha_1 + \ldots + \alpha_d$, and $\partial_\alpha u(x) := \partial_1^{\alpha_1} \cdots \partial_d^{\alpha_d} u(x)$. (For instance, in the bivariate setting $d = 2$, we have $N_l = l + 1$, and thus $l + 1$ different partial l-th order derivatives.)

Accordingly, the Taylor polynomial of u at x can be described by its coefficients w.r.t. the monomial basis which is shifted to the point x, i.e., it is fully described by the vector

$$\Big(u(x), \partial_1 u(x), \partial_2 u(x), \ldots, (\partial_\alpha u(x))_{|\alpha|=k}\Big) \tag{2.15}$$

which is an element of $\mathbb{R}^r$, $r = \sum_{l=0}^{k} N_l = \binom{d+k}{k}$. The representation in (2.15) depends on the base point x. If we want to compare two Taylor polynomials, we have to represent Taylor polynomials w.r.t. the monomial basis shifted to another point y given by the monomial functions

$$z \mapsto (z - y)^\alpha := (z_1 - y_1)^{\alpha_1} \cdots (z_d - y_d)^{\alpha_d}, \ |\alpha| \le k. \tag{2.16}$$

We note that the (polynomial) coefficients of a Taylor polynomial –except the ones of the highest order k– depend on the base point x.

Example 2.9. *For a domain* $\Omega \subset \mathbb{R}^2$ *and a twice continuously differentiable bivariate function* $u : \Omega \to \mathbb{R}$, *the first order Taylor polynomial* Ju_x *of* u *at the point* x *can be represented (in* x*) by*

$$\big(u(x),\ \partial_1 u(x),\ \partial_2 u(x)\big) \tag{2.17}$$

and its representation at $y \neq x$ *is given by*

$$\big(u(x) + \partial_1 u(x)(y_1 - x_1) + \partial_2 u(x)(y_2 - x_2),\ \partial_1 u(x),\ \partial_2 u(x)\big). \tag{2.18}$$

As a consequence, we consider the Taylor polynomials as the objects to study rather than a specific vector representation. For notational convenience we use the representation of the Taylor jet by

$$J_x^k := \big(u(x), \partial_1 u(x), \partial_2 u(x), \ldots, (\partial_\alpha u(x))_{|\alpha|=k}\big) \tag{2.19}$$

which actually encodes a polynomial with coefficients that depend on the considered point dependent basis. In the bivariate first order case, we use the x-dependent vector

$$J_x := J_x^1 := (u(x), a_J(x), b_J(x)), \tag{2.20}$$

to represent the Taylor jet of the function u at the point x, where we let $a_J(x) := \partial_1 u(x)$ and $b_J(x) := \partial_2 u(x)$. As we will be mostly concerned with the bivariate first order case, we will frequently drop the order in the notation as on the left-hand side of (2.20).

2.4. Mean structural similarity index (MSSIM)

In this thesis, we will frequently need to assess the similarity between two images in a quantitative fashion. As an example, we consider an image which has been corrupted by noise. The result of a denoising method which has been applied to the noisy image should be “closer” to the uncorrupted image than the noisy data. Here, a notion of closeness can serve as a quantitative measure for the quality of a reconstructed image. The simplest choice for measuring this closeness corresponds to the Euclidean distance between two images. Frequently used quantities which are based on the Euclidean distance are the mean squared error or the (peak) signal-to-noise-ratio. However, it is known that these measures can differ considerably from the perceived visual quality; see [198] and the references therein. Therefore, we will use the mean structural similarity index (MSSIM) which is better suited to perceived visual quality/closeness than measures based on the Euclidean distance.

We start out by defining the structural similarity index (SSIM). To this end, we let $u, f \in \mathbb{R}^{m\times n}$ be two (digital) images of size $m \times n$. As a first step, we define the mean intensity $\mu_u = \frac{1}{mn}\sum_{i,j} u_{i,j}$, the (discrete) standard deviation $\sigma_u = \left(\frac{1}{mn-1}\sum_{i,j} |u_{i,j} - \mu_u|^2\right)^{\frac{1}{2}}$ and the (discrete) covariance $\sigma_{uf} = \frac{1}{mn-1}\sum_{i,j}(u_{i,j}-\mu_u)(f_{i,j}-\mu_f)$. The SSIM-index is a multiplicative combination of the so-called luminance term, the contrast term and the structural term which are defined as follows

$$l(u,f) = \frac{2\mu_u\mu_f + C_1}{\mu_u^2 + \mu_f^2 + C_1}, \qquad \text{luminance term,} \tag{2.21}$$

$$c(u,f) = \frac{2\sigma_u\sigma_f + C_2}{\sigma_u^2 + \sigma_f^2 + C_2}, \qquad \text{contrast term,} \tag{2.22}$$

$$s(u,f) = \frac{\sigma_{uf} + C_3}{\sigma_u\sigma_f + C_3}, \qquad \text{structural term,} \tag{2.23}$$

where the constants C_i are set to $C_1 = (0.01L)^2, C_2 = (0.03L)^2, C_3 = C_2/2$ for the dynamic range of the pixel values L (in this thesis: $L = 1$) [198]. Then the SSIM-index is given by

$$\mathrm{SSIM}(u,f) = l(u,f)^\alpha \cdot c(u,f)^\beta \cdot s(u,f)^\gamma, \tag{2.24}$$

where the exponents are typically chosen as $\alpha = \beta = \gamma = 1$, [198]. The SSIM is symmetric, i.e., $\mathrm{SSIM}(u,f) = \mathrm{SSIM}(f,u)$ bounded above by 1 and $\mathrm{SSIM}(u,f) = 1$ if and only if $u = f$.

Towards image quality assessment, that is, the MSSIM-value, the SSIM-index is applied locally rather than globally. To this end, the SSIM-index is computed for a local window around each pixel. This window corresponds to an 11 × 11 circular-symmetric Gaussian weighting function with standard deviation 1.5 normalized to unit sum [198] (boundary pixels are replicated). The MSSIM is then given by the mean of all these local SSIM-values, i.e.,

$$\mathrm{MSSIM}(u,f) = \frac{1}{mn}\sum_{j=1}^{mn} \mathrm{SSIM}(f_j, u_j), \tag{2.25}$$

where f_j and u_j denote the j-th window of f and u, respectively. By definition the MSSIM is symmetric, bounded above by 1 and $\mathrm{MSSIM}(u,f) = 1$ if and only if $u = f$. For multi-channel images, e.g., RGB images, the SSIM is computed for each channel in each pixel, where the windows are taken across the channels. Then the mean is taken over all these values.

3. An Efficient Algorithm for Univariate Higher Order Mumford-Shah and Potts Models

In this chapter, we consider higher order Mumford-Shah and Potts models for univariate data. This chapter is based on the publication [179].

Organization of the chapter. In Section 3.1, we give an overview and discuss related work. In Section 3.2, we describe and discuss higher order univariate Mumford-Shah and Potts models. Next, in Section 3.3, we develop a fast solver for higher order Mumford-Shah problems and for higher order Potts problems, and we analyze the stability. A numerical study is given in Section 3.4. Finally, Section 3.5 summarizes the chapter.

3.1. Overview and related work

Smoothing is an important processing step when working with measured signals or time series. Signals of interest often have discontinuities which typically indicate significant changes; examples are the cross-hybridization of DNA [175, 67, 104], the reconstruction of brain stimuli [207], single-molecule fluorescence resonance energy transfer [109], cellular ion channel functionalities [103], photo-emission spectroscopy [78] and the rotations of the bacterial flagellar motor [176, 157]. Further examples can be found in [137, 138] and [78] as well as in the references therein.

It is a standard technique to smooth signals with splines. However, classical spline smoothing does not preserve discontinuities. The (discrete) Mumford-Shah model extends spline smoothing by allowing for discontinuities between smooth signal parts [148, 149, 21]. However, a significant limitation of the Mumford-Shah model emerges when applied to data with linear or polynomial trends. In particular, if the slope of the signal is too high, the Mumford-Shah model produces spurious segments. This is known as the *gradient limit effect* [21]. It originates in the first order smoothing of the Mumford-Shah model, i.e., the model penalizes deviations from constant signals to enforce smoothness (within segments). It is often more desirable to penalize deviations from locally linear or locally polynomial signals instead.

A natural approach to preserving linear or polynomial trends is the usage of higher order smoothing penalties. This gives rise to *higher order Mumford-Shah models* given by

$$(u^*, \mathcal{I}^*) = \underset{u \in \mathbb{R}^N, \mathcal{I} \text{ partition}}{\operatorname{argmin}} \quad \|u - f\|_2^2 + \sum_{I \in \mathcal{I}} \beta^{2k} \|\nabla^k u_I\|_2^2 + \gamma \, |\mathcal{I}|, \tag{$\mathcal{P}_{k,\beta,\gamma}$}$$

where the data is denoted by $f \in \mathbb{R}^N$, $\mathcal{I}$ is a partition of the data domain $\Omega = (1, \dots, N)$ and $\nabla^k u_I$ denotes the k-th order finite difference operator (which is a discrete approximation of the k-th derivative of u) applied to the vector u restricted to the discrete "interval" or segment I of the domain partition $\mathcal{I}$. More precisely, the functional value comprises cost terms for the data deviation, for the inner energy of a (smoothing) spline on the single segments, and for the complexity of the partition. The parameters $\beta, \gamma > 0$ are model parameters which control the tradeoff between data fidelity and smoothness. To fix ideas, choosing a large β leads to stronger smoothing on the segments, and choosing a large γ leads to fewer segments. Together, the minimizer u^* is a (discrete) piecewise k-th order spline approximation with smoothing parameter β to f. The discontinuities or breakpoints of u^* are given by the boundaries of the corresponding partition $\mathcal{I}^*$.

The result of $(\mathcal{P}_{k,\beta,\gamma})$ consists of a partition $\mathcal{I}^*$ and a corresponding (piecewise) smoothed signal u^*. The partition $\mathcal{I}^*$ can serve as a basis for identifying segment neighborhoods [6] and indicates changepoints of the signal [121]. The corresponding optimal signal u^* can serve as a smoother for a signal with discontinuities [21, 206].

It is worth considering the limit cases of β and γ, respectively. Concerning β, we note that the kernel of ∇^k corresponds to the polynomials of maximum degree $k-1$. Thus, the limit case of $(\mathcal{P}_{k,\beta,\gamma})$ for $\beta \to \infty$ can be written as

$$\operatorname*{argmin}_{u,\mathcal{I}} \|u - f\|_2^2 + \gamma\, |\mathcal{I}|, \quad \text{s.t. } u_I \text{ is a polynomial of maximum degree } k-1 \text{ for all } I \in \mathcal{I}. \tag{$\mathcal{P}_{k,\infty,\gamma}$}$$

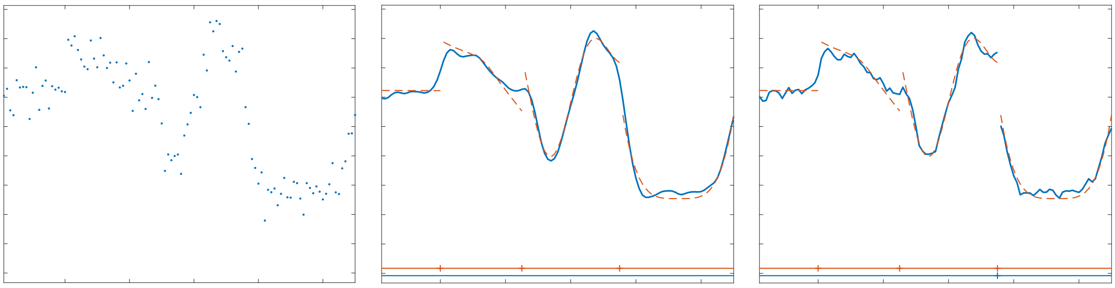
Since the special case $k = 1$ in $(\mathcal{P}_{k,\infty,\gamma})$ is known as the Potts model, we call $(\mathcal{P}_{k,\infty,\gamma})$ the *higher order Potts model.* In the other limit case $\gamma \to \infty$, no discontinuities are allowed anymore as the functional value would be infinitely large. Hence, it reduces to the (discrete) k-th order spline approximation

$$\operatorname*{argmin}_{u} \|u - f\|_2^2 + \beta^{2k} \|\nabla^k u\|_2^2, \tag{$\mathcal{P}_{k,\beta,\infty}$}$$

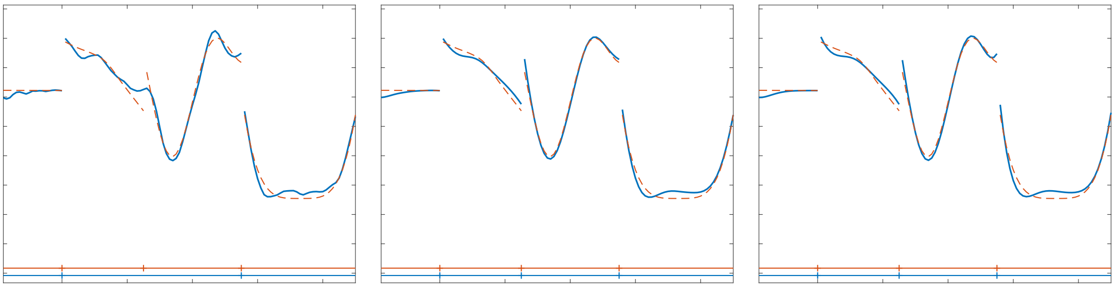
which is a classical method for smoothing data; see [204, 193].

To sum up, an important advantage of higher order Mumford-Shah and Potts models is that they provide a richer representation, most notably the ability to reproduce polynomials of order $k-1$. As a consequence, they improve the estimation of data with linear, quadratic or other polynomial trends. On the flipside, higher order Mumford-Shah models are theoretically and practically more involved: important questions are the uniqueness of minimizers, the choice of a reasonable order, and efficient and stable solvers for the involved non-convex optimization problem. An illustration of the smoothing effect of higher order Mumford-Shah models in comparison to classical splines and to first order models is given in Figure 3.1.

In this chapter, we study higher order Mumford-Shah and Potts models for univariate data. We first discuss basic properties of the models, prove that the solutions are unique for almost all data and discuss connections with related models. As a central contribution, we develop a fast minimization algorithm for higher order Mumford-Shah and Potts models. We prove that its worst case complexity is quadratic in the length of the signal and derive stability results. By including pruning techniques, the practical complexity can be improved to a linear one. Finally, we provide a numerical study in which we particularly compare higher order models with the first order models w.r.t. runtime and reconstruction quality.



(a) Noisy signal with discontinuities (b) Smoothing spline (c) Classical Mumford-Shah ($\mathcal{P}_{1,\beta,\gamma}$)



(d) Higher order Mumford-Shah ($\mathcal{P}_{2,\beta,\gamma}$) (e) Higher order Mumford-Shah ($\mathcal{P}_{3,\beta,\gamma}$) (f) Higher order Mumford-Shah ($\mathcal{P}_{4,\beta,\gamma}$)

Figure 3.1.: Smoothing of a noisy signal with discontinuities (a) using various models (with optimal model parameters). The red dashed lines depict the ground truth; the streaks at the bottom indicate the discontinuities or segments of the ground truth (top, red) and the computed segmentations (bottom, blue). (b) Classical spline approximation smooths out the discontinuities. (c) The classical Mumford-Shah model allows for discontinuities, but the estimate misses most of them and the result remains noisy. (d–f) The higher order Mumford-Shah models provide improved smoothing and segmentation. In particular, the third and the fourth order models recover the discontinuities of the ground truth.

Related work. Most literature studies the members of the family ($\mathcal{P}_{k,\beta,\gamma}$) individually for the orders $k = 1$ or $k = 2$ and for $\beta < \infty$ or $\beta = \infty$. As pointed out in Chapter 1, the piecewise constant variant ($\mathcal{P}_{1,\infty,\gamma}$) is often called Potts model [164]. (Therefore we refer to ($\mathcal{P}_{k,\infty,\gamma}$) as higher order Potts model.) An early work by Bruce [34] used a related weighted version of the Potts model for minimum error quantization. Bellman [14] studied curve fitting by segmented straight lines, that is, the case ($\mathcal{P}_{2,\infty,\gamma}$). Kleinberg and Tardos [125] discuss this approach and refer to it as segmented least squares problem. As we have seen in Chapter 1, the first order problems ($\mathcal{P}_{1,\beta,\gamma}$) for arbitrary parameters β have been studied in the seminal works by Mumford and Shah [148, 149]. Hence, we call the family ($\mathcal{P}_{k,\beta,\gamma}$) higher order Mumford-Shah models. At about the same time, Blake and Zisserman [21] studied ($\mathcal{P}_{1,\beta,\gamma}$) under the name weak string model. They already pointed out a shortcoming of the first order model, namely the gradient limit effect which we mentioned above; it describes the phenomenon that the first order model penalizes large slopes and produces spurious extra intervals to account for them; cf. Figure 3.2. In the same work, they proposed a second order extension, called the weak rod model. Compared with $(\mathcal{P}_{2,\beta,\gamma})$, the weak rod model has an extra penalty for discontinuities in the first derivative. A recent survey on (continuous domain) Blake-Zisserman models can be found in [43]. To the author's knowledge, a systematically study of the models $(\mathcal{P}_{k,\beta,\gamma})$ for arbitrary orders k has not been conducted.

For the first order models, other penalties and other data spaces than the ℓ_2-based penalties have been studied. Fornasier and Ward [75] consider ℓ_p-based variation penalties with $p \geq 1$. The variant with ℓ_1 variation penalty –also known as truncated total variation– was investigated in [101, 126]. ℓ_1-based data terms have been used for example in [79, 202, 184]. Mumford-Shah and Potts models for manifold-valued data spaces have been investigated in [200].

Recall that computing solutions of the Mumford-Shah problem corresponds to solving a non-convex minimization problem. It is well-known that the univariate situation $(\mathcal{P}_{k,\beta,\gamma})$ can be formulated as a partitioning problem, which can be solved by dynamic programming; see [14, 20, 6, 206, 107, 79]. The state-of-the-art solver has the worst case complexity $O(N^2\phi(N))$. Here, $\phi(N)$ denotes the costs of computing a spline approximation error on an interval of maximum length N; see [206, 125, 79]. Killick et al. [121] proposed a pruning strategy which accelerates the algorithm: if the expected number of segments $|\mathcal{I}^*|$ grows linearly in N and if the expected log-likelihood fulfills certain estimates, the expected complexity will be $O_P(N\phi(N))$. Another pruning scheme has been established in [181]. An algorithm for computing the solutions of the first order Mumford-Shah problem for all parameters γ simultaneously was proposed in [79].

By exploiting the band structure of the accompanying linear systems on each segment, one can achieve the cost $\phi(N) = N$, so that the overall worst case complexity of the above solver is $O(N^3)$. For the first order problem $(\mathcal{P}_{1,\beta,\gamma})$, [101] proposed an algorithm of $O(N^2)$ worst case complexity. It utilizes a fast computation scheme for the approximation errors which was first proposed by Blake [20]. However, this scheme is based on algebraic recurrences and a generalization to arbitrary orders k is difficult. The approximation errors associated with the higher order Potts model $(\mathcal{P}_{k,\infty,\gamma})$ may be computed in constant time by precomputing the moments of the data so that $\phi(N) = 1$; see [79]. Unfortunately, as the algorithm works with large cumulative sums, they suffer from loss of significance and become unstable for higher orders. Blake and Zisserman [21] also discussed an alternative dynamic programming approach of Viterbi-type. Here, the range of the signal u has to be restricted to a finite range of R real values so that we cannot expect to obtain an exact minimizer without prior knowledge of the exact range. In addition, this approach costs $O(NR^{k+1})$ which becomes too expensive for higher orders [21].

3.2. Higher order univariate Mumford-Shah and Potts models

We start with the basic setup and notation. We aim at recovering the unknown signal $g \in \mathbb{R}^N$ of length N from noisy samples given by

$$f_n = g_n + \eta_n, \quad n = 1, \ldots, N, \tag{3.1}$$

where we assume additive white Gaussian noise, i.e., the η_n are independently and identically distributed Gaussian random variables with zero mean and variance σ^2. We denote a (discrete) interval I with lower bound l and upper bound r by the Matlab-type notation $I = l : r = \{l, l+1, \dots, r\}$. Further, for convenience, we use the Matlab-type indexing $x_I = x_{l:r} = (x_l, x_{l+1}, \dots, x_r)$. We define a partition into intervals $\mathcal{I}$ of the discrete signal domain $\Omega = 1 : N$ as a set of intervals $I \subset \Omega$ which are pairwise disjoint and segment Ω. In particular, $I \cap J = \emptyset$ holds for all distinct $I, J \in \mathcal{I}$ and $\bigcup_{I \in \mathcal{I}} = \Omega$. As we deal only with partitions into intervals in this chapter, we briefly call $\mathcal{I}$ a partition. Moreover, the squared Euclidean norm of u is denoted by $\|u\|^2 = \|u\|_2^2 = \sum_{n=1}^N u_n^2$ for $u \in \mathbb{R}^N$.

3.2.1. First order Mumford-Shah model and the gradient limit effect

We first recall some important properties of the classical first order model. On the discrete domain Ω, we write the first order Mumford-Shah problem as

$$(u^*, \mathcal{I}^*) = \underset{u \in \mathbb{R}^N,\ \mathcal{I} \text{ partition of } \Omega}{\operatorname{argmin}} \sum_{n=1}^{N} |u_n - f_n|^2 + \beta^2 \sum_{I \in \mathcal{I}} \sum_{i=1}^{|I|-1} |(u_I)_{i+1} - (u_I)_i|^2 + \gamma\, |\mathcal{I}|. \tag{3.2}$$

In this instructive formulation, the arguments of the minimization are the signal u and the partition $\mathcal{I}$. For the derivation of algorithms based on dynamic programming [20, 79], the problem is typically reformulated in terms of partitions only given by

$$\mathcal{I}^* = \underset{\mathcal{I} \text{ partition of } 1:N}{\operatorname{argmin}} \sum_{I \in \mathcal{I}} \left(\mathcal{E}^I + \gamma \right), \quad \text{with } \mathcal{E}^I = \min_{v \in \mathbb{R}^{|I|}} \sum_{i=1}^{|I|} |v_i - f_i|^2 + \beta^2 \sum_{i=1}^{|I|-1} |v_{i+1} - v_i|^2. \tag{3.3}$$

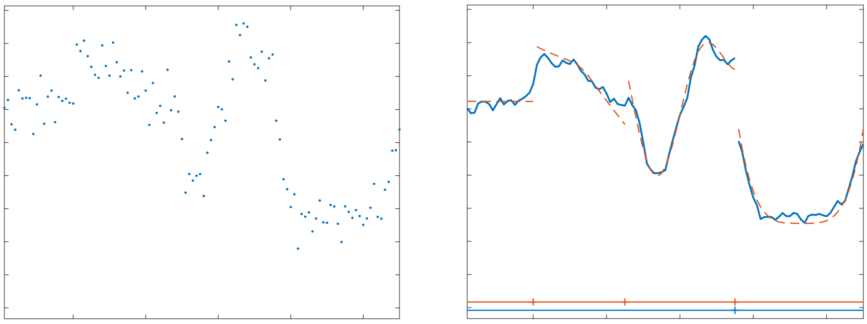
Here $\mathcal{E}^I$ is the (optimal) first order discrete spline approximation error on the interval I, that is, $\mathcal{E}^I$ corresponds to the minimal sum of the data fidelity term and the smoothing term in (3.2) w.r.t. the reduced data f_I.

For the first order model (3.2), the signal u^* can be recovered from the partition $\mathcal{I}^*$ and vice-versa. As a consequence, we can also write the functional in terms of the signal u only. The corresponding formulation (in terms of a potential in u) reads

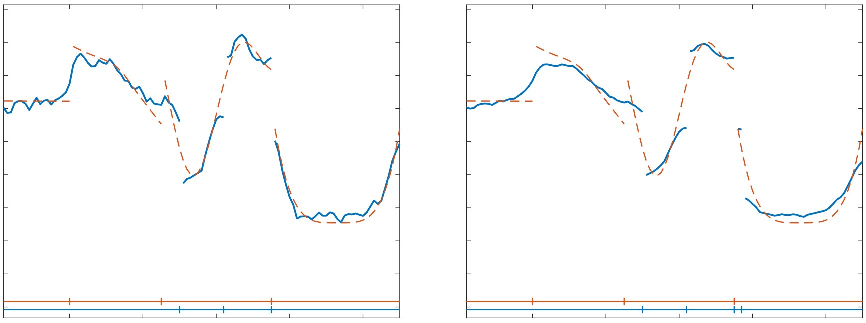
$$u^* = \underset{u \in \mathbb{R}^N}{\operatorname{argmin}} \sum_{n=1}^{N} |u_n - f_n|^2 + \sum_{n=1}^{N-1} \min\left(\beta^2 |u_{n+1} - u_n|^2, \gamma \right). \tag{3.4}$$

This formulation is the basis for the derivation of algorithms based on iterative thresholding techniques [75].

The classical Mumford-Shah model is limited by the so-called gradient limit effect [21]. It describes the introduction of extra (spurious) segments to account for steep slopes in the data. Thus, linear or polynomial trends in the data are not well approximated which is illustrated in Figure 3.2. We see that the solution of the first order model (with optimally chosen model parameters) does not recover all discontinuities. However, when choosing the model parameters to allow for more discontinuities to compensate this, spurious extra segments are introduced at steep slopes. This exemplifies that first order models are not comprehensive enough for data with locally linear or polynomial trends.



(a) Noisy signal with discontinuities (b) Result of $(\mathcal{P}_{1,\beta,\gamma})$ with optimal β, γ



(c) Smaller complexity penalty γ (d) Larger smoothing parameter β

Figure 3.2.: Limitations of the classical (first order) Mumford-Shah model: (b)-(d) show reconstructions of the signal in (a) by the first order Mumford-Shah model $(\mathcal{P}_{1,\beta,\gamma})$. (b) Result for the optimized model parameters $\gamma = 0.04$, $\beta = 1.3625$ (w.r.t. the ℓ_2-error to the ground truth). The smoothing is insufficient and not all discontinuities are detected. (c) The lower complexity penalty $\gamma = 0.02$ yields more but falsely located discontinuities. (d) Increasing the smoothing parameter to $\beta = 3$ leads to stronger smoothing, but to spurious segments as well. In either case, the first order model suffers from the gradient limit effect, i.e., the creation of spurious segments at steep slopes.

3.2.2. Basic properties of higher order Mumford-Shah and Potts models

In contrast to the first order Mumford-Shah model which makes use of first order finite differences, the k-th order Mumford-Shah models incorporate k-th order finite differences. Towards a compact notation, we denote by the matrix $\nabla^k \in \mathbb{R}^{(q-k)\times q}$ the k-th order finite difference operator. For example, for $k = 1$ and $k = 2$, ∇^k is given by

$$\nabla = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(q-1)\times q} \quad \text{and} \quad \nabla^2 = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(q-2)\times q}$$

respectively. Regarding higher orders $k \geq 3$, we recall the notion of the (discrete) convolution of two vectors $v \in \mathbb{R}^m$, $w \in \mathbb{R}^p$. It is given by the vector $v * w \in R^{m+p-1}$ with entries $(v * w)_i = \sum_j v_j w_{i-j+1}$ where the sum is taken over all admissible subscripts j. Then, for higher orders

$k \geq 3$ the row pattern of the matrix ∇^k corresponds to the k-fold convolution $t * t * \cdots * t$ of the finite difference vector $t = (-1, 1)$ with itself.

With this notation at hand, we write the higher order Mumford-Shah problem $(\mathcal{P}_{k,\beta,\gamma})$ on the discrete domain Ω as

$$(u^*, \mathcal{I}^*) = \underset{u \in \mathbb{R}^N,\ \mathcal{I} \text{ partition of } \Omega}{\operatorname{argmin}} \underbrace{\sum_{n=1}^{N} |u_n - f_n|^2}_{\text{data penalty}} + \underbrace{\beta^{2k} \sum_{I \in \mathcal{I}} \sum_{i=1}^{|I|-k} (\nabla^k u_I)_i^2}_{\text{smoothness penalty}} + \underbrace{\gamma\, |\mathcal{I}|.}_{\text{complexity penalty}} \tag{3.5}$$

We recall that the objective function incorporates a data penalty (or data fidelity) term, which ensures closeness to the data, a smoothness term, which provides smoothness within segments, and a complexity penalty that controls the fineness of the partition. As a result, on each segment I of the partition $\mathcal{I}$ the minimizer u^* is a k-th order discrete smoothing spline for the data f. The model parameter β controls the amount of smoothing within each segment and the model parameter γ determines the costs of introducing a new segment. In view of the definition of ∇^k, the order k is the derivative order of the (discrete) splines. Thus, polynomials of order $k - 1$ have zero smoothness penalty on a segment.

Formulation as partitioning problem. In Section 3.2.1, we reformulated the first order Mumford-Shah model in terms of the partition only, i.e., as a partitioning problem. Accordingly, we formulate higher order Mumford-Shah models (3.5) as partitioning problems by

$$\mathcal{I}^* = \underset{\mathcal{I} \text{ partition of } 1{:}N}{\operatorname{argmin}} \sum_{I \in \mathcal{I}} (\mathcal{E}^I + \gamma). \tag{3.6}$$

Here, $\mathcal{E}^I$ is the k-th order spline approximation error on the interval I which given by

$$\mathcal{E}^I = \min_{v \in \mathbb{R}^{|I|}} \|v - f_I\|_2^2 + \beta^{2k} \|\nabla^k v\|_2^2 = \min_{v \in \mathbb{R}^{|I|}} \sum_{i=1}^{|I|} |v_i - (f_I)_i|^2 + \sum_{i=1}^{|I|-k} \beta^{2k} (\nabla^k v)_i^2. \tag{3.7}$$

From the definition of ∇^k it follows immediately that the approximation error $\mathcal{E}^I$ is only well-defined for vectors of length greater than k. Hence, for intervals with length $|I| \leq k$, we have $\mathcal{E}^I = 0$. The signal u^* is recovered from an optimal partition $\mathcal{I}^*$ by solving the least squares problems corresponding to the respective approximation errors for each interval, i.e.,

$$u_I^* = \underset{v \in \mathbb{R}^{|I|}}{\operatorname{argmin}} \|v - f_I\|_2^2 + \beta^2 \|\nabla^k v\|_2^2, \quad \text{for all } I \in \mathcal{I}^*. \tag{3.8}$$

Thus, after an optimal partition $\mathcal{I}^*$ of the domain Ω has been found, the corresponding optimal signal estimate u^* is uniquely determined.

A formulation of higher order models in terms of the signal u only, similarly to the first order model (3.4), is not feasible. In contrast to the first order model, a solution u can correspond to multiple partitions which can be seen in the following basic example.

Example 3.1. *Let $k \geq 2$, $\gamma < 2/3$ and β be sufficiently large. Then for the data $f = (0, 1, 0)$, a minimizer is given by $u = f$. However, the three distinct partitions $\big\{(1, 2), (3)\big\}$, $\big\{(1), (2, 3)\big\}$ and $\big\{(1), (2), (3)\big\}$ all induce this $u = f$.*

We elaborate on the relation between an optimal signal u^* and the optimal partition $\mathcal{I}^*$ in terms of block matrices. Taking k-th order differences of a signal u on each interval of a partition $\mathcal{I}$ amounts to applying a block diagonal matrix $L_\mathcal{I}$ which is given by

$$L_\mathcal{I} = \begin{pmatrix} L_{|\mathcal{I}_1|} & & & \\ & L_{|\mathcal{I}_2|} & & \\ & & \ddots & \\ & & & L_{|\mathcal{I}_M|} \end{pmatrix}, \qquad \text{with} \qquad L_p = \nabla^k \in \mathbb{R}^{(p-k)\times p}, \tag{3.9}$$

where the k-th order finite difference matrices ∇^k of the appropriate size $(p-k)\times p$ is defined as above. (The symbol $|\mathcal{I}_m|$ denotes the cardinality of the m-th element of the partition $\mathcal{I}$.) If $p \leq k$, we let L_p denote an "empty" block of length p. While the number of columns of $L_\mathcal{I}$ is equal to the signal length N, the number of rows depends on the length of the intervals in the partition with minimum size $k+1$, i.e., the number of rows of $L_\mathcal{I}$ is given by $\sum_{m=1}^{M} \max(|I_m| - k, 0)$. As an example, for $k = 2$ and the partition $\mathcal{I} = \{(1,2,3,4),(5,6),(7,8,9)\}$, we have

$$L_\mathcal{I} = \begin{pmatrix} 1 & -2 & 1 & & & & & & \\ & 1 & -2 & 1 & & & & & \\ & & & & & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 9}.$$

We use the block matrix notation (3.9) to formulate the minimization problem (3.5) as

$$\operatorname*{argmin}_{u\in\mathbb{R}^N,\ \mathcal{I}\text{ partition}} \quad \beta^{2k}\, \|L_\mathcal{I} u\|_2^2 + \|u - f\|_2^2 + \gamma\, |\mathcal{I}|. \tag{3.10}$$

If we fix a partition $\mathcal{I}$, a minimizer $u_\mathcal{I}$ satisfies the normal equations of the corresponding least squares system, i.e., after taking derivatives w.r.t. u, a minimizer $u_\mathcal{I}$ solves the linear system

$$\beta^{2k} L_\mathcal{I}^T L_\mathcal{I} u_\mathcal{I} + (u_\mathcal{I} - f) = 0. \tag{3.11}$$

We note that the system matrix on the left-hand side of (3.11) has full rank for all $\beta \geq 0$. Hence, (3.11) has the unique solution

$$u_{f,\mathcal{I}} = S_{\mathcal{I},\beta} f, \qquad \text{where } S_{\mathcal{I},\beta} = (\beta^{2k} L_\mathcal{I}^T L_\mathcal{I} + \mathrm{id})^{-1}. \tag{3.12}$$

For readability, we drop the subscript if the dependence on $\mathcal{I}$ or β is clear. Towards a closed-form $\mathcal{G}'_\mathcal{I}(f)$ for the functional value of the higher order Mumford-Shah functional restricted to the partition $\mathcal{I}$, we plug (3.12) into (3.5) and obtain

$$\mathcal{G}'_\mathcal{I}(f) = \beta^{2k}\, \|L_\mathcal{I} S_\mathcal{I} f\|_2^2 + \|S_\mathcal{I} f - f\|_2^2 + \gamma\, |\mathcal{I}|. \tag{3.13}$$

As a consequence, a minimizing partition $\mathcal{I}^*$ corresponds to a minimizer of $\mathcal{I} \mapsto \mathcal{G}'_\mathcal{I}(f)$.

Remark 3.2. In Example 3.1, all three partitions induce an empty matrix $L_\mathcal{I}$. Hence, they yield the same map $S_{\mathcal{I},\beta}$ and by (3.12) the same signal $u = f$.

Minimum functional values and minimum segment lengths. The following lemma concerns the minimal functional values of the higher order Mumford-Shah problem. The proof follows an argument similar to the one used in [21, Appendix C] for a continuous domain second order problem.

Lemma 3.3. *Let $\mathcal{I}^*$ be a minimizing partition of the higher order Mumford-Shah problem* (3.5). *Then the corresponding minimal functional value satisfies*

$$\mathcal{G}'_{\mathcal{I}^*}(f) = \|f\|_2^2 - f^T S_{\mathcal{I}^*,\beta} f + \gamma \, |\mathcal{I}^*|.$$

Proof. Let $\tilde{u} = S_{\mathcal{I}^*,\beta} f$. Expanding the functional yields

$$\begin{aligned}
\mathcal{G}'_{\mathcal{I}^*}(f) &= \beta^{2k} \|L_{\mathcal{I}^*}\tilde{u}\|_2^2 + \|\tilde{u} - f\|_2^2 + \gamma \, |\mathcal{I}^*| \\
&= \beta^{2k} \tilde{u}^T L_{\mathcal{I}^*}^T L_{\mathcal{I}^*} \tilde{u} + (\tilde{u} - f)^T (\tilde{u} - f) + \gamma \, |\mathcal{I}^*| \\
&= \beta^{2k} \tilde{u}^T L_{\mathcal{I}^*}^T L_{\mathcal{I}^*} \tilde{u} + \tilde{u}^T (\tilde{u} - f) + f^T (\tilde{u} - f) + \gamma \, |\mathcal{I}^*| \\
&= \tilde{u}^T (\beta^{2k} L_{\mathcal{I}^*}^T L_{\mathcal{I}^*} \tilde{u} + (\tilde{u} - f)) - f^T (\tilde{u} - f) + \gamma \, |\mathcal{I}^*| \\
&= -f^T (\tilde{u} - f) + \gamma \, |\mathcal{I}^*| = \|f\|_2^2 - f^T \tilde{u} + \gamma \, |\mathcal{I}^*|,
\end{aligned}$$

where we used that a minimizer satisfies the normal equations (3.11) in the penultimate line. □

In the next lemma, we show the existence of an optimal partition $\mathcal{I}^*$ whose segments have length not smaller than k with at most one exception.

Lemma 3.4. *For any partition $\mathcal{I}$ there exists a partition $\mathcal{I}'$ such that all intervals $I' \in \mathcal{I}'$ have length greater or equal than k (except possibly the leftmost one) and*

$$\sum_{I' \in \mathcal{I}'} \mathcal{E}^{I'} \leq \sum_{I \in \mathcal{I}} \mathcal{E}^{I} \quad \text{and} \quad |\mathcal{I}'| \leq |\mathcal{I}|.$$

Consequently, the corresponding functional values satisfy $\mathcal{G}'_{\mathcal{I}'}(f) \leq \mathcal{G}'_{\mathcal{I}}(f)$.

Proof. The proof is based on the fact that intervals of length k have zero approximation error. First, we fix a partition $\mathcal{I}$ and denote by I its right-most segment with $|I| < k$ and by l its left boundary index. If $l = 1$, the assertion holds and $\mathcal{I}' = \mathcal{I}$. For $l \neq 1$, we move the element $l - 1$ from the left neighboring segment to I. We denote the partition modified in this way by $\mathcal{I}'$. (If the left neighboring segment becomes empty, we delete it from the partition.) Since the new segment $I \cup \{l - 1\}$ has at most k elements, we have $\mathcal{E}^{\{l-1\} \cup I} = 0$ for the corresponding approximation error. Furthermore, we have $|\mathcal{I}'| \leq |\mathcal{I}|$. After a finite number of repetitions of the above, we obtain a partition $\mathcal{I}''$ whose segments have length greater or equal to k, except possibly the leftmost segment. □

Higher order Potts models. Recall that the higher order Potts model ($\mathcal{P}_{k,\infty,\gamma}$) can be interpreted as the limit case of the higher order Mumford-Shah model for smoothing parameter $\beta \to \infty$. As a consequence, while the higher order Mumford-Shah model approximates the data on segments in a nonparametric way by k-th order smoothing splines, the higher order Potts model approximates the data on the segments in a parametric way by a polynomial of (maximum) degree $k-1$. Therefore, for a segment $I = l : r$, the approximation error is now given by

$$\mathcal{E}^I = \min_{\substack{v \text{ polynomial of} \\ \text{degree} \leq k-1 \text{ on } I}} \|v - f_I\|_2^2. \tag{3.14}$$

After plugging the constraint on the right-hand side of (3.14) in the functional, we obtain a least squares problem w.r.t. the k coefficients of the polynomial.

By producing piecewise polynomial solutions, higher order Potts models are more restrictive than higher order Mumford-Shah models. However, the piecewise polynomial prior is more robust to noise. In practice, only one model parameter, that is the complexity parameter γ, has to be determined.

As higher order Potts models correspond to the limit cases of higher order Mumford-Shah models, they have similar properties. Specifically, there is no approximation error on a segment if the data on the segment can be described by a polynomial of order $k-1$. In particular, an interval I with $|I| \leq k$ has always approximation error zero. Therefore, Lemma 3.4 is also valid for higher order Potts models.

3.2.3. Existence and uniqueness of minimizers

We show the existence of minimizers whose proof is straightforward.

Theorem 3.5. *The higher order Mumford-Shah ($\mathcal{P}_{k,\beta,\gamma}$) and the higher order Potts model ($\mathcal{P}_{k,\infty,\gamma}$) have a minimizer for each order $k \in \mathbb{N}$ and all model parameters $\gamma > 0$, $\beta \in (0, \infty]$.*

Proof. Let the partition $\mathcal{I}$ be fixed. Now the minimization amounts to least squares problems on the intervals in $\mathcal{I}$ which all possess a minimizer. Hence, as the number of partitions of the discrete domain Ω is finite, there exists at least one which minimizes the functional value. □

The uniqueness of minimizers is more cumbersome. The following example illustrates that the solutions of higher order Mumford-Shah models ($\mathcal{P}_{k,\beta,\gamma}$) do not have to be unique.

Example 3.6. *Consider data $f = (0,1,0)$ and $k = 2$. The optimal signal corresponding to the partition $\mathcal{I}^3 = \{(1),(2),(3)\}$ is given by $u^3 = f$ and the corresponding functional value equals 2γ. The optimal solution of a partition with two segments is given by $u^2 = f$ as well, and yields the lower functional value γ. The one-segment partition $\mathcal{I}^1 = \{(1,2,3)\}$ induces the signal $u^1 = \frac{1}{1+6\beta^4}(2\beta^4, 1+2\beta^4, 2\beta^4)^T$ and the functional value is $\frac{4\beta^4}{1+6\beta^4}$. For $\gamma = \frac{4\beta^4}{1+6\beta^4}$ or, equivalently, $\beta^4 = \gamma/(4-6\gamma)$, the functional values of the two-segment and the one-segment solutions are both minimal given that $\gamma < 2/3$. Hence, for each $\gamma < 2/3$ there exists $\beta > 0$ such that both the two-segment and the one-segment solutions are minimizers and that $u^1 \neq u^2$.*

For convenience, we considered the second order case $k = 2$ in Example 3.6. Analogous examples exist for any order $k \geq 3$. We will see next that configurations as in Example 3.6 are very unlikely. In preparation for proving this, we define a notion of equivalence of partitions.

Definition 3.7. We call two partitions $\mathcal{I}, \mathcal{J}$ equivalent, i.e.,

$$\mathcal{I} \sim \mathcal{J} \quad :\Leftrightarrow \quad (I \in \mathcal{I},\ |I| > k \Rightarrow I \in \mathcal{J} \quad \text{and} \quad J \in \mathcal{J},\ |J| > k \Rightarrow J \in \mathcal{J}), \tag{3.15}$$

if their intervals of minimum length $k + 1$ are the same. (The smaller intervals are irrelevant.)

Since the block matrices in (3.9) do not depend on intervals shorter than $k+1$, two equivalent partitions $\mathcal{I}, \mathcal{J}$ define the same block matrices $L_{\mathcal{I}} = L_{\mathcal{J}}$. Hence, by (3.12), their corresponding minimizers are equal, i.e.,

$$u_{f,\mathcal{I}} = S_{\mathcal{I},\beta} f = S_{\mathcal{J},\beta} f = u_{f,\mathcal{J}}. \tag{3.16}$$

In summary, each equivalence class of partitions $[\mathcal{I}]$ defines a unique restricted minimizer $u_{f,\mathcal{I}}$ and for each equivalence class $[\mathcal{I}]$, there is a unique matrix $L_{\mathcal{I}}$. The latter is even a one-to-one relationship. Altogether,

$$\begin{aligned} \text{the assignments} \quad & [\mathcal{I}] \to L_{\mathcal{I}},\ [\mathcal{I}] \to u_{f,\mathcal{I}} \quad \text{are well-defined,} \\ \text{and} \quad & [\mathcal{I}] \to L_{\mathcal{I}} \quad \text{is one-to-one.} \end{aligned} \tag{3.17}$$

The minimization problem (3.10) can now be recast in terms of equivalence classes $[\mathcal{I}]$ by

$$\operatorname*{argmin}_{u \in \mathbb{R}^N, [\mathcal{I}]} F_{[\mathcal{I}]}(u), \qquad \text{where} \qquad F_{[\mathcal{I}]}(u) = \beta^{2k} \, \|L_{\mathcal{I}} u\|_2^2 + \|u - f\|_2^2 + \gamma \, |[\mathcal{I}]|. \tag{3.18}$$

Here, the symbol $|[\mathcal{I}]|$ denotes the smallest cardinality among the members of $[\mathcal{I}]$, i.e., $|[\mathcal{I}]| = \min_{\mathcal{J} \in [\mathcal{I}]} |\mathcal{J}|$. By employing this notation, the functional (3.13) is well-defined w.r.t. the equivalence classes and we can write

$$\mathcal{G}_{[\mathcal{I}]}(f) = \beta^{2k} \, \|L_{\mathcal{I}} S_{\mathcal{I}} f\|_2^2 + \|S_{\mathcal{I}} f - f\|_2^2 + \gamma \, |[\mathcal{I}]|. \tag{3.19}$$

We obtain the following result on the uniqueness of minimizers.

Theorem 3.8. *Let $\gamma > 0$, $\beta \in (0, \infty]$, and $k \in \mathbb{N}$. The minimizer u^* of $(\mathcal{P}_{k,\beta,\gamma})$ is unique for almost all input data f.*

Proof. We infer from the notation introduced right above that the solution of $(\mathcal{P}_{k,\beta,\gamma})$ is unique for any data $f \in \mathcal{F}$, where the set $\mathcal{F} \subset \mathbb{R}^N$ is given by

$$\mathcal{F} = \left\{ f \in \mathbb{R}^N : \text{ there is a partition } \mathcal{I}^* \text{ with } F_{[\mathcal{I}^*]}(u_{f,\mathcal{I}^*}) < F_{[\mathcal{I}]}(u_{f,\mathcal{I}}) \text{ for all } \mathcal{I} \notin [\mathcal{I}^*] \right\}. \tag{3.20}$$

We will show that the complement $\mathcal{F}^C$ of $\mathcal{F}$ is a null set, that is, it has Lebesgue measure zero. The minimal function value constraint to $[\mathcal{I}]$ for data f is given by $\mathcal{G}_{[\mathcal{I}]}(f)$ as defined in (3.19). We note that the main part of (3.19) is a quadratic form, i.e., $\bar{\mathcal{G}}_{[\mathcal{I}]}(f) := \mathcal{G}_{[\mathcal{I}]}(f) - \gamma |[\mathcal{I}|]$ is a quadratic form in f. Hence, $\mathcal{F}^C$ is a subset of

$$\mathcal{H} = \left\{ f \in \mathbb{R}^N : \text{ there are } \mathcal{I}, \mathcal{I}' \text{ with } [\mathcal{I}] \neq [\mathcal{I}'] \text{ and } \bar{\mathcal{G}}_{[\mathcal{I}]}(f) - \bar{\mathcal{G}}_{[\mathcal{I}']}(f) = \gamma \left(|[\mathcal{I}']| - |[\mathcal{I}]| \right) \right\}. \tag{3.21}$$

It follows from (3.17) and $[\mathcal{I}] \neq [\mathcal{I}']$ that the quadratic form given by $f \mapsto \bar{\mathcal{G}}_{[\mathcal{I}]}(f) - \bar{\mathcal{G}}_{[\mathcal{I}']}(f)$ is nonzero. Thus, each of its level sets is a hypersurface of dimension $N-1$ whose N-dimensional Lebesgue measure is zero [64]. By forming the (finite) union over all $\mathcal{I}, \mathcal{I}'$, we obtain that $\mathcal{H}$ has Lebesgue measure zero and since $\mathcal{F}^C$ is a subset of $\mathcal{H}$, it has Lebesgue measure zero as well. □

3.2.4. Related models

Complexity-constrained models. Higher order Mumford-Shah models $(\mathcal{P}_{k,\beta,\gamma})$ control the complexity of the partition in terms of a (complexity) penalty. Another approach is to hard constrain the number of segments via

$$(u^*, \mathcal{I}^*) = \underset{u \in \mathbb{R}^N, |\mathcal{I}| \leq J}{\operatorname{argmin}} \|u - f\|_2^2 + \beta^{2k} \sum_{I \in \mathcal{I}} \|\nabla^k u_I\|_2^2. \tag{$C_{k,\beta,J}$}$$

The models $(\mathcal{P}_{k,\beta,\gamma})$ and their constrained counterparts $(C_{k,\beta,J})$ are closely related. We denote by $(u^J, \mathcal{I}^J)$ a solution of $(C_{k,\beta,J})$ constraint to at most J segments. Having determined all constrained solutions $(u^J, \mathcal{I}^J)$, $J = 1, \ldots, N$, an optimal solution of $(\mathcal{P}_{k,\beta,\gamma})$ can be found by simply choosing the constrained solution with smallest functional value $(\mathcal{P}_{k,\beta,\gamma})$:

$$J^* \in \underset{J=1,\ldots,N}{\operatorname{argmin}} \ \gamma J + \|u^J - f\|_2^2 + \beta^{2k} \sum_{I \in \mathcal{I}^J} \|\nabla^k u_I^J\|_2^2.$$

This relation was exploited in [20] to derive a solver for the first order problem $(\mathcal{P}_{1,\beta,\gamma})$.

As a useful consequence, the relation above allows us to compute minimizers of $(\mathcal{P}_{k,\beta,\gamma})$ for all $\gamma > 0$ simultaneously. More precisely, the set of solutions of $(C_{k,\beta,J})$ for all $J = 1, \ldots, N$, can be used to identify a finite number of intervals for γ on which the corresponding solution of $(\mathcal{P}_{k,\beta,\gamma})$ does not change.

ℓ_0-penalized problems. Similarly to the first order Mumford-Shah model (3.4), the first order Potts model $(\mathcal{P}_{1,\infty,\gamma})$ can be formulated in terms of the signal u only by employing the ℓ_0-"norm" of the vector of finite differences, i.e.,

$$u^* = \underset{u \in \mathbb{R}^N}{\operatorname{argmin}} \ \gamma \|\nabla u\|_0 + \|u - f\|_2^2. \tag{3.22}$$

Here, $\|v\|_0$ denotes the number of non-zero elements of a vector; that is $\|v\|_0 = |\{n : v_n \neq 0\}|$. In contrast to the first order Potts model, plugging ∇^k in (3.22) does *not* give an equivalent formulation of the higher order Potts model $(\mathcal{P}_{k,\infty,\gamma})$. In general, for $k \geq 2$ we have

$$\underset{w \in \mathbb{R}^N}{\operatorname{argmin}} \ \gamma \|\nabla^k w\|_0 + \|w - f\|_2^2, \neq u^*, \quad \text{where } u^* \text{ is the minimizer of } (\mathcal{P}_{k,\infty,\gamma}). \tag{3.23}$$

The following example illustrates the difference for $k = 2$.

Example 3.9. *Let $f = (-1, -1, 1, 1)$. The optimal partition with two elements is given by $\mathcal{I}^2 = \{(1,2),(3,4)\}$ and the corresponding optimal signal by $u_{\mathcal{I}^2} = f$ whose approximation errors* *are equal to* 0. *(Other partitions with two elements have a positive approximation error.) The best (linear) approximation on the one-segment partition $\mathcal{I}^1 = \{(1,2,3,4)\}$ is given by $u_{\mathcal{I}^1} = (-\frac{6}{5}, -\frac{2}{5}, \frac{2}{5}, \frac{6}{5})$ so that $\mathcal{E}^{(1:4)} = \frac{4}{5}$. The functional values are $\frac{4}{5}+\gamma$ for $(\mathcal{I}^1, u_{\mathcal{I}^1})$ and 2γ for $(\mathcal{I}^2, u_{\mathcal{I}^2})$. Thus, ($\mathcal{P}_{2,\infty,\gamma}$) has the solution $(\mathcal{I}^1, u_{\mathcal{I}^1})$ for $\gamma > \frac{4}{5}$, and $(\mathcal{I}^2, u_{\mathcal{I}^2})$ for $\gamma < \frac{4}{5}$. (For $\gamma = \frac{4}{5}$ they are both optimal.) In contrast, as $\|\nabla^2 u_{\mathcal{I}^1}\|_0 = 0$ and $\|\nabla^2 u_{\mathcal{I}^2}\|_0 = \|(2,-2)\|_0 = 2$, the critical value for the ℓ_0-model* (3.23) *is $\gamma = \frac{2}{5}$. Thus, the solutions of* (3.23) *for $k = 2$ and ($\mathcal{P}_{2,\infty,\gamma}$) are different for $\gamma \in (\frac{2}{5}, \frac{4}{5})$.*

The intuition behind this difference is that (3.23) penalizes the number of kinks of u for $k = 2$, whereas ($\mathcal{P}_{2,\infty,\gamma}$) penalizes the number of changes in the affine parameters of the piecewise affine-linear signal. The kink model, i.e., (3.23) for $k = 2$, was studied in [73].

3.3. Fast and stable solvers for higher order Mumford-Shah problems

In this section, we derive efficient and stable solvers for higher order Mumford-Shah and Potts problems ($\mathcal{P}_{k,\beta,\gamma}$) for any order $k \in \mathbb{N}$ and model parameters $\beta \in (0,\infty]$, $\gamma > 0$. The approach is based on a dynamic programming strategy which we briefly recall. It turns out that the key for an efficient algorithm is the computation of spline approximation errors for which we develop an efficient and stable recurrence scheme. Finally, we give an analysis on the stability of the proposed method.

3.3.1. Dynamic programming scheme for partitioning problems

We denote the functional in the partition-based formulation (3.6) of the higher order Mumford-Shah model by

$$P(\mathcal{I}) = \sum_{I \in \mathcal{I}} (\mathcal{E}^I + \gamma). \tag{3.24}$$

It is an important observation that the functional P is also well-defined on the reduced domain $1:r$ for the reduced data $f_{1:r}$ and in terms of a partition $\mathcal{I}$ of $1:r$. We denote the minimal functional value on $1:r$ by

$$P_r^* = \min_{\mathcal{I} \text{ partition on } 1:r} P(\mathcal{I}). \tag{3.25}$$

Next, we let $l:r$ be the rightmost interval in the partition $\mathcal{I}$ in (3.24) , i.e., $\mathcal{I} = \mathcal{J} \cup (l:r)$ for a partition $\mathcal{J}$ of the domain $1:l-1$. Thus, we can write $P(\mathcal{I}) = \sum_{I \in \mathcal{J}} (\mathcal{E}^I + \gamma) + (\mathcal{E}^{l:r} + \gamma)$. By taking the minimum we see that the value P_r^* for the domain $1:r$ satisfies the Bellman equation [13], i.e.,

$$P_r^* = \min_{l=1,\dots,r} \left\{ \mathcal{E}^{l:r} + \gamma + P_{l-1}^* \right\}, \tag{3.26}$$

where we let $P_0^* = -\gamma$; see also [79]. Recall that $\mathcal{E}^{1:r} = 0$ if the interval $1:r$ is shorter than k, i.e., if $r - l + 1 \leq k$. Hence, the minimum on the right hand side of (3.26) actually has to be taken only over the values $l = 1, \dots, r-k$ which are the candidates for the left boundary of an optimal rightmost interval. In view of the dynamic programming principle, we successively compute the optimal functional values for the reduced domains P_1^*, P_2^*, until we reach P_N^*. Since we are interested in an optimal partition rather than the functional values, we keep track of a corresponding optimal partition $\mathcal{I}^*$ by storing at step r the minimizing argument l' of (3.26) as the value L_r. Then L encodes the boundaries of an optimal partition $\mathcal{I}^*$ [79] by

$$\mathcal{I}^* = \Big\{\Big(L(r_i) : r_i\Big) \ : \ r_0 = N,\ r_{i+1} = L(r_i) - 1 \text{ for } i \geq 0 \text{ such that } r_i > 0\Big\}. \tag{3.27}$$

In order to solve (3.6) by the above procedure, we have to compute $O(N^2)$ approximation errors $\mathcal{E}^I$. Thus, if we denote by ϕ the upper bound for the complexity of computing $\mathcal{E}^I$, the procedure has $O(N^2\phi(N))$ worst time complexity. Using a standard solver for the least squares problem (3.7) results in $\phi(N) = N$ (the involved matrices have a band structure) and an overall $O(N^3)$ algorithm. We develop a strategy that achieves $\phi(N) = 1$ in the next section.

Pruning the search space. It has been observed that the search space for problems of type (3.26) can be pruned to speed up computations. We recall the two strategies developed in [181] and [121], respectively. The strategy in [181] exploits that the approximation errors satisfy $\mathcal{E}^{l:r} \leq \mathcal{E}^{l':r}$ if $l' \leq l$. Then it follows from (3.26) that if the current value P_r for P_r^* obeys

$$P_r < \mathcal{E}^{l:r} + \gamma \tag{3.28}$$

for some l, one can skip all $l' < l$ for this r and hence $P_r^* = P_r$. Therefore, we do not have to compute $\mathcal{E}^{l':r}$. The second strategy to prune the search space follows from the observation that the approximation errors satisfy the inequality $\mathcal{E}^{l:s} + \mathcal{E}^{s+1:r} \leq \mathcal{E}^{l:r}$, for all $l \leq s < r$. Killick et al. [121] showed that if

$$P_s^* \leq P_l^* + \mathcal{E}^{l+1:s} \tag{3.29}$$

holds, then l cannot be an optimal last changepoint at any future time point r. That means, the intervals $l+1:r$ for all $r = l+1, \ldots, N$ cannot be reached and consequently l can be omitted for any future time point r.

3.3.2. Fast computation of the approximation errors for higher order Mumford-Shah problems

As seen in the last section, the computational complexity for solving (3.6) by dynamic programming depends on the complexity ϕ of computing the approximation errors $\mathcal{E}^I$ needed in (3.26). In this section, we develop recurrence formulas for computing the $\mathcal{E}^I$ in $\phi(N) = O(1)$. For readability, we describe the basic scheme for the left bound $l = 1$, that is, computing $\mathcal{E}^{1:r}$ for $r = 1, \ldots, N$. The approach for general left interval borders $l > 1$ works analogously.

Recall that the approximation errors for intervals of length shorter or equal than k is always zero. Thus, we assume $r > k$ in the following. First, we rewrite the underlying least squares problem (3.7) for $I = 1 : N$ in matrix form

$$\mathcal{E}^{1:N} = \min_{v \in \mathbb{R}^N} \|Av - y\|_2^2. \tag{3.30}$$

Here, we denote the system matrix and the data vector by

$$A = \begin{pmatrix} E_N \\ \beta^k \nabla^k \end{pmatrix} \in \mathbb{R}^{(2N-k)\times N} \quad \text{and} \quad y = (f_1, \ldots, f_N, 0)^T \in \mathbb{R}^{2N-k}, \tag{3.31}$$

respectively. The identity matrix of dimension N is given by E_N. Consequently, computing $\mathcal{E}^{1:r}$ for $r < N$ amounts to solving the least squares problem of smaller size

$$\mathcal{E}^{1:r} = \min_{v \in R^r} \left\| A^{(r)} v - y^{(r)} \right\|_2^2, \tag{3.32}$$

where $A^{(r)}$ is the submatrix of A given by

$$A^{(r)} = \begin{pmatrix} A_{1:r,1:r} \\ A_{(N+1:N+r-k),(1:r)} \end{pmatrix}, \quad \text{and} \quad y^{(r)} = \begin{pmatrix} f_{1:r} \\ 0 \end{pmatrix}. \tag{3.33}$$

Note that it is not necessary to compute a minimizer v^* of (3.32) to assess $\mathcal{E}^{1:r}$. Rather, we derive a recurrence formula for computing $\mathcal{E}^{1:r}$ directly which is based on applying Givens rotations. To this end, we denote by $Q^{(r)}$ and $R^{(r)}$ the QR decomposition of the system matrix $A^{(r)}$, i.e., $R^{(r)}$ is an upper triangular matrix and $Q^{(r)}$ is an orthogonal matrix such that

$$A^{(r)} = Q^{(r)} \begin{pmatrix} R^{(r)} \\ 0 \end{pmatrix}.$$

After applying $(Q^{(r)})^T$ to (3.32), the approximation error $\mathcal{E}^{1:r}$ is given by

$$\begin{aligned} \mathcal{E}^{1:r} &= \min_{v \in \mathbb{R}^r} \left\| \begin{pmatrix} R^{(r)} \\ 0 \end{pmatrix} v - (Q^{(r)})^T y^{(r)} \right\|_2^2 \\ &= \min_{v \in \mathbb{R}^r} \| R^{(r)} v - ((Q^{(r)})^T y^{(r)})_{1:r} \|_2^2 + \| ((Q^{(r)})^T y^{(r)})_{r+1:2r-k} \|_2^2 \\ &= \| ((Q^{(r)})^T y^{(r)})_{r+1:2r-k} \|_2^2, \end{aligned} \tag{3.34}$$

where we used that the ℓ_2-norm is invariant under orthogonal transformations for the first equality and that the upper-triangular linear system induced by $R^{(r)}$ can be solved exactly for the last equality. Now the approximation errors are explicitly given by (3.34) and no minimization is involved. Our goal is to recursively compute $\mathcal{E}^{1:r+1}$ without explicitly computing QR decompositions and without carrying out the summation in the last line of (3.34).

As a first step, we derive the recurrence coefficients. We assume that we have computed the QR decomposition $Q^{(r)}, R^{(r)}$ of $A^{(r)}$ and explain how to obtain a QR decomposition $Q^{(r+1)}, R^{(r+1)}$ of $A^{(r+1)}$ from these data $Q^{(r)}, R^{(r)}$. Therefore, we define the auxiliary matrix $W^{(r)}$ given by

$$W^{(r)} = \begin{pmatrix} R^{(r)} & 0 \\ 0 & 1 \\ 0 & 0 \\ \multicolumn{2}{c}{A_{N+r-k+1,(1:r+1)}} \end{pmatrix}, \tag{3.35}$$

i.e., $W^{(r)}$ is given by the upper triangular matrix $R^{(r)}$ and the beginning of the $(N+r-k+1)$-th row of A. By the definition of ∇^k the matrix A has a band structure and only the last $k+1$ entries of $A_{N+r-k+1,(1:r+1)}$ are non-zero. We bring $W^{(r)}$ to upper triangular form without altering the already present zeros. To this end, we employ Givens rotations as a Givens rotation operates exclusively on two rows of $W^{(r)}$. Then, by the band structure of $W^{(r)}$ the present zeros are not destroyed. (We remark that Householder reflections would destroy the existing zero entries.)

Recall that the matrix $G = G(j, m, \theta)$, which represents a Givens rotation with rotation angle θ, is an identity matrix with the 2×2 submatrix $(G_{jj}, G_{jm}; G_{mj}, G_{mm})$ replaced by a planar rotation:

$$G(j, m, \theta) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos(\theta) & \cdots & \sin(\theta) & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin(\theta) & \cdots & \cos(\theta) & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}. \tag{3.36}$$

We apply the Givens rotation $G(j, m, \Theta_{mj})$ to eliminate the matrix entry $a = A_{mj}$ by the pivot element $b = A_{jj}$ for which the parameters are determined by

$$\cos(\Theta_{mj}) = b/\rho, \quad \sin(\Theta_{mj}) = a/\rho, \tag{3.37}$$

where $\rho = \operatorname{sign}(b)\sqrt{a^2 + b^2}$ and the corresponding rotation angle is denoted by Θ_{mj}. An important observation is that a Givens rotation $G(j, m, \theta)$ operates exclusively on the j-th and m-th row of a matrix and does not destroy the zeros present in other rows. Therefore, we employ $k + 1$ Givens rotations with parameters chosen according to (3.37) to eliminate the last row of $W^{(r)}$ which yields $R^{(r+1)}$. In this manner, we can recursively compute $R^{(r+1)}$ from $R^{(r)}$, where the rotation angles Θ_{mj} are the recurrence coefficients.

If the coefficients Θ_{mj} have been computed, the error update step $\mathcal{E}^{1:r} \to \mathcal{E}^{1:r+1}$ can be performed in $O(1)$. We assume that we have computed $\mathcal{E}^{1:r}$ and let the vector $q^{(r)}$ be given by

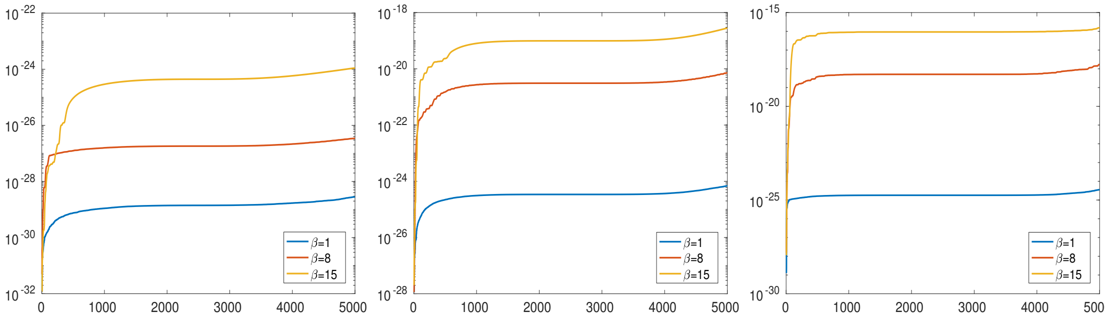
$$q^{(r)} = (Q^{(r)})^T y^{(r)}. \tag{3.38}$$

The vectors $q^{(r)}$ satisfy the recurrence relation

$$q^{(r+1)} = G^{(r+1)} \begin{pmatrix} q^{(r)}_{1:r} \\ f_{r+1} \\ q^{(r)}_{r+1:2r-k} \\ 0 \end{pmatrix}. \tag{3.39}$$

Here, $G^{(r+1)}$ denotes the composition of the above $k + 1$ Givens rotations; that is, it is the elimination matrix given by

$$G^{(r+1)} = \prod_{j=1}^{k+1} G\Big(r - k + j, 2(r + 1) - k, \Theta_{N+r+1-k,r-k+j}\Big), \tag{3.40}$$



(a) (b) (c)

Figure 3.3.: Approximation errors $\mathcal{E}^{1:r}$, $r = 1, \ldots, N$, for (a) a linear polynomial and order $k = 2$, for (b) a quadratic polynomial and order $k = 3$, and for (c) a cubic polynomial and order $k = 4$. The theoretical approximation errors are equal to zero; the approximation errors computed using the recurrence formula (3.42) are accurate up to machine precision.

where we use the convention $\prod_{j=1}^{k} Z_j = Z_k Z_{k-1} \cdots Z_1$. As $G^{(r+1)}$ only operates on the first $r+1$ rows and the last row of the vector on the right hand side of (3.39), we obtain

$$\left\| q^{(r+1)}_{r+1:2r+1-k} \right\|_2^2 = \left\| q^{(r)}_{r+1:2r-k} \right\|_2^2 + \left| q^{(r+1)}_{2(r+1)-k} \right|^2. \tag{3.41}$$

Thus, in view of (3.34), the error update is given by

$$\mathcal{E}^{1:r+1} = \mathcal{E}^{1:r} + \left| q^{(r+1)}_{2(r+1)-k} \right|^2. \tag{3.42}$$

To sum up, the update scheme needs to compute $q^{(k+1)}$ by (3.39) and update $\mathcal{E}^{1:r+1}$ by (3.42).

The update scheme for general left interval borders l works analogously. The above procedure is applied to the data $(f_l, \ldots f_r)$ using the same recurrence coefficients Θ_{mj} since they do not depend on the data. In particular, we have to compute the $(N-k)(k+1)$ coefficients Θ_{mj} only once and use them for computing all $\mathcal{E}^{l:r}$.

We briefly discuss the accuracy of the error update scheme (3.42). As Givens rotations are orthogonal, they have the optimal condition number one. Hence, there is no inherent error amplification in the elimination steps. The practical accuracy of the error update is illustrated by the following numerical experiment. We compute the approximation errors of a polynomial of degree $k-1$. As these are in the null space of ∇^k, the approximation errors $\mathcal{E}^{1:r}$ are exactly zero for all $r = 1, \ldots, N$. Figure 3.3 shows that the proposed scheme reproduces the exact results up to machine precision.

Incorporating the pruning strategies. We elaborate on how to incorporate the two strategies discussed in Section 3.3.1. To effectively apply the first strategy which checks (3.28), we have to run over the l-index in the order $l = r, r-1, \ldots, 1$. The pruning condition of the second strategy (3.29) can be checked when P_r^* was computed. If the first condition (3.28) holds for some l for the subdata $1:r$, then the second condition (3.29) cannot be checked for $l' < l$ since $\mathcal{E}^{l':r}$ has not been computed yet. In order to use both strategies effectively, we proceed as follows. At domain $1:r$, we run through all l stored in the list L (sorted in descending order) and update the corresponding approximation errors until $\mathcal{E}^{l:r}$ was reached. After each update step, we check condition (3.29) for the current upper interval bound of the error and if so, delete l from L and continue with the next entry in L. As a consequence, it is not necessary to adapt the first pruning strategy essentially: we check condition (3.28) after testing if l is the current optimal last changepoint. Combining the pruning strategies decreases the total number of error updates (3.42) that have to be performed; see Section 3.4.2 for a numerical study.

Algorithm 3.1 provides a pseudocode of the proposed scheme. We summarize the above derivation in the following theorem.

Theorem 3.10. *Let* $f \in \mathbb{R}^N$, $k \in \mathbb{N}$, *and* $\beta, \gamma > 0$. *The proposed algorithm computes a global minimizer of* $(\mathcal{P}_{k,\beta,\gamma})$. *The worst case time complexity is* $\mathcal{O}(N^2)$.

Proof. Recall that the considered problem (3.5) satisfies the Bellman equation (3.26) and decomposes into the according subproblems on the reduced domains $1 : r$. Algorithm 3.1 successively computes the required optimal values P_r^* of the subproblems in the Bellman equation (3.26) for $r = 1, 2, \ldots$ until P_N^* is reached and keeps track of an optimal partition by saving the minimizing argument l' for each P_r^*. Each P_r^* on the other hand is computed by iterating through the arguments $l = 1, \ldots, r$ in the Bellman equation of the corresponding subproblem in order to find the optimal value and argument. Thus, Algorithm 3.1 computes a global minimizer of problem (3.5). Regarding the computational complexity, we have to check for the costs of computing the approximation errors $\mathcal{E}^{l:r}$ in the Bellman equations and for the costs of recovering the optimal signal after the optimal partition has been determined. Concerning the recovery of the optimal signal, we note that it corresponds to solving a least squares system of band matrices with at most $2N - k$ rows each having at most $k + 1$ entries. Hence, this step has linear costs in N. Regarding the computation of the approximation errors, we first have to compute the rotation angles of the Givens rotations (i.e., the recurrence coefficients Θ) which bring the system matrix A^N (see (3.31)) to upper triangular form. The elimination of a single row of A^N amounts to eliminating at most $k + 1$ entries in view of the band structure of A^N. The elimination of a single entry by a Givens rotation only creates nonzeros above the diagonal A_{ii}^N. Hence, the previously eliminated rows are preserved. Since applying a Givens rotation to the band matrix A^N has constant computational costs, this step has linear computational costs in N. By using the recurrence (3.39), (3.42) together with the recurrence coefficients given by the already computed Givens rotation angles, the computation of a single approximation error $\mathcal{E}^{l:r}$ has constant computational costs because applying a Givens rotation to a vector has constant costs. Since there are quadratically many approximation errors, the computational costs for computing all approximation errors is quadratic in N. Thus, the overall worst case time complexity of Algorithm 3.1 is $\mathcal{O}(N^2)$. □

3.3.3. Fast computation of the approximation errors for higher order Potts problems

We derive a fast and stable scheme to compute the approximation errors for higher order Potts problems (3.14). We start by rewriting the approximation errors (3.14) as the optimal value of

Algorithm 3.1: Solver for higher order Mumford-Shah and higher order Potts problems

```
   Input: Data $f \in \mathbb{R}^N$; model parameters $k \in \mathbb{N}, \beta \in (0, \infty], \gamma > 0$;
   Output: Global minimizer $(u^*, \mathcal{I}^*)$ of $(\mathcal{P}_{k,\beta,\gamma})$ or $(\mathcal{P}_{k,\infty,\gamma})$
   /* Precomputations                                                                    */
1  Row-wise transform the matrix $A$ from (3.31) for $(\mathcal{P}_{k,\beta,\gamma})$ or $B$ from (3.44) for $(\mathcal{P}_{k,\infty,\gamma})$ to upper
     triangular form using successive Givens rotations and store the rotation angles in $\Theta$.
2  Compute $\mathcal{E}^{1:r}$ for all $r = 1, \ldots, N$ with $\Theta$
   /* Find optimal changepoints                                                             */
3  Initialize lists $L = [2], R = [2], E = [0]$
4  $J_1 \leftarrow 0, P_1^* \leftarrow 0$
5  for $r \leftarrow 2, \ldots, N$ do
       /* Initialization                                                                    */
6      $J_r \leftarrow 0, P_r^* \leftarrow \mathcal{E}^{1:r}$
       /* Find optimal $P_r^*$ using (3.26)                                                 */
7      for $i = 1, \ldots,$ length of $L$ do
8          while $r_i < r$ do
               /* Update approximation error                                                */
9              $\begin{cases} (\mathcal{P}_{k,\beta,\gamma}): \text{ Compute } \mathcal{E}^{l_i:r_i+1} \text{ from } \mathcal{E}^{l_i:r_i} \text{ using the recurrence (3.39)-(3.42)} \\ (\mathcal{P}_{k,\infty,\gamma}): \text{ Compute } \mathcal{E}^{l_i:r_i+1} \text{ from } \mathcal{E}^{l_i:r_i} \text{ using the recurrence (3.47)-(3.49)} \end{cases}$
10             $E_i \leftarrow \mathcal{E}^{l_i:r_i+1}, \; r_i \leftarrow r_i + 1$
               /* Pruning (3.29)                                                            */
11             if $P_{l_i-1}^* + E_i \geq P_{r_i}^*$ then
12                 Delete: $l_i$ from $L$, $r_i$ from $R$ and $E_i$ from $E$
13                 go to 7
14             end
15         end
16         $b \leftarrow P_{l_i-1}^* + \gamma + E_i$
17         if $b \leq P_r^*$ then
18             $P_r^* \leftarrow b, J_r \leftarrow l - 1$
19         end
           /* Pruning (3.28)                                                                */
20         If $E_i + \gamma > P_r^*$ then break end
21     end
       /* Update lists                                                                      */
22     Prepend: $r + 1$ to $L$, $r + 1$ to $R$, 0 to $E$
23 end
   /* Recover optimal partition $\mathcal{I}^*$ from segment boundary locations $J$          */
24 $r \leftarrow N, \mathcal{I}^* \leftarrow \emptyset$
25 while $r > 0$ do
26     $l \leftarrow J(r) + 1, \mathcal{I}^* \leftarrow \mathcal{I}^* \cup \{(l : r)\}, r \leftarrow l - 1$
27 end
   /* Reconstruction of $u^*$ by solving the least squares problems on segments (using $\Theta$ for speedup)   */
28 for $I \in \mathcal{I}^*$ do
29     $\begin{cases} (\mathcal{P}_{k,\beta,\gamma}): \text{ Solve } u_I^* = \operatorname{argmin}_{v \in \mathbb{R}^{|I|}} \|v - f_I\|_2^2 + \beta^{2k} \|\nabla^k v\|_2^2 \\ (\mathcal{P}_{k,\infty,\gamma}): \text{ Solve } u_I^* = \operatorname{argmin} \|v - f_I\|_2^2 \text{ such that } v \text{ is polynomial of degree} \leq k - 1 \text{ on } I \end{cases}$
30 end
```

the least squares problem in terms of the polynomial coefficients $p \in \mathbb{R}^k$; it is given by

$$\mathcal{E}^{l:r} = \min_{p \in \mathbb{R}^k} \|B_{l:r,1:r} p - f_{l:r}\|_2^2, \tag{3.43}$$

where B is the $\mathbb{R}^{N\times k}$ system matrix defined by

$$B = \begin{pmatrix} 1 & 1 & \cdots & 1^{k-1} \\ 1 & 2 & \cdots & 2^{k-1} \\ \vdots & \vdots & & \vdots \\ 1 & N-1 & \cdots & (N-1)^{k-1} \\ 1 & N & \cdots & N^{k-1} \end{pmatrix} \in \mathbb{R}^{N\times k}. \tag{3.44}$$

Analogous to Section 3.3.2, we describe the method for the prototypical case $l = 1$. Moreover, we assume that $r > k$ since otherwise $\mathcal{E}^{1:r} = 0$. We denote the submatrix $B_{1:r,1:k}$ by $B^{(r)}$ and its QR decomposition by $Q^{(r)}, R^{(r)}$. Similarly to (3.34), we obtain for the approximation errors

$$\mathcal{E}^{1:r} = \min_{p \in \mathbb{R}^k} \left\| \begin{pmatrix} R^{(r)} \\ 0 \end{pmatrix} p - (Q^{(r)})^T f_{1:r} \right\|_2^2 = \|q^{(r)}_{k+1:r}\|_2^2, \tag{3.45}$$

where $q^{(r)}$ is given by

$$q^{(r)} = (Q^{(r)})^T f_{1:r}. \tag{3.46}$$

In analogy to higher order Mumford-Shah models, the recurrence coefficients for the error updates $\Theta_{r+1,j}$, $j = 1, \ldots, k$ are given by the Givens rotation angles for eliminating the entry $B_{r+1,j}$ of the system matrix B with the pivot element $R^{(r)}_{j,j}$. Suppose that we have computed $q^{(r)}$ and $\mathcal{E}^{1:r}$. Then –in accordance with (3.39)– $q^{(r+1)}$ can be expressed by the recurrence relation

$$q^{(r+1)} = G^{(r+1)} \begin{pmatrix} q^{(r)} \\ f_{r+1} \end{pmatrix}, \tag{3.47}$$

where $G^{(r+1)}$ is the composition of the Givens rotations $G(j, r+1, \Theta_{r+1,j})$ for $j = 1, \ldots, k$; i.e.,

$$G^{(r+1)} = \prod_{j=1}^{k} G(j, r+1, \Theta_{r+1,j}). \tag{3.48}$$

Here, we again use the convention $\prod_{j=1}^k Z_j = Z_k Z_{k-1} \cdots Z_1$. As $G^{(r+1)}$ operates exclusively on the first k entries and the last entry of $q^{(k)}$, it follows from (3.45) that

$$\mathcal{E}^{1:r+1} = \|q^{(r+1)}_{k+1:r+1}\|_2^2 = \|q^{(r)}_{k+1:r}\|_2^2 + (q^{(r+1)})^2_{r+1} = \mathcal{E}^{1:r} + (q^{(r+1)})^2_{r+1}. \tag{3.49}$$

Numerical comparison with the precomputed moment method. There exist closed formulas for the approximation errors $\mathcal{E}^{l:r}$ in higher order Potts problems. For the first and second order Potts problem such formulas were derived in [79, 135]. We derived corresponding formulas for the orders $k = 3$ and $k = 4$ with computer algebra. By using these formulas and precomputing the moments of the data, each of the errors $\mathcal{E}^{l:r}$ can be computed in $O(1)$. Typically, this

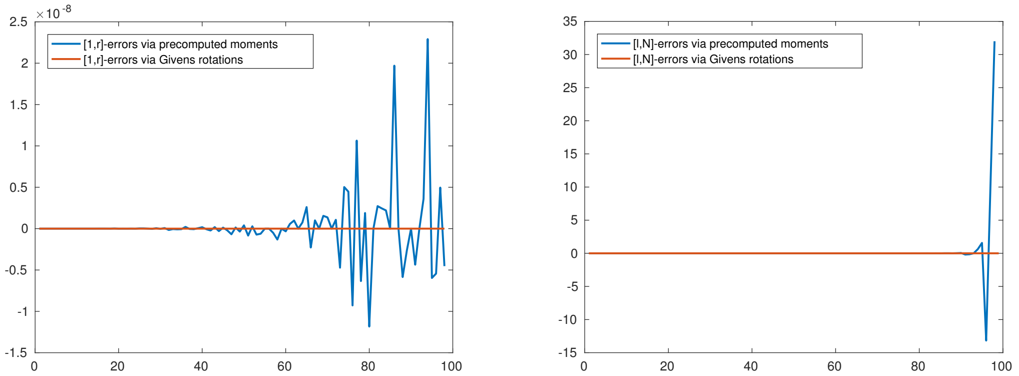


Figure 3.4.: Computation of the approximation errors for a parabolic signal for the higher order Potts model of order $k = 3$. The graphs show the results based on precomputed moments (blue) and based on the proposed scheme (red). The true approximation errors are all equal to zero. *Left:* Errors $\mathcal{E}^{1:r}$ for $r = 1, ..., N$. The computation based on precomputed moments is distorted beyond machine precision and gives even negative values. *Right:* Errors $\mathcal{E}^{l:N}$ for $l = 1, ..., N-1$. The values derived from precomputed moments are strongly distorted when l approaches N. The proposed computation based on Givens rotations is accurate up to machine precision in either case.

method yields acceptable results for the piecewise constant and piecewise affine-linear Potts model ($k = 1, 2$) and moderate signal length. However, for $k > 2$ or longer signals, the approach becomes numerically unstable. To illustrate this, we consider the following experiment (cf. Figure 3.4). We assume a parabolic signal $f_n = n^2/100$, $n = 0, \ldots, N$ with $N = 100$. For $k = 3$, the true approximation errors $\mathcal{E}^{l:r}$ for the higher order Potts model are zero since the data follows the model. Figure 3.4 shows that the results for $\mathcal{E}^{1:r}$ are distorted, when using the approach based on precomputed moments. This becomes particularly severe when r approaches N. The errors $\mathcal{E}^{l:N}$ are even more severely affected because of loss of significance. In contrast to the precomputation of moments approach, the proposed method gives accurate results up to machine precision.

Remark 3.11. In Section 3.3.2 and this section, we derived efficient update schemes for the approximation errors of higher order Mumford-Shah and Potts models as they are needed in Algorithm 3.1. Recall that the approximation errors are given by the sums of squared residuals of the least squares systems (3.32) w.r.t. the system matrix A and (3.43) w.r.t the system matrix B, respectively. The proposed update scheme uses Givens rotations to update the QR decomposition of the system matrix (and accordingly the approximation errors) as they preserve already present zeros, or, in other words, using Givens rotations allows us to exploit the band structure of A and the tallness of B , respectively. In principle, the proposed scheme can be used to approach any partitioning problem for which the approximation error on each interval is given by sums of residuals of least squares systems whose system matrices either have a band structure or are tall.

3.3.4. Stability results

In this section, we study the stability of the proposed algorithm. We begin with some basic lemmas which will be needed later on. In the following, we consider the functional $\mathcal{G}_{[\mathcal{I}]}(f)$ defined in (3.19). For convenience, we omit the brackets and simply write $\mathcal{G}_{\mathcal{I}}(f)$.

Lemma 3.12. *We consider data $f \in \mathbb{R}^N$. If there is a partition $\mathcal{I}'$ and $\varepsilon > 0$ such that*

$$\mathcal{G}_{\mathcal{I}'}(f) < \mathcal{G}_{\mathcal{I}}(f) - \varepsilon \quad \text{for all} \quad \mathcal{I} \notin [\mathcal{I}'], \tag{3.50}$$

then there is an Euclidean ball $B(f, \delta)$ around f such that for any data $g \in B(f, \delta)$ holds: there is a unique optimal solution $u = u_{g,\mathcal{I}^*}$ of the problem $(\mathcal{P}_{k,\beta,\gamma})$ with data g, and the corresponding partition $\mathcal{I}^*$ fulfills $[\mathcal{I}^*] = [\mathcal{I}']$. We may choose δ by

$$\delta := \min\left(\frac{1}{2}, \frac{\varepsilon}{2((2\beta)^{2k} + 1)\,(2\|f\| + 1/2)}\right). \tag{3.51}$$

Proof. The essential argument here relies on the continuity of the quadratic forms $x \mapsto \bar{\mathcal{G}}_{\mathcal{I}}(x)$ given by

$$\bar{\mathcal{G}}_{\mathcal{I}}(f) := \beta^{2k}\, \|L_{\mathcal{I}} S_{\mathcal{I}} f\|_2^2 + \|S_{\mathcal{I}} f - f\|_2^2 \tag{3.52}$$

which are the main parts of $\mathcal{G}_{\mathcal{I}}$ given by (3.19). Each $\bar{\mathcal{G}}_{\mathcal{I}}$ may be represented w.r.t. the Euclidean standard scalar product $\langle \cdot, \cdot \rangle$ via a symmetric matrix $A_{\mathcal{I}}$ as $\bar{\mathcal{G}}_{\mathcal{I}}(x) = \langle A_{\mathcal{I}} x, x \rangle$. The operator norm of $A_{\mathcal{I}}$ equals the norm of the corresponding bilinear form which in turn, since the $\bar{\mathcal{G}}_{\mathcal{I}}$ are positive (semi-definite), corresponds to

$$\|A_{\mathcal{I}}\| = \sup_{x : \|x\| = 1} \bar{\mathcal{G}}_{\mathcal{I}}(x). \tag{3.53}$$

We first let δ' be defined by

$$\delta' := \min\left(\frac{1}{2}, \frac{\varepsilon}{2 \max\limits_{\mathcal{J}} \|A_{\mathcal{J}}\|\,(2\|f\| + 1/2)}\right). \tag{3.54}$$

We want to estimate $\mathcal{G}_{\mathcal{I}'}(f)$ from above for g in a δ'-ball around f. For brevity, we write $\mathcal{G}_{\mathcal{I}'}(g) = \bar{\mathcal{G}}_{\mathcal{I}'}(g) + \gamma N_{\mathcal{I}'}$, where we let $N_{\mathcal{I}'} := |[\mathcal{I}']|$. Then we may estimate for $\mathcal{I} \notin [\mathcal{I}']$,

$$\begin{aligned}
\mathcal{G}_{\mathcal{I}'}(g) = \bar{\mathcal{G}}_{\mathcal{I}'}(g) + \gamma|[\mathcal{I}']| &= \bar{\mathcal{G}}_{\mathcal{I}'}(f) + \bar{\mathcal{G}}_{\mathcal{I}'}(f-g) - 2\langle A_{\mathcal{I}'} f, f - g\rangle + \gamma|[\mathcal{I}']| \\
&< \bar{\mathcal{G}}_{\mathcal{I}}(f) - \varepsilon + \delta'^2\, \|A_{\mathcal{I}'}\| + 2\delta'\, \|A_{\mathcal{I}'}\|\, \|f\| + \gamma|[\mathcal{I}]| \\
&= \bar{\mathcal{G}}_{\mathcal{I}}(g) + \bar{\mathcal{G}}_{\mathcal{I}}(f-g) + 2\langle A_{\mathcal{I}} f, f - g\rangle \\
&\quad - \varepsilon + \delta'^2\, \|A_{\mathcal{I}'}\| + 2\delta'\, \|A_{\mathcal{I}'}\|\, \|f\| + \gamma|[\mathcal{I}]| \\
&\le \bar{\mathcal{G}}_{\mathcal{I}}(g) - \varepsilon + 2\delta' \max_{\mathcal{J}} \|A_{\mathcal{J}}\|\ \ (2\|f\| + \delta') + \gamma|[\mathcal{I}]| \\
&\le \bar{\mathcal{G}}_{\mathcal{I}}(g) - \varepsilon + \varepsilon + \gamma|[\mathcal{I}]| = \bar{\mathcal{G}}_{\mathcal{I}}(g) + \gamma|[\mathcal{I}]| = \mathcal{G}_{\mathcal{I}}(g).
\end{aligned} \tag{3.55}$$

For the first inequality, we applied (3.50) for $\mathcal{G}_{\mathcal{I}'}(f)$ and used the assumption that $g \in B(f, \delta')$. For the second inequality, we used that $\|A_{\mathcal{I}'}\|$ and $\|A_{\mathcal{I}}\|$ are clearly bounded by $\max_{\mathcal{J}} \|A_{\mathcal{J}}\|$. For the last inequality, we employed (3.54). In order to relate (3.54) with (3.51), we now estimate $\max_{\mathcal{J}} \|A_{\mathcal{J}}\|$ using basic spectral theory for self-adjoint bounded operators. Since $A_{\mathcal{J}}$ is the matrix representing the bilinear form corresponding to $\bar{\mathcal{G}}_{\mathcal{J}}$, we may estimate using (3.52) that

$$\|A_{\mathcal{J}}\| \leq \beta^{2k} \|L_{\mathcal{J}}^T L_{\mathcal{J}}\| \, \|S_{\mathcal{J}}\|^2 + \|\mathrm{id} - S_{\mathcal{J}}\|^2, \qquad \text{for any partition } \mathcal{J}, \tag{3.56}$$

with the definitions of $L_{\mathcal{J}}$ given in (3.9) and that of $S_{\mathcal{J}}$ given in (3.12); here we only employed the triangle inequality, the submultiplicativity of operator norms and that $\|L_{\mathcal{J}}^T L_{\mathcal{I}}\| = \|L_{\mathcal{J}}\|^2$ holds. By (3.12), $S_{\mathcal{J}} = (\beta^{2k} L_{\mathcal{J}}^T L_{\mathcal{J}} + \mathrm{id})^{-1}$. Since $L_{\mathcal{J}}^T L_{\mathcal{I}}$ is self-adjoint and positive, the spectrum of $\beta^{2k} L_{\mathcal{J}}^T L_{\mathcal{J}} + \mathrm{id}$ is contained in $[1, \infty)$. Hence, its inverse $S_{\mathcal{I}}$ has its spectrum contained in $(0, 1]$. Being again self-adjoint, and positive, $\|S_{\mathcal{I}}\| \leq 1$. Further, since $S_{\mathcal{I}}$ has its spectrum contained in $(0, 1]$, $\mathrm{id} - S_{\mathcal{J}}$ has its spectrum contained in $(0, 1]$ as well. Then, with the same argument, $\|\mathrm{id} - S_{\mathcal{J}}\| \leq 1$. In order to estimate $L_{\mathcal{J}}^T L_{\mathcal{J}}$, we consider $L_{\mathcal{J}}$ in (3.9), and notice that $L_{\mathcal{J}}^T L_{\mathcal{J}}$ is block diagonal with entries consisting of convolutions of k-th differences with themselves. Thus the row-sums of the absolute values as well as the column sums of the absolute values of $L_{\mathcal{J}}^T L_{\mathcal{J}}$ are bounded by 2^k. By using the Schur criterion, the operator norm of $L_{\mathcal{J}}^T L_{\mathcal{J}}$ w.r.t. Euclidean norm in the base space can then be estimated by 2^{2k}. Summing up, we conclude invoking these estimates in (3.56) that

$$\|A_{\mathcal{J}}\| \leq \beta^{2k} 2^{2k} + 1, \qquad \text{for any partition } \mathcal{J}. \tag{3.57}$$

We gathered all ingredients to prove the lemma. If $g \in B(f, \delta)$, then $g \in B(f, \delta')$, by the estimate (3.57) relating (3.54) with (3.51). In consequence, the estimate (3.55) applies to g. Hence the solution for g is unique and given by (3.12); in particular, the corresponding equivalence class of partitions equals $\mathcal{I}'$ which shows the assertion. □

The next lemma is a stability result for the QR algorithm from [84]. We adapt it to our setup as we need it later on. Similarly to (3.12), we denote by S_I the linear mapping from f_I (the data restricted to the interval I) to the solution u_I.

Theorem 3.13. *The QR algorithm $\tilde{S}_I$ for computing the approximation errors $\mathcal{E}^I$ is backward stable, i.e., given data f_I on the subinterval $I \subset \{1, \dots, N\}$, there is a perturbation $\tilde{f}_I$ of the data f_I such that*

$$\tilde{S}_I(f_I) = S_I(\tilde{f}_I) \quad \text{with} \quad \|\tilde{f}_I - f_I\| \leq \delta_I, \tag{3.58}$$

where δ_I depends on the machine precision τ and on the norm of the data on the subinterval $\|f_I\|$ via

$$\delta_I(\tau) \leq 6\tau\sqrt{|I|}\left(\frac{9|I| - 5}{4} - k\right)(1 + 6\tau)^{3(|I|-1)-k} \|f_I\|. \tag{3.59}$$

Here, the QR algorithm is understood as in the analysis setup of [84, 205].

Proof. If $|I| \leq k$, then $\tilde{S}_I(f_I) = S_I(f_I) = 0$. Thus, we assume $|I| > k$. We recall that computing the approximation errors $\mathcal{E}^I$ corresponds to computing the residual vector of the least squares problem with model matrix $A \in \mathbb{R}^{2|I|-k\times|I|}$ from (3.31) and data vector f_I. In [84], it was shown that

$$\tilde{R} = \bar{Q}^T(A + \Delta A), \quad \|\Delta A\|_F \leq \mu_I(\tau)\|A\|_F, \tag{3.60}$$

for $\mu_I(\tau)$ given by

$$\mu_I(\tau) = 6\tau\sqrt{|I|} \cdot \left(\frac{9|I|-5}{4} - k\right)(1+6\tau)^{3(|I|-1)-k}, \tag{3.61}$$

where $\tilde{R}$ is the computed upper triangular matrix by means of Givens rotations and $\bar{Q}^T$ is the orthogonal matrix given by the composition of the exact Givens rotations we want to apply. Analogously, for transforming the data vector it holds

$$\widetilde{Q^T f_I} = \bar{Q}^T(f_I + \Delta f_I), \quad \|\Delta f_I\| \leq \mu_I(\tau)\|f_I\|,$$

and therefore $\tilde{S}_I(f_I) = S_I(f_I + \Delta f_I)$ which implies (3.59). ☐

Corollary 3.14. *We consider bounded data* $f \in \mathbb{R}^N$, $\|f\| < C$. *For any partition* $\mathcal{I}$ *considered in the proposed algorithm for the higher order Mumford-Shah and Potts problem, there is a perturbation* $\tilde{f}$ *of* f *such that*

$$\tilde{S}_{\mathcal{I}}(f) = S_{\mathcal{I}}(\tilde{f}) \quad \textit{where} \quad \|\tilde{f} - f\| < \delta_{\mathcal{I}}(\tau), \tag{3.62}$$

where the dependency on the machine precision is given by $\delta_{\mathcal{I}}(\tau)^2 = \sum_i \delta_{I_i}(\tau)^2$. *It depends on the estimation on intervals* $\delta_{I_i}(\tau)$ *given in Theorem 3.13 and on* C, *but not on the data* f. *Concretely,* $\delta_{\mathcal{I}}(\tau)$ *can be estimated from above by*

$$\delta_{\mathcal{I}}(\tau)^2 \leq 36C^2\tau^2 \sum_i |I_i| \left(\frac{9|I_i|-5}{4} - k\right)^2 (1+6\tau)^{6(|I_i|-1)-2k}. \tag{3.63}$$

Proof. The statement follows from Theorem 3.13. For a fixed partition, the proposed algorithm computes optimal solutions $u_{f,\mathcal{I}} = S_{\mathcal{I},\beta} f$ (cf. (3.12)) by using the QR algorithm with Givens rotations for each interval $I \in \mathcal{I}$. Thus, (3.63) is a consequence of the estimation on intervals given in (3.59). ☐

In the next statement, we denote by $\tilde{\mathcal{G}}_{\mathcal{I}}$ the algorithm to compute the energy $\mathcal{G}_{\mathcal{I}}$ given by (3.19). The bound on the approximation error between $\tilde{\mathcal{G}}_{\mathcal{I}}$ and $\mathcal{G}_{\mathcal{I}}$ for all $\mathcal{I}$ is denoted by $g(\tau)$ w.r.t. the precision τ.

Proposition 3.15. *We consider bounded data* $f \in \mathbb{R}^N$, $\|f\| < C$, *and assume that* (3.50) *is fulfilled for* f. *Let*

$$\delta^*(\tau) = \max_{\mathcal{I}} \delta_{\mathcal{I}}(\tau) \leq 6C\tau N^{\frac{3}{2}} \left(\frac{9N-5}{4} - k\right)(1+6\tau)^{3(N-1)-k} \tag{3.64}$$

for $\delta_{\mathcal{I}}(\tau)$ *in Corollary 3.14 and assume that* τ *is small enough such that* $\delta^*(\tau) < \delta/2$ *for* δ *given by* (3.51) *and such that* $g(\tau) \leq \varepsilon/4$ *with* ε *given in* (3.50). *Then, the higher order Potts and Mumford-Shah problem* $(\mathcal{P}_{k,\beta,\gamma})$ *has a unique minimizer* u_f, *and the proposed algorithm for computing this* *minimizer of the higher order Potts and Mumford-Shah problem ($\mathcal{P}_{k,\beta,\gamma}$) is backward stable in the sense that*

$$\tilde{u}_f = u_{\tilde{f}} \quad \text{where} \quad \|\tilde{f} - f\| < \delta^*(\tau). \tag{3.65}$$

Here, $\tilde{u}_f$ is the result produced by the proposed algorithm for data f and $u_{\tilde{f}}$ is the (unique) solution of the higher order Potts and Mumford-Shah problem ($\mathcal{P}_{k,\beta,\gamma}$) for perturbed data $\tilde{f}$.

Remark 3.16. A more explicit relation between ε and the precision τ (without using the δ's sufficient for the assumptions of Proposition 3.15 to hold) is given by

$$(1+6\tau)^{3N-k}\tau < \frac{\varepsilon}{12CN(2\beta)^{2k}(C+\frac{1}{2})\left(\frac{9N-5}{4}-k\right)} \qquad \text{if } \varepsilon \le (2\beta)^{2k}(C+\frac{1}{2}), \tag{3.66}$$

$$(1+6\tau)^{3N-k}\tau < \frac{1}{12CN\left(\frac{9N-5}{4}-k\right)} \qquad \text{if } \varepsilon > (2\beta)^{2k}(C+\frac{1}{2}), \tag{3.67}$$

$$\mu_{1:N}(\tau) < \frac{1}{2}\left(\frac{4C^2N+\varepsilon}{C^2N}\right)^{\frac{1}{2}} - 1 \tag{3.68}$$

w.r.t. $\mu_{1:N}(\tau)$ from the proof of Theorem 3.13. Conditions (3.66) and (3.67) are sufficient for $\delta^*(\tau) < \delta/2$ which is an immediate implication of combining (3.50) and (3.64). From (3.68) follows $g(\tau) \le \varepsilon/4$ since: for any admissible discrete interval I, we have

$$\begin{aligned}\left|\|\widetilde{Q^T f_I}\|^2 - \|Q^T f_I\|^2\right| &\le \left(\|Q^T f_I\| + \|\widetilde{Q^T f_I}\|\right)\left|\|Q^T f_I\| - \|\widetilde{Q^T f_I}\|\right| \\ &\le \left(C + \|\widetilde{Q^T f_I}\|\right)\|Q^T f_I - \widetilde{Q^T f_I}\|\end{aligned}$$

and

$$\|\widetilde{Q^T f_I}\| \le \|Q^T f_I\| + \|Q^T f_I - \widetilde{Q^T f_I}\| \le C + \|Q^T f_I - \widetilde{Q^T f_I}\| \le C + \mu_{1:N}C.$$

Combining both yields

$$\begin{aligned}g(\tau) = \left|\tilde{\mathcal{G}}_{\mathcal{I}}(f) - \mathcal{G}_{\mathcal{I}}(f)\right| &\le \sum_{I\in\mathcal{I}}\left|\tilde{\mathcal{E}}^I - \mathcal{E}^I\right| \\ &\le \sum_{I\in\mathcal{I}}\left|\|\widetilde{Q^T f_I}\|^2 - \|Q^T f_I\|^2\right| \le NC^2\mu_{1:N}(\tau)\left(2+\mu_{1:N}(\tau)\right).\end{aligned}$$

An easy computation shows that (3.68) is equivalent to requiring the latter to be smaller than $\varepsilon/4$.

Proof of Proposition 3.15. By (3.50), the solution u_f of ($\mathcal{P}_{k,\beta,\gamma}$) is unique for data f. We denote the equivalence class of partitions corresponding to this optimal solution u_f by its representer $\mathcal{I}'$. As a first step, we show that the solution $\tilde{u}_f$ computed by the proposed algorithm for data f has partition $\mathcal{I}'$ as well. To that end, we first notice that by Corollary 3.14, there is $\tilde{f}_{\mathcal{I}}$ with $\|\tilde{f}_{\mathcal{I}} - f\| < \delta_{\mathcal{I}}(\tau)$ such that $\tilde{S}_{\mathcal{I}}(f) = S_{\mathcal{I}}(\tilde{f}_{\mathcal{I}})$, for any partition $\mathcal{I}$ which is considered by the algorithm. In particular, using the notation of Theorem 3.13,

$$\|\tilde{S}_I(f_I) - S_I(f_I)\| = \|S_I(\tilde{f}_I) - S_I(f_I)\| \le \|S_I\|\,\|\tilde{f}_I - f_I\| \le \|\tilde{f}_I - f_I\| < \delta_I. \tag{3.69}$$

For the second before last inequality, we used that $\|S_{\mathcal{I}}\| \leq 1$ which we have shown in the proof of Lemma 3.12. As a consequence, by summing over all intervals of $\mathcal{I}$ of length at least $k+1$, we obtain from (3.69) that

$$\|\tilde{S}_{\mathcal{I}}(f) - S_{\mathcal{I}}(f)\| = \|S_{\mathcal{I}}(\tilde{f}_{\mathcal{I}}) - S_{\mathcal{I}}(f)\| \leq \|\tilde{f}_{\mathcal{I}} - f\| < \delta_{\mathcal{I}}(\tau) \tag{3.70}$$

for any partition $\mathcal{I}$ which is considered by the algorithm. Then, using the notation $\tilde{\mathcal{G}}_{\mathcal{I}}$ for the algorithmic variant of $\mathcal{G}_{\mathcal{I}}$, we have (with the notation as in Lemma 3.12) that

$$\begin{aligned}
\tilde{\mathcal{G}}_{\mathcal{I}'}(\tilde{f}_{\mathcal{I}'}) &\leq \mathcal{G}_{\mathcal{I}'}(\tilde{f}_{\mathcal{I}'}) + |\tilde{\mathcal{G}}_{\mathcal{I}'}(\tilde{f}_{\mathcal{I}'}) - \mathcal{G}_{\mathcal{I}'}(\tilde{f}_{\mathcal{I}'})| \\
&\leq \mathcal{G}_{\mathcal{I}'}(\tilde{f}_{\mathcal{I}'}) + g(\tau) \\
&< \mathcal{G}_{\mathcal{I}}(f) - \varepsilon + \delta_{\mathcal{I}'}(\tau)^2 \ \|A_{\mathcal{I}'}\| + 2\delta_{\mathcal{I}'}(\tau) \ \|A_{\mathcal{I}'}\| \ \|f\| + g(\tau) \\
&\leq \mathcal{G}_{\mathcal{I}}(\tilde{f}_{\mathcal{I}}) - \varepsilon + 2\delta^*(\tau) \ \max_{\mathcal{I}} \|A_{\mathcal{I}}\| \ \ (2\|f\| + \delta^*(\tau)) + g(\tau) \\
&\leq \mathcal{G}_{\mathcal{I}}(\tilde{f}_{\mathcal{I}}) - \varepsilon + \varepsilon/2 + g(\tau) \\
&\leq \tilde{\mathcal{G}}_{\mathcal{I}}(\tilde{f}_{\mathcal{I}}) - \varepsilon + \varepsilon/2 + 2g(\tau) \leq \tilde{\mathcal{G}}_{\mathcal{I}}(\tilde{f}_{\mathcal{I}}).
\end{aligned} \tag{3.71}$$

Here, the third inequality is the central estimate which is obtained in analogy to the first part of the computation in (3.55) by replacing the role of the vector g there (not to be confused with $g(\tau)$) by that of the perturbation $\tilde{f}'_{\mathcal{I}}$ of f here. The fourth inequality is obtained in analogy to the second part of the computation in (3.55) with the role of the vector g there replaced by the perturbation $\tilde{f}_{\mathcal{I}}$ of f. The second before last and the last inequality follow from our assumptions made on τ. Together, (3.71) tells us that the solution $\tilde{u}_f$ computed by the proposed algorithm has partition $\mathcal{I}'$ and

$$\tilde{u}_f = \tilde{S}_{\mathcal{I}'}(f). \tag{3.72}$$

Using again Corollary 3.14, we obtain

$$\tilde{S}_{\mathcal{I}'}(f) = S_{\mathcal{I}'}(\tilde{f}) \quad \text{ for } \quad \|\tilde{f} - f\| < \delta_{\mathcal{I}'}(\tau) \leq \delta^*(\tau), \tag{3.73}$$

with the perturbation $\tilde{f}$ of f. We have that $\|\tilde{f} - f\| < \delta^*(\tau) < \delta$ with δ defined by (3.51). Therefore, we may now employ Lemma 3.12 to conclude that the solution of the higher order Potts and Mumford-Shah problem $(\mathcal{P}_{k,\beta,\gamma})$ denoted by $u_{\tilde{f}}$ agrees with the optimal solution for the partition $\mathcal{I}'$ which we have denoted by $u_{\tilde{f},\mathcal{I}'} = S_{\mathcal{I}}(\tilde{f})$, i.e.,

$$u_{\tilde{f}} = u_{\tilde{f},\mathcal{I}'} = S_{\mathcal{I}'}(\tilde{f}). \tag{3.74}$$

Combined with (3.72) and (3.73), this shows (3.65) which completes the proof. □

Lemma 3.17. *We consider a nonzero quadratic form H in a ball of radius C in $\mathbb{R}^N$. Then, the Lebesgue measure λ of the set $H_{\varepsilon,c} = \{x : \|x\| \leq C, c - \varepsilon < H(x) < c + \varepsilon\}$ satisfies*

$$\lambda(H_{\varepsilon,c}) \leq 2\sqrt{\frac{\varepsilon}{\|A\|}}\, C^{N-1}, \tag{3.75}$$

where $\|A\|$ denotes the spectral norm of the representing matrix A of H.

Proof. Without loss of generality, we may use an orthogonal transformation of the coordinate system to represent H by $H(x) = \sum_i \alpha_i x_i^2$ with the eigenvalues α_i of the corresponding representing matrix of the quadratic form. We sort the α_i by modulus, i.e., $|\alpha_1| \geq |\alpha_2| \geq \ldots$. With respect to this coordinate system, we consider the C-ball with respect to the maximum norm $D = \{x : \|x\|_\infty \leq C\}$. We distinguish the eigenvalue α_1 of highest modulus which agrees with the norm $\|A\|$ of the representing matrix A of H. We estimate the Lebesgue measure of $\{x : c-\varepsilon < H(x) < c+\varepsilon\}$ on the larger set D which provides an upper bound for that of $H_{\varepsilon,c}$. To this end, we notice that, for fixed $x_2, \ldots, x_N$, we may estimate the univariate Lebesgue measure λ^1 of the section $\{x_1 : x_1 \leq C, c-\varepsilon < H(x_1, x_2, \ldots, x_N) < c+\varepsilon\}$ by

$$\lambda^1\left(c + \sum_{i=2}^{N} \frac{\alpha_i}{|\alpha_1|} x_i^2 - \varepsilon < \operatorname{sign}(\alpha_1)\, x_1^2 < c + \sum_{i=2}^{N} \frac{\alpha_i}{|\alpha_1|} x_i^2 + \varepsilon\right) \leq 2\sqrt{\frac{\varepsilon}{|\alpha_1|}}. \tag{3.76}$$

(Notice that if $\alpha_1 = 0$ the quadratic form would be zero.) Hence, on D, the Lebesgue measure of $\{x : c-\varepsilon < H(x) < c+\varepsilon\}$ is bounded by $2\sqrt{\frac{\varepsilon}{|\alpha_1|}}\, C^{N-1}$ which implies the assertion of the lemma. □

Theorem 3.18. *Let $\varepsilon > 0$ be given and assume that the precision τ fulfills the assumptions of Proposition 3.15. We consider the set of bounded data $\{f : \|f\| \leq C\}$ in $\mathbb{R}^N$ for some $C > 0$. Then, up to a set of Lebesgue measure $2\binom{\sigma_{N,k}}{2}\sqrt{\frac{\varepsilon}{\sup_{\mathcal{I}}\|A_{\mathcal{I}}\|}}\, C^{N-1}$, $A_{\mathcal{I}}$ given by (3.53), $\sigma_{N,k}$ the number of different means to choose intervals of length at least $k+1$ from a (discrete) set of length N, the proposed algorithm for computing a minimizer of the higher order Potts and Mumford-Shah problem $(\mathcal{P}_{k,\beta,\gamma})$ is backward stable in the sense that*

$$\tilde{u}_f = u_{\tilde{f}} \quad \textit{where} \quad \|\tilde{f} - f\| < \delta^*(\tau), \tag{3.77}$$

where $\delta^*(\tau)$ is given by (3.64). Here, $\tilde{u}_f$ is the result produced by the proposed algorithm for data f and $u_{\tilde{f}}$ is the (unique) solution of the higher order Potts and Mumford-Shah problem $(\mathcal{P}_{k,\beta,\gamma})$ for perturbed data $\tilde{f}$.

Proof. We proceed similarly to the proof of Theorem 3.8 to show that the set of those data which do not fulfill (3.50) have Lebesgue measure smaller or equal to $2\binom{\sigma_{N,k}}{2}\sqrt{\frac{\varepsilon}{\sup_{\mathcal{I}}\|A_{\mathcal{I}}\|}}\, C^{N-1}$. We choose two different partitions $\mathcal{I}, \mathcal{I}'$ with $[\mathcal{I}] \neq [\mathcal{I}']$, i.e., their equivalence classes do not agree, and consider the corresponding quadratic forms $\mathcal{G}_{\mathcal{I}}, \mathcal{G}_{\mathcal{I}'}$. Their difference $\mathcal{G}_{\mathcal{I}} - \mathcal{G}_{\mathcal{I}'}$ is again a quadratic form (plus a constant). By Lemma 3.17, the set where $\mathcal{G}_{\mathcal{I}}$ and $\mathcal{G}_{\mathcal{I}'}$ are closer than ε has Lebesgue measure $2\sqrt{\frac{\varepsilon}{|\alpha_1|}}\, C^{N-1}$. Iterating this for all $\binom{\sigma_{N,k}}{2}$ different bilinear forms $\mathcal{G}_{\mathcal{I}} - \mathcal{G}_{\mathcal{I}'}$ shows that the Lebesgue measure of those data for which (3.50) is not fulfilled can be estimated from above by the quantity written in the formulation of the theorem. We may now apply Proposition 3.15 to the complementary set to conclude the assertion of the theorem. □

3.4. Numerical study

In this section, we study the reconstruction quality of higher order Mumford-Shah and Potts models as well as the runtimes of the proposed solvers in numerical experiments. We implemented the proposed algorithm in the C++ programming language. The algorithm was then called from Matlab by using mex-files. (The pseudocode was given in Algorithm 3.1.) We performed all experiments on a standard desktop computer with 3.1 GHz Intel Core i5-2400 processor and 8 GB RAM.

3.4.1. Reconstruction results

We start out by studying the reconstruction quality of higher Mumford-Shah and Potts models. To this end, we consider common test signals with discontinuities; see [66, 144].

Experimental setup. The signals were corrupted by additive white Gaussian noise with variance σ^2. We denote by η the noise level which is given by $\eta = \sigma N / \|g\|_1$ in terms of the clean signal g. In order to compare the models in a meaningful way, we use optimal complexity parameters γ and smoothing parameters β for each model. In particular, we determined the optimal γ and β such that the result u^* provides the smallest relative reconstruction error

$$\varepsilon_{\mathrm{rel}} = \|u^* - g\|_2 / \|g\|_2.$$

In practice, this is accomplished by performing a full grid search over $\gamma \in (0, 1]$ with step size 0.001, and over $\beta \in (0, 25]$ with step size 0.025. Additionally, we check for the higher Potts model $\beta = \infty$. Besides the reconstruction error, we also assess the quality of the computed partition $\mathcal{I}^*$. We quantify it by the Rand index [167] which we briefly explain in the following. The Rand index $\mathrm{R_{ind}}$ quantifies the closeness of two partitions $\mathcal{I}, \mathcal{I}'$ by means of

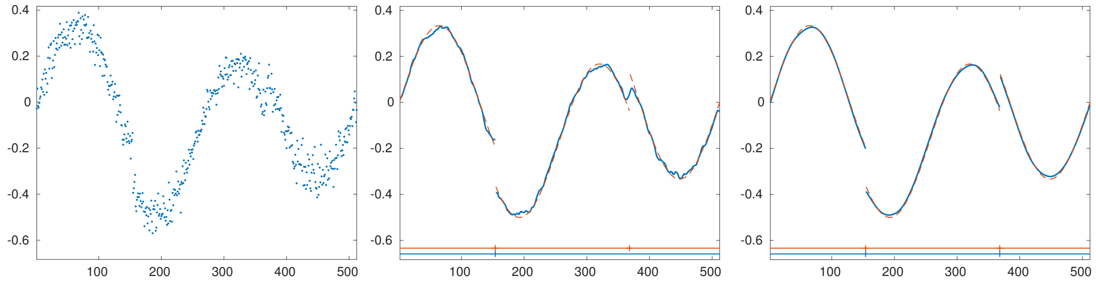
$$\mathrm{R_{ind}}(\mathcal{I}, \mathcal{I}') = \binom{N}{2} \sum_{\{i,j :\, 1 \leq i < j \leq N\}} t_{ij}.$$

Here, t_{ij} is equal to one if there are intervals/segments $I \in \mathcal{I}$ and $I' \in \mathcal{I}'$ such that the indexes i and j are in both I and I', or if i is in both I and I' while j is in neither I and I'. Otherwise, $t_{ij} = 0$. (Recall that N denotes the length of the signal.) The value of $\mathrm{R_{ind}}$ lies in $[0, 1]$ and higher values indicate a better match. In particular, $\mathrm{R_{ind}} = 1$ means that the two segmentations agree. We use the Rand index of the computed segmentation and the ground truth segmentation to measure the quality of the computed segmentation.[1] We recall that larger values of β give a stronger smoothing and larger values of γ give less segments.

Heavy sine signal. We first consider a sinusoidal signal with two steps and noise level 0.2. The reconstruction results are shown in Figure 3.5. For the first order model, the optimal smoothing parameter β is relatively small to avoid the gradient limit effect. As a consequence, the reconstructed signal remains affected by the noise and further only one of the two discontinuities is detected. The higher order models provide an improved smoothing and produce better results w.r.t. the relative reconstruction error and the segmentation quality. It is worth mentioning that the reconstruction quality starts decaying from order $k = 6$ on which can be attributed to overfitting.

---

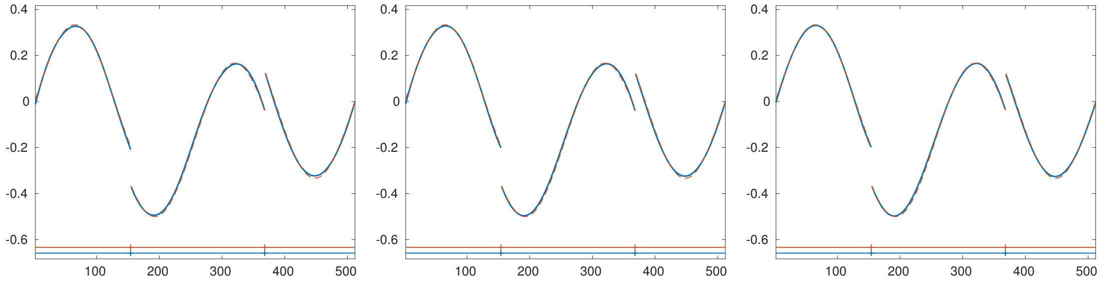
[1] The numerical evaluation of the Rand index was done using the implementation of K. Wang and D. Corney available at the Matlab File Exchange.



(a) Noisy data

(b) $(\mathcal{P}_{1,\beta,\gamma})$, $\beta = 4.050$, $\gamma = 0.109$
$\varepsilon_{\text{rel}} = 0.053$, $R_{\text{ind}} = 0.76$

(c) $(\mathcal{P}_{2,\beta,\gamma})$, $\beta = 9.500$, $\gamma = 0.067$
$\varepsilon_{\text{rel}} = 0.029$, $R_{\text{ind}} = 1.00$



(d) $(\mathcal{P}_{3,\beta,\gamma})$, $\beta = 12.850$, $\gamma = 0.102$
$\varepsilon_{\text{rel}} = 0.027$, $R_{\text{ind}} = 1.00$

(e) $(\mathcal{P}_{4,\beta,\gamma})$, $\beta = 15.300$, $\gamma = 0.115$
$\varepsilon_{\text{rel}} = 0.026$, $R_{\text{ind}} = 1.00$

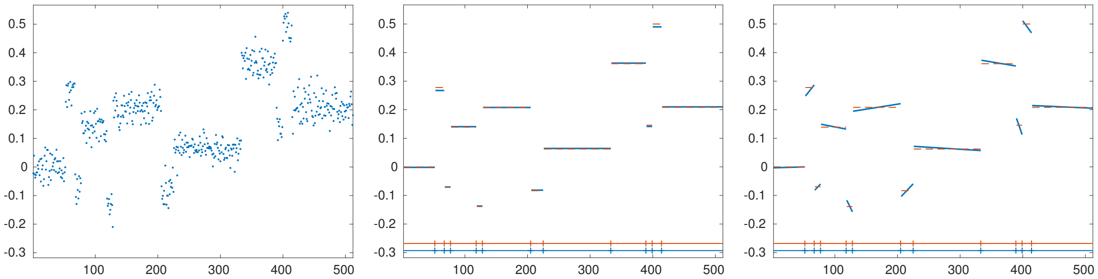
(f) $(\mathcal{P}_{5,\beta,\gamma})$, $\beta = 18.850$, $\gamma = 0.13$
$\varepsilon_{\text{rel}} = 0.024$, $R_{\text{ind}} = 1.00$

Figure 3.5.: Reconstructions of the “Heavy Sine”-signal from noisy data. (a) Data corrupted by Gaussian noise of level 0.2. (b–f) Reconstructions for higher order Mumford-Shah and Potts models.

Blocks signal. The next signal is piecewise constant and the noise level is 0.2. In Figure 3.6, we show the reconstruction results. Since the signal is constant within its segments, it is intuitive that the optimal smoothing parameters are large and mostly $\beta = \infty$, i.e., the k-th order Potts model, is preferable. The best result is produced by the first order Potts model whose piecewise constancy assumption matches the signal perfectly. Nevertheless, using higher order models yields very good segmentation results as well (up to order $k = 5$). The reconstructions are satisfying up to order $k = 3$.

Piecewise smooth reconstruction for different noise levels. In Figure 3.7, we show the reconstruction results of a piecewise smooth signal for different noise levels. The results of the first order model remain relatively noisy on the segments. Again, this is due to relatively small choices of the smoothing parameter to prevent the gradient limit effect. The second order model improves the reconstructions significantly. However, it also tends to produce spurious segments at the parts of high curvature. Using higher orders $k \geq 3$ leads to improved smoothing and better segmentations.

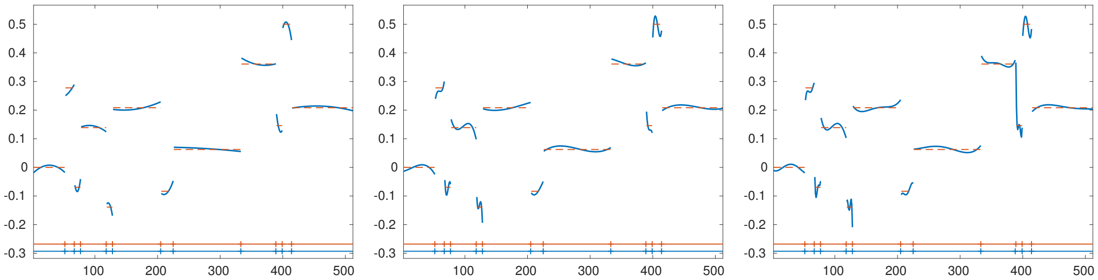
Real data. We consider a real data times series. In particular, we consider the time-averaged (hourly) wind speeds at the summit of the highest German mountain Zugspitze collected from



(a) Noisy data

(b) $(\mathcal{P}_{1,\beta,\gamma})$, $\beta = \infty$, $\gamma = 0.343$
$\varepsilon_{\mathrm{rel}} = 0.015$, $R_{\mathrm{ind}} = 1.00$

(c) $(\mathcal{P}_{2,\beta,\gamma})$, $\beta = \infty$, $\gamma = 0.205$
$\varepsilon_{\mathrm{rel}} = 0.036$, $R_{\mathrm{ind}} = 1.00$



(d) $(\mathcal{P}_{3,\beta,\gamma})$, $\beta = \infty$, $\gamma = 0.078$
$\varepsilon_{\mathrm{rel}} = 0.044$, $R_{\mathrm{ind}} = 1.00$

(e) $(\mathcal{P}_{4,\beta,\gamma})$, $\beta = \infty$, $\gamma = 0.044$
$\varepsilon_{\mathrm{rel}} = 0.054$, $R_{\mathrm{ind}} = 1.00$

(f) $(\mathcal{P}_{5,\beta,\gamma})$, $\beta = 24.300$, $\gamma = 0.045$
$\varepsilon_{\mathrm{rel}} = 0.058$, $R_{\mathrm{ind}} = 0.99$

Figure 3.6.: Reconstructions of the “Blocks”-signal from noisy data. (a) Data corrupted by Gaussian noise of level 0.2. (b–f) Reconstructions using higher order Mumford-Shah and Potts models.

November to December 2016.[2] We observe that breakpoints are introduced for strong changes of the wind speed Further, we can associate some breakpoints with a meaning: the break at 492 and the two breaks near 1154 and 1182 correspond to the days of strongest squalls in November and December 2016, respectively.[3]

3.4.2. Computation time

We study the computation time of the proposed algorithm depending on the signal length N. In order to measure the effectiveness of the pruning strategies explained at the end of Section 3.3.1, we further investigate the number of executed error updates $\mathcal{E}^{l:r} \to \mathcal{E}^{l:r+1}$ in terms of (3.42). Recall that the computation times strongly depend on the number of executed error updates.

Signal generation. We generate two types of synthetic signals: signals whose number of discontinuities increases with their length N and signals with constant number of discontinu-

---

[2] The data were collected by German climate data center and are available via ftp at ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate/hourly/wind/historical/ (station id: 02115).

[3] *Monatsrückblick der Wetterwarte Garmisch-Partenkirchen/Zugspitze* at http://www.schneefernerhaus.de/fileadmin/web_data/bilder/pdf/MontasrueckblickeZG/MORZG1116.pdf and at http://www.schneefernerhaus.de/fileadmin/web_data/bilder/pdf/MontasrueckblickeZG/MORZG1216.pdf

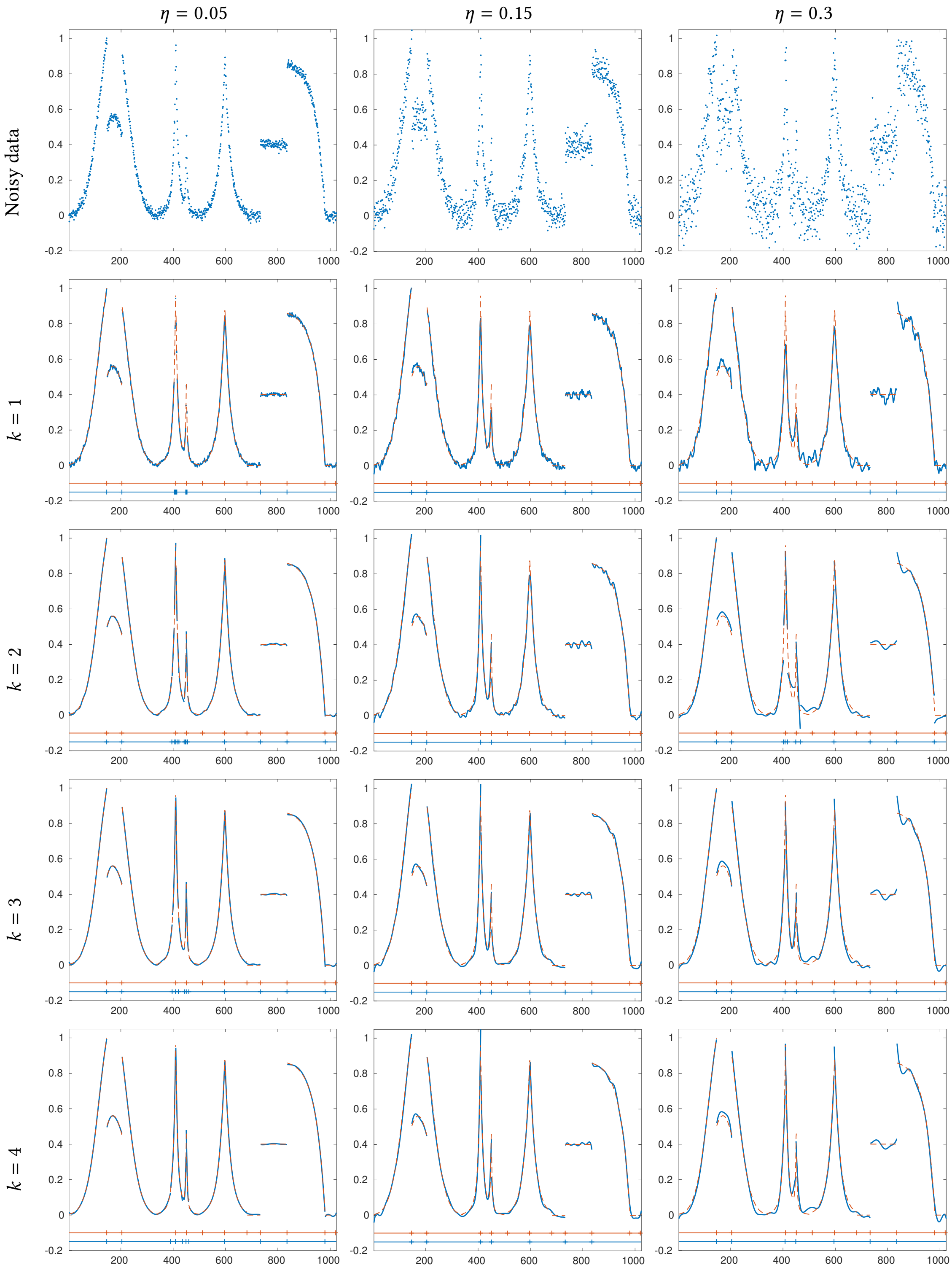


Figure 3.7.: Smoothing of a piecewise defined signal with increasing noise level η *(top row).* We observe that the segmentation quality improves and that the noise is smoothed out better on the segments when using higher order Mumford-Shah models.

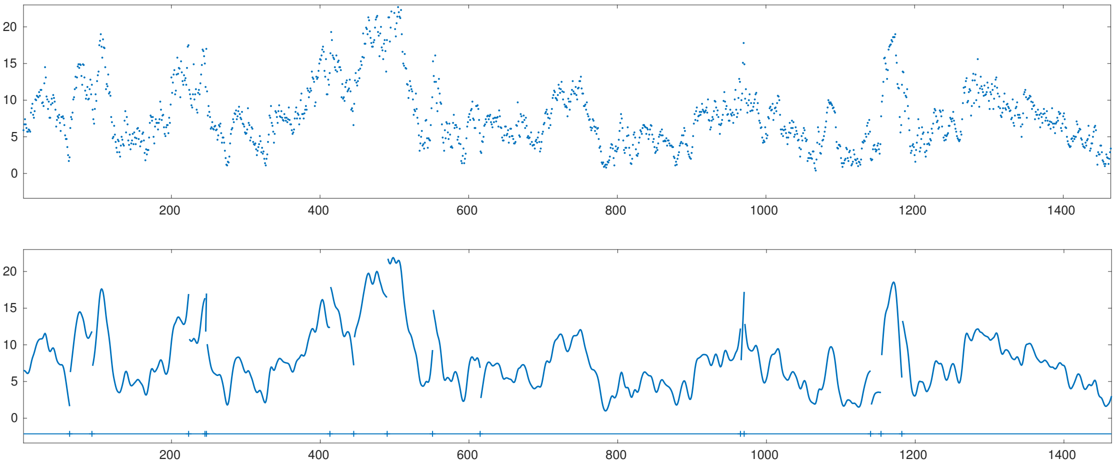


Figure 3.8.: *Top:* hourly averaged windspeeds [m/s] at the summit of the Zugspitze from November to December 2016. *Bottom:* result of the higher order Mumford-Shah model ($\mathcal{P}_{2;2;15}$).

ities for all lengths N. Concerning the first type of signals, we fix the probability of a jump discontinuity after each data point to $p = 0.01$. Thus, the individual segment length follows a geometric distribution with parameter p and the expected length is $1/p = 100$. Further, the expected number of segments grows linearly with respect to N. On a segment I, the signal g is a polynomial of degree $k-1$ and its coefficients are generated by the random variables $\frac{1}{(j+1)^2}X_j$, $j = 1, \ldots, k$, where X_j are i.i.d. uniformly distributed on $[-1, 1]$. The domain of g_I is $[0, hp]$ for the length h of I sampled with step size p. For model order k, the polynomial degree is set to $k - 1$. The second type of signals is obtained by (equidistant) sampling of the continuous counterpart of the signal shown in Figure 3.7. In all cases, the signals are corrupted by additive Gaussian noise with noise level $\eta = 0.1$. For every considered signal length, we computed 1000 instances. We report the mean computation time and the mean number of performed error updates, respectively.

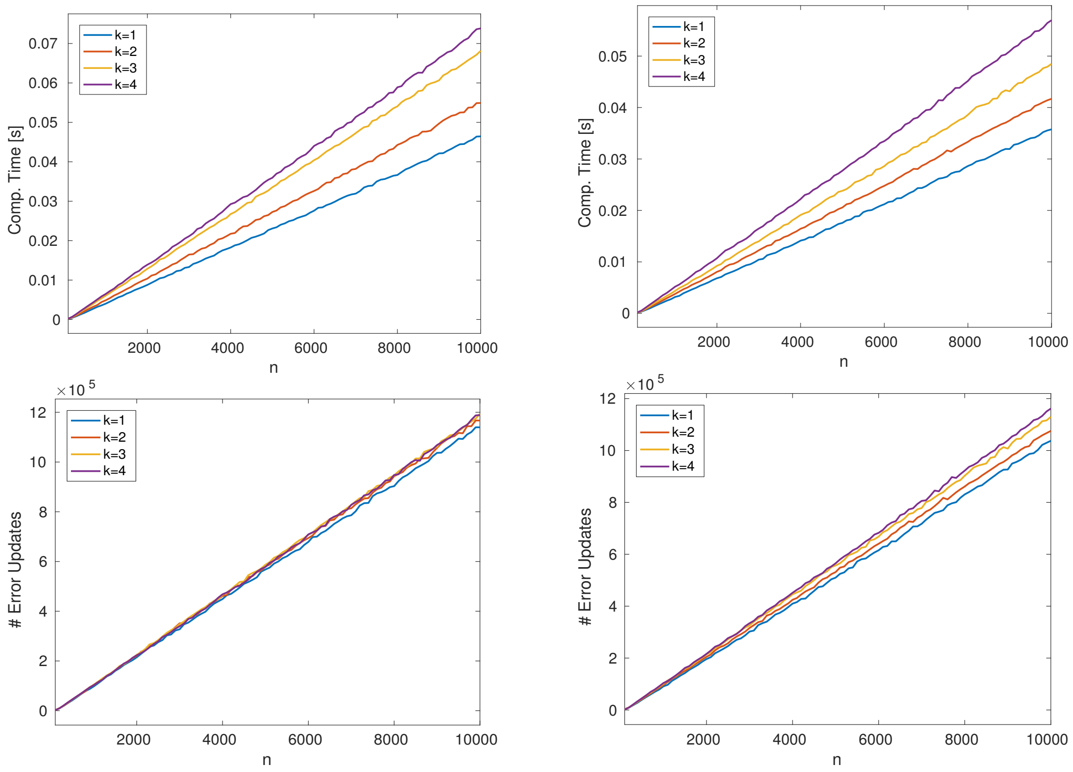
Results. In Figure 3.9, we report the results for the first type of signals. We make the important observation that the runtimes and number of error updates grow linearly in the signal length. Hence, the proposed algorithm does not exhibit the worst case complexity which is quadratic. The pruning strategies are therefore highly effective. The results for the second type of signals are shown in Table 3.1. Here –in contrast to the first type of signals– the computation time is approximately quadratic in the number of elements. This means that the proposed algorithm attains its worst case complexity. The results indicate that the proposed algorithm is particularly efficient for an increasing number of discontinuities.

3.5. Summary of the chapter

In this chapter, we studied higher order Mumford-Shah and Potts models for univariate data. In contrast to the classical first order models, which penalize deviations from constant sig-

Time [s]	k =1	2	3	4
n =1000	0.0040	0.0049	0.0061	0.0065
4000	0.0183	0.0217	0.0268	0.0293
7000	0.0319	0.0382	0.0473	0.0513
10000	0.0464	0.0549	0.0682	0.0738

Time [s]	k =1	2	3	4
n =1000	0.0032	0.0038	0.0044	0.0052
4000	0.0141	0.0164	0.0191	0.0222
7000	0.0247	0.0291	0.0333	0.0395
10000	0.0358	0.0417	0.0485	0.0570



(a) Higher order Mumford-Shah model.

(b) Higher order Potts model.

Figure 3.9.: Computational costs of Algorithm 3.1 for randomly generated piecewise polynomial signals corrupted by Gaussian noise of level $\eta = 0.1$. *Top:* tables with computation times for selected lengths N. *Center:* plots of computation times for $N = 100, 200, \ldots, 10000$. *Bottom:* plots of the number of performed error updates for $N = 100, 200, \ldots, 10000$. The runtime only grows linearly in N, which is much more favorable than the worst quadratic growth. This shows that the pruning strategies show their full effectiveness for this type of signals, which is reflected by the linear growth of the number of performed error updates.

nals, higher order Mumford-Shah models penalize deviations from polynomial trends of oder $k-1$. This improves the estimation of data with piecewise linear or polynomial trends. In particular, higher order models provide improved smoothing as well as improved segmentations compared to the first order models. We showed that the underlying minimization problems have unique minimizers for almost all input signals. We derived a fast solver for higher order Mumford-Shah and Potts models and obtained stability results for it. The solver uses a dynamic

	N								
k	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}
1	0.0025	0.0091	0.0363	0.1132	0.4230	1.6557	6.5899	28.8285	123.2784
2	0.0025	0.0064	0.0210	0.0714	0.4875	1.9040	7.5987	33.0966	139.5842
3	0.0030	0.0075	0.0196	0.0873	0.3435	2.3475	9.4270	40.4083	166.7910
4	0.0032	0.0078	0.0569	0.0935	0.3685	1.6173	10.1401	44.7423	179.0001

(a) Runtime [s] for (higher order) Mumford-Shah solver

	N								
k	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}
1	0.0010	0.0030	0.0055	0.0147	0.0502	0.1627	0.5713	2.0363	7.1102
2	0.0016	0.0037	0.0090	0.0292	0.1036	0.3693	1.3562	4.7746	16.8562
3	0.0019	0.0048	0.0116	0.0423	0.1613	0.6022	2.2298	8.2050	30.1515
4	0.0024	0.0057	0.0172	0.0517	0.1977	0.7536	2.8576	10.8020	41.7345

(b) Runtime [s] for (higher order) Potts solver

Table 3.1.: Mean computation times of Algorithm 3.1 (in seconds) of the signal from Figure 3.7 sampled on N points and corrupted by Gaussian noise with noise level 0.1. The runtime grows approximately quadratically in N which is the worst case complexity. The relevant difference to the experiment in Figure 3.9 is that the number of discontinuities does not grow with the signal length N. Yet, the solver processes signals of size 2^{16} in less than one minute.

programming scheme and the involved computation of approximation errors is based on an update strategy with Givens rotations. The numerical experiments confirmed the stability of the proposed method and its robustness to noise. We showed that the worst case complexity of the proposed algorithm is quadratic in the length of the signal for all orders $k \geq 1$. In our experiments, we illustrated the advantages of higher order models for signals with polynomial trends and we further observed that the practical runtime grows only linear for signals with linearly increasing number of discontinuities. Our C++ implementation processes even long signals in reasonable time. In particular, signals, which have length $10,000$, are processed in less than one second. Thus, higher order Mumford-Shah and Potts models can serve as efficient smoothers for signals with discontinuities.

4. Surrogate Functional Approaches to the Inverse Potts Model

In this chapter, we consider the inverse Potts model for images which are indirectly given in terms of measured data. This chapter is based on the publication [118].

Organization of the chapter. First, in Section 4.1, we give an overview and discuss related work. Next, we derive the proposed algorithmic schemes in Section 4.2. In Section 4.3, we provide a convergence analysis for the proposed schemes. In Section 4.4, we apply the proposed algorithms to concrete reconstruction and partitioning problems, respectively. In Section 4.5, we give a summary of the chapter.

4.1. Overview and related work

In many applications, images are not given directly but indirectly in terms of measured data. Such applications are for example computed tomography (CT), magnetic resonance imaging (MRI) or microscopy. Typically, the measured data are noisy and often incomplete so that the reconstruction of the unknown image becomes difficult. In particular, the reconstruction process is ill-posed. Therefore, it is necessary to apply prior knowledge/assumptions on the image to be reconstructed, i.e., the reconstruction process needs to be regularized.

To this end, we apply the *inverse Potts model* which regularizes the reconstruction process by assuming piecewise constancy and thereby partitions the unknown image directly from its data. We point out that executing the reconstruction and partitioning steps jointly typically leads to better results than performing the two steps successively [124, 165, 166, 183]. Further, the usage of the Potts model in connection with inverse problems is theoretically motivated by the considerations in [166]. The authors showed (after restricting the solution space) under relatively mild assumptions that the method provides a regularizer in the sense of inverse problems. In this context, we note that Mumford-Shah and Potts approaches also regularize the boundaries of the underlying signal [108].

We start out by formulating the inverse Potts model, i.e., the counterpart of the classical Potts model (1.2) for image data given in terms of an imaging operator. The Potts model for indirect data f is given by

$$\begin{aligned} &\operatorname*{argmin}_{u,\mathcal{P}} \|Au - f\|_2^2 + \sum_{P \in \mathcal{P}} \frac{\gamma}{2} \operatorname{length}(\partial P) \\ &\text{subject to } u|_P \text{ is constant for all } P \in \mathcal{P}. \end{aligned} \tag{4.1}$$

The linear operator A –the forward operator– models the measurement or imaging process. Examples for A are point spread functions in microscopy, the Radon transform in CT or sampled Fourier transforms, which appear in MRI. (Recall that (1.2) in Chapter 1 corresponds to $A = \mathrm{id}$.) Accordingly, the available data f is an element of the data space, e.g., a blurred image, a sinogram in CT or k-space data in MRI. As for $A = \mathrm{id}$ in (1.2), the minimization in (4.1) takes place w.r.t. the partitions $\mathcal{P}$ of the function domain $\Omega \subset \mathbb{R}^2$ into disjoint connected open sets P and the functions $u : \Omega \to \mathbb{R}$ which are constant on each segment $P \in \mathcal{P}$. In order to avoid oversegmentation, the total length of the segment boundaries is penalized. More precisely, $\mathrm{length}(\partial P) = \mathcal{H}^1(\partial P)$ denotes the one-dimensional Hausdorff measure of the segment boundary ∂P (see Definition A.1). To fix ideas, if the boundary of P corresponds to a C^1-curve, $\mathrm{length}(\partial P)$ equals its arc length. The second term measures the fidelity of u to the data f in terms of the squared ℓ_2-norm. The parameter $\gamma > 0$ balances the data fidelity and the length term.

The formulation of the Potts model with partitions (4.1) is instructive, but can become cumbersome in practice. An equivalent formulation in terms of the image u only is given by

$$\operatorname*{argmin}_u \|Au - f\|_2^2 + \gamma \|\nabla u\|_0, \tag{4.2}$$

where the gradient ∇u is understood in the distributional sense (recall (1.2) and Definition A.2) and the ℓ_0-term in (4.2) is given by

$$\|\nabla u\|_0 = \mathcal{H}^1\Big(\{x \in \Omega : \nabla u(x) \neq 0\}\Big) \tag{4.3}$$

which is finite if and only if u is piecewise constant. Thus, feasible solutions of (4.2) are piecewise constant as in (4.1). In view of this, the set of points where $\nabla u(x) \neq 0$ is called the *jump set* of the piecewise constant function u. Thus, we call $\|\nabla u\|_0$ the *jump term* or *jump penalty*. The jump set of u corresponds exactly to the boundaries of the segments $P \in \mathcal{P}$ in (4.1). Note that we dropped the factor $\frac{1}{2}$ in (4.1) which compensated for the double counting of boundaries.

Summing up, the inverse Potts model (4.2) can be seen as a reconstruction approach if the (unknown) image under consideration is (approximately) piecewise constant and as an partitioning approach since the piecewise constant result induces directly a partitioning. Thus, the (inverse) Potts model (4.2) is an approach to *joint reconstruction and partitioning*.

Solving the Potts problem (4.2) is algorithmically challenging as the underlying minimization problem is NP-hard even for the simplest forward operator $A = \mathrm{id}$; see [27, 190]. (In contrast, for univariate signals it can be solved efficiently if $A = \mathrm{id}$ as seen in Chapter 3, but it also becomes NP-hard for general linear operators $A \neq \mathrm{id}$ [201].) In this chapter, we develop new algorithmic approaches to the Potts model (4.2) based on majorization-minimization with surrogate functionals, give a convergence analysis of the proposed algorithms and underpin our findings in numerical experiments.

Related Work. We start with the Potts problem for general operators A and discuss the special case $A = \mathrm{id}$ separately. In [9], Bar et al. use an approximation of Ambrosio-Tortorelli type. The authors of [122] employ a level-set based active contour method for deconvolution. Ramlau and Ring [165] consider joint reconstruction and segmentation of X-ray CT images for which they apply a level-set approach. Other applications are electron tomography [123] and SPECT [124]. An ADMM strategy was proposed in [182] for the univariate case and in [183] for the multivariate case.

We discuss work related to Mumford-Shah models for indirect data. Fornasier and Ward [75] derive a reformulation of the problems as pointwise penalized problems and propose iterative thresholding algorithms to approach them. They prove that the method converges to a local minimizer in the univariate case. This approach principally carries over to the Potts problem which results in an ℓ_0 sparsity problem as explained in [182, 201]. This unconstrained NP-hard optimization problem may be addressed by iterative hard thresholding algorithms for ℓ_0 penalizations, which were analyzed by Blumensath and Davies in [22, 23]. (Daubechies, Defrise, and De Mol [59] considered iterative soft thresholding for related ℓ_1 penalized problems.) Artina et al. [5] consider the multivariate Mumford-Shah model and use the pointwise penalization approach of [75] which results in a corresponding linearly constrained non-convex and non-smooth problem. The authors approach this problem by successively minimizing local quadratic and strictly convex perturbations (depending on the previous iterate) of a smoothed version of the objective by augmented Lagrangian iterations, which themselves can be accomplished by iterative thresholding via a Lipschitz continuous thresholding function. They show that the sequences produced by their algorithm yield accumulation points, which are constrained critical points of the smoothed problem. A similar approach for rewriting the Potts problem results in an ℓ_0 sparsity problem with additional equality constraints. Algorithmic approaches for such constrained ℓ_0 sparsity problems are the penalty decomposition methods of [142, 143, 217]. They are related to iterative thresholding in the sense that the inner loop of the employed two stage process usually is of the iterative hard thresholding type. In contrast to methods using hard thresholding, the approaches developed in this chapter do not have to deal with constraints and the full matrix A but with the regularizing term $\|\nabla u\|_0$, which is non-separable, so that we cannot use hard thresholding.

An often used method for restoring piecewise constant images is total variation minimization [169]. Here, one uses the convex term $\|\nabla u\|_1$ instead of the jump penalty $\|\nabla u\|_0$. Therefore, the corresponding minimization problem is convex and thus numerically tractable by convex optimization techniques [51, 57]. Towards genuinely piecewise constant solutions, Candès, Wakin, and Boyd [37] use iteratively reweighted total variation minimization. Results of compressed sensing type related to the Potts problem have been derived in [151, 152]: if certain conditions are fulfilled, the minimizers of the Potts functional and of total variation will agree. In the presence of noise –which is always the case in practical applications– the results of total variation minimization and Potts minimization can differ significantly. However, the minimizers of the Potts problem are frequently desired in practice. Algorithms based on convex relaxations of the Potts problem (4.2) have been often considered in recent years [162, 134, 7, 50, 33, 87, 185].

We next discuss approaches to the multivariate Potts problem for the situation $A = \mathrm{id}$, that is, image partitioning for which further approaches exist. Classical algorithms are based on simulated annealing [83] and on approximations by elliptic functionals [3]. Another class of approaches corresponds to the application of graph cuts. As a first step, the codomain of u has to be restricted to a relatively small number of values. While the problem remains NP-hard, it can then be approached by sequentially solving binary partitioning problems via minimal graph cut algorithms [27, 127, 26]. There is also an approach which limits the number of values of u to k values without fixing them a priori. For $k = 2$, Chan and Vese [53] used the active contours to minimize the corresponding binary Potts model. The partitions are represented by a level set function which evolves according to the Euler-Lagrange equations of the Potts model. In [52], the authors propose a strategy for the binary segmentation problem, which is globally convergent. Later in [191], the active contour method for the binary case was extended to k values (for $k > 2$ the problem remains NP-hard). Further works on active contours are [70, 219, 58]. For an overview on level set segmentation, we refer to [58]. In [98, 99, 100], Hirschmüller proposes a non-iterative strategy for the Potts problem based on cost aggregation. Compared to graph cuts it has lower computational cost at the expense of lower quality of the results. Since the number of potential values of u has to be restricted, these methods are typically used in connection with image segmentation. Other approaches are based on semi-global matching [100, 68], fused coordinate descent [56], region fusion [154], iterative thresholding type techniques [75], and the alternating direction method of multipliers [181]. To obtain faster computation times, parallelization has recently received a lot of attention; see [162, 185, 163, 180]. Nikolova et al. [156, 155] proposed methods for restoring piecewise constant images, in which they use non-convex regularizers and the problems are algorithmically approached using a graduated non-convexity approach. We note that the Potts problem (4.2) is not a member of the class of problems considered in [156, 155]. Xu et al. [208] proposed a piecewise constant model strongly reminiscent of the Potts model which they approach by a half-quadratic splitting and using a pointwise iterative thresholding type technique. This was later extended to blind image deconvolution [210].

4.2. Majorization-minimization algorithms for multivariate Potts problems

In this section, we derive the proposed algorithmic approaches to multivariate inverse Potts problems (4.2). In particular, we start with the discretization of the model (4.2), relax the discrete optimization problem and reformulate the relaxed problem in a convenient way. Based on these preparations we derive a majorization-minimization algorithm for the relaxed problem. Building on that, we derive an algorithm for the original problem. It turns out that the subproblems which arise in the algorithmic schemes do not decompose into pixelwise thresholding as for ℓ_0-sparsity problems. However, they can be solved exactly and efficiently by the dynamic programming approach of Chapter 3.

4.2.1. The discrete problem

We use a common discretization of the jump term in (4.2) in terms of finite differences [49, 183]. Concretely, the Potts problem for discrete data is given by

$$\min_u \left\{ P_\gamma(u) = \|Au - f\|_2^2 + \gamma \sum_{s=1}^{S} \omega_s \, \|\nabla_{a_s} u\|_0 \right\}. \tag{4.4}$$

Here, the integer vectors $a_s \in \mathbb{Z}^2$, $s = 1, \dots, S$ are the directions of the finite differences $\nabla_{a_s} u(x) = u_{(i,j)+a_s} - u_{i,j}$ which are weighted by $\omega_s > 0$. We denote by $\nabla_{a_s} u$ the vector which comprises these differences for all x. The symbol $\|\nabla_{a_s} u\|_0$ now denotes the number of nonzero entries of $\nabla_{a_s} u$. (Note that from now on we only consider the discrete setting, so we overloaded the symbol $\|\cdot\|_0$.) The set of directions $\mathcal{N} = \{a_s\}_{s=1}^S$ forms a neighborhood system. The simplest choice of the neighborhood system $\mathcal{N}$ corresponds to the unit vectors $a_1 = (0,1)$, $a_2 = (1,0)$ with the unit weights $\omega_1 = \omega_2 = 1$. However, this discretization converges to a (continuous domain) limit which penalizes the boundaries in terms of the ℓ_1-counterpart of the Hausdorff measure [48]. In practice, this manifests in undesired block artifacts (geometric staircasing). This can be improved by including the diagonal directions $a_3 = (1,1)$, $a_4 = (1,-1)$ to the directions a_1 and a_2 which overall yields an 8-neighborhood system. Another choice is the 16-neighborhood system which also incorporates the knight-moves $a_5 = (1,2)$, $a_6 = (2,1)$, $a_7 = (1,-2)$, $a_8 = (2,-1)$. (The additional directions correspond to the moves of the knights in chess.) In Figure 4.1, we give an illustration of these neighborhood systems.

We need to determine the weights ω_s of the 8-neighborhood and 16-neighborhood system, respectively. The weights we use here are chosen so that the discrete length of an isolated jump set/edge along each of the directions a_s equals its length in the continuous setting [183]. For the 8-neighborhood system this is accomplished by the weights $\omega_s = \sqrt{2} - 1$ for the coordinate part $s = 1, 2$ and $\omega_s = 1 - \frac{\sqrt{2}}{2}$ for the diagonal part $s = 3, 4$. When incorporating the knight-move directions, the weights are chosen as $\omega_s = \sqrt{5} - 2$ for the coordinate part $s = 1, 2$, $\omega_s = \sqrt{5} - \frac{3}{2}\sqrt{2}$ for diagonal part $s = 3, 4$, and $\omega_s = \frac{1}{2}(1 + \sqrt{2} - \sqrt{5})$ for diagonal part $s = 5, \dots, 8$. Concerning the weight design for general neighborhood systems, we refer to [49, 183]. For the 8-neighborhood system the weights obtained in [49] and [183] coincide. For the 16-neighborhood system the weights above correspond to the weights derived by the scheme in [183]. In this chapter, we focus on the 8-neighborhood discretization.

We record that the discrete problem has a minimizer.

Theorem 4.1. *The discrete multivariate Potts problem* (4.4) *has a minimizer.*

The proof follows the lines of the proof of [101, Theorem 2.1] which proved the analogous statement for the inverse piecewise smooth Mumford-Shah problem.

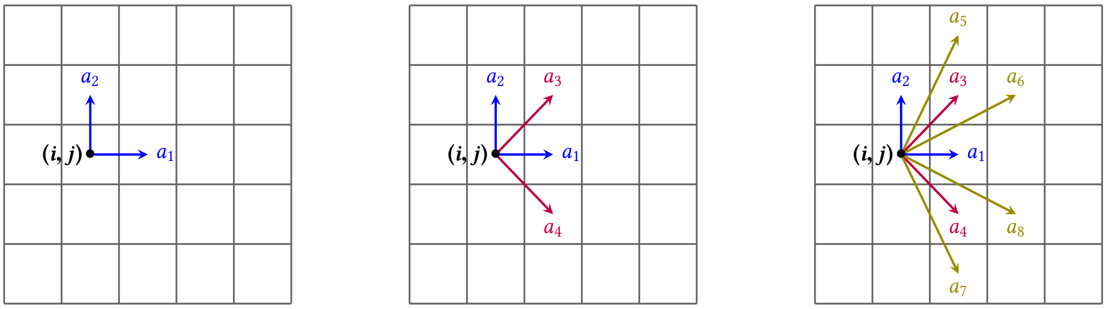
4.2.2. Derivation of the proposed algorithmic schemes

As a first step, we reformulate the minimization problem (4.2) as a constrained problem. To this end, we introduce the S splitting variables $u_1, \dots, u_S$ under the constraint that they are equal, i.e., we consider

$$P_\gamma(u_1, \dots, u_S) \to \min, \qquad \text{s.t.} \quad u_1 = \dots = u_S. \tag{4.5}$$

The functional $P_\gamma(u_1, \dots, u_S)$ of the S variables $u_1, \dots, u_S$ is given by

$$P_\gamma(u_1, \dots, u_S) = \sum_{s=1}^{S} \frac{1}{S} \|Au_s - f\|_2^2 + \gamma \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s} u_s\right\|_0 . \tag{4.6}$$



(a) 4-neighborhood system $\{a_1, a_2\}$ (b) 8-neighborhood system $\{a_1, a_2, a_3, a_4\}$ (c) 16-neighborhood system $\{a_1, \dots, a_8\}$

Figure 4.1.: Illustration of the discrete gradient of the image u at the point (i, j) for different neighborhood systems. The vectors a_s are the offsets of the employed finite differences $u_{(i,j)+a_s} - u_{i,j}$.

It is a basic but important observation that the constrained problem (4.5) and the original problem (4.4) are equivalent. Note that we have overloaded the symbol P_γ which, for one argument u, denotes the discrete Potts functional in (4.4) and for S arguments $u_1, \dots, u_S$ denotes the functional in (4.6). In particular, we have $P_\gamma(u, \dots, u) = P_\gamma(u)$.

Quadratic penalty relaxations of the Potts functional. First, we relax the equality constraints in (4.5) by replacing them by corresponding quadratic penalty terms in the functional. The resulting quadratic penalty relaxation of (4.6) is given by

$$P_{\gamma,\rho}(u_1, \dots, u_S) = \sum_{s=1}^{S} \frac{1}{S} \|Au_s - f\|_2^2 + \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{a_s} u_s\|_0 + \rho \sum_{1 \leq s < s' \leq S} c_{s,s'} \|u_s - u_{s'}\|_2^2. \quad (4.7)$$

The constraints $u_1 = \dots = u_S$ are now part of the functional in the form of soft constraints which are given by the squared Euclidean norms $\sum_{1 \leq s < s' \leq S} c_{s,s'} \|u_s - u_{s'}\|_2^2$. Here, the $c_{s,s'}$ are nonnegative weights (which are possibly zero, if we do not want to directly couple u_s and $u_{s'}$) and $\rho > 0$ denotes a penalty parameter which promotes the soft constraints. Thus, larger values of ρ result in a tighter coupling of the u_s as they are forced to be closer to each other in the ℓ_2-sense. We will later quantify analytically the size of ρ necessary to obtain an a priori prescribed tolerance in the deviations of the u_s; see (4.23) below. For convenience we will frequently use the notation

$$\rho_{s,s'} = \rho \, c_{s,s'}. \quad (4.8)$$

Two typical coupling strategies correspond to coupling all splitting variables or coupling only consecutive splitting variables. They are realized by the following values of the $\rho_{s,s'}$,

$$\rho_{s,s'} = \rho \quad \text{for all } s, s', \qquad \text{or} \qquad \rho_{s,s'} = \rho \, \delta_{(s+1) \bmod S, s'}. \quad (4.9)$$

The first corresponds to the constant choice $c_{s,s'} = 1$. The second corresponds to letting $c_{s,s'} = 1$ if and only if s and s' are (circularly) consecutive which we denote compactly in terms of Kronecker deltas. For both couplings only one additional parameter $\rho > 0$ appears which is tied to the tolerance one is willing to accept on the distances between the u_i; see Algorithm 4.1.

Majorization-minimization approach to quadratic penalty relaxations. Towards a majorization-minimization approach, we first derive a surrogate functional [59] for $P_{\gamma,\rho}$ in (4.7). To this end, we derive a vectorized Tikhonov type formulation of $P_{\gamma,\rho}$, that is,

$$P_{\gamma,\rho}(u_1,\ldots,u_S) = \|B(u_1,\ldots,u_S)^T - g\|_2^2 + \gamma F(u_1,\ldots,u_S)$$

for a matrix B, data g and a prior/regularizing term F. We define the block matrix B and the vector g given by

$$B = \begin{pmatrix}
S^{-1/2}A & 0 & & \cdots & & 0 \\
0 & S^{-1/2}A & & \cdots & & 0 \\
\vdots & & & \ddots & & \vdots \\
0 & 0 & & \cdots & S^{-1/2}A & 0 \\
0 & 0 & & \cdots & 0 & S^{-1/2}A \\
\rho_{1,2}^{1/2} I & -\rho_{1,2}^{1/2} I & 0 & \cdots & 0 & 0 \\
\rho_{1,3}^{1/2} I & 0 & -\rho_{1,3}^{1/2} I & \cdots & 0 & 0 \\
 & \vdots & & & \vdots & \\
\rho_{1,S}^{1/2} I & 0 & 0 & \cdots & 0 & -\rho_{1,S}^{1/2} I \\
0 & \rho_{2,3}^{1/2} I & -\rho_{2,3}^{1/2} I & \cdots & 0 & 0 \\
 & \vdots & & & \vdots & \\
0 & \rho_{2,S}^{1/2} I & 0 & \cdots & 0 & -\rho_{2,S}^{1/2} I \\
 & & & \vdots & & \\
 & & & \vdots & & \\
0 & 0 & 0 & \cdots & \rho_{S-1,S}^{1/2} I & -\rho_{S-1,S}^{1/2} I
\end{pmatrix}, \quad g = \begin{pmatrix}
S^{-1/2}f \\ S^{-1/2}f \\ \vdots \\ S^{-1/2}f \\ S^{-1/2}f \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ 0
\end{pmatrix}. \tag{4.10}$$

Here, I denotes the identity matrix and 0 the zero matrix. In total, the matrix B has S block columns and $S + S(S-1)/2$ block rows. Further, we introduce the difference operator D given by

$$D(u_1,\ldots,u_S) = \begin{pmatrix} \nabla_{a_1} u_1 \\ \vdots \\ \nabla_{a_S} u_S \end{pmatrix} \tag{4.11}$$

which takes the differences in direction a_s of the s-th component u_s of u. By including the weights $\omega_1,\ldots,\omega_S$ we define the quantity $\|D(u_1,\ldots,u_S)\|_{0,\omega}$ which counts the weighted number of jumps via

$$\|D(u_1,\ldots,u_S)\|_{0,\omega} = \sum_{s=1}^{S} \omega_s \, \|\nabla_{a_s} u_s\|_0 \, . \tag{4.12}$$

Now we can rewrite (4.7) as

$$P_{\gamma,\rho}(u_1,\ldots,u_S) = \left\|B(u_1,\ldots,u_S)^T - g\right\|_2^2 + \gamma \left\| D(u_1,\ldots,u_S) \right\|_{0,\omega} \tag{4.13}$$

and the surrogate functional of (4.13) in the sense of [59] is given by

$$\begin{aligned} P^{\mathrm{surr}}_{\gamma,\rho}(u_1,\dots,u_S,v_1,\dots,v_S) = {} & \frac{1}{L_\rho^2}\left\|B(u_1,\dots,u_S)^T - g\right\|_2^2 + \frac{\gamma}{L_\rho^2}\left\| D(u_1,\dots,u_S)\right\|_{0,\omega} \\ & - \frac{1}{L_\rho^2}\left\|B(u_1,\dots,u_S)^T - B(v_1,\dots,v_S)^T\right\|_2^2 + \left\|(u_1,\dots,u_S)^T - (v_1,\dots,v_S)^T\right\|_2^2, \end{aligned} \tag{4.14}$$

where $L_\rho \geq 1$ is a constant larger than the spectral norm $\|B\|_2$ of B (i.e., the operator norm w.r.t. the ℓ_2-norm) which ensures that B/L_ρ is contractive. A sufficient choice of L_ρ in terms of the measurement operator A and the penalties $\rho_{s,s'}$ corresponds to requiring

$$L_\rho^2 > \|A\|_2^2/S + 2\max_{s\in\{1,\dots,S\}} \sum_{s':s'\neq s}^{S} \rho_{s,s'}. \tag{4.15}$$

For the coupling strategies in (4.9) we have sharper estimates. In particular, for the full coupling $\rho_{s,s'} = \rho$ we can choose

$$L_\rho^2 > \|A\|_2^2/S + S\rho \tag{4.16}$$

and when coupling only neighboring u_s for the same constant ρ, we have

$$L_\rho^2 > \|A\|_2^2/S + \alpha\rho, \tag{4.17}$$

where $\alpha = 4$, if S is even, and $\alpha = 2 - 2\cos\left(\frac{\pi(S-1)}{S}\right)$ if S is odd. All these estimates are proven in Lemma 4.9. Basics on surrogate functionals as we need them in this chapter are gathered in Section 4.3.4. Further details on surrogate functionals can be found in [59, 22, 23].

In order to minimize the quadratic penalty relaxation (4.7), we use the surrogate iteration

$$\left(u_1^{(n+1)},\dots,u_S^{(n+1)}\right) \in \operatorname*{argmin}_{u_1,\dots,u_S} P^{\mathrm{surr}}_{\gamma,\rho}(u_1,\dots,u_S,u_1^{(n)},\dots,u_S^{(n)}). \tag{4.18}$$

In the following, we derive a more explicit formulation of (4.18). First, after using elementary properties of the inner product we obtain

$$\begin{aligned} P^{\mathrm{surr}}_{\gamma,\rho}(u_1,\dots,u_S,v_1,\dots,v_S) = {} & \left\|(u_1,\dots,u_S)^T - \left((v_1,\dots,v_S)^T - \frac{1}{L_\rho^2}B^T(B(v_1,\dots,v_S)^T - g)\right)\right\|_2^2 \\ & + \frac{\gamma}{L_\rho^2}\left\| D(u_1,\dots,u_S)\right\|_{0,\omega} + R(v_1,\dots,v_S), \end{aligned} \tag{4.19}$$

where $R(v_1,\dots,v_S)$ is a rest term which is negligible when minimizing P_{surr} w.r.t. $u_1,\dots,u_S$ for fixed $v_1,\dots,v_S$. We write (4.19) in terms of A and the data f as

$$\begin{aligned} & P^{\mathrm{surr}}_{\gamma,\rho}\left(u_1,\dots,u_S,v_1,\dots,v_S\right) \\ & = \sum_{s=1}^{S}\left[\left\|u_s - \left(v_s + \tfrac{1}{SL_\rho^2}A^*f - \tfrac{1}{SL_\rho^2}A^*Av_s - \sum_{s\neq s'}\tfrac{\rho_{s,s'}}{L_\rho^2}(v_s - v_{s'})\right)\right\|_2^2 + \tfrac{\gamma\omega_s}{L_\rho^2}\left\|\nabla_{a_s}u_s\right\|_0\right] + R(v). \end{aligned} \tag{4.20}$$

In view of (4.20), the surrogate iteration (4.18) is given by

$$\left(u_1^{(n+1)}, \dots, u_S^{(n+1)}\right) \in \operatorname*{argmin}_{u_1,\dots,u_S} \sum_{s=1}^{S} \left[\left\| u_s - h_s^{(n)} \right\|_2^2 + \tfrac{\gamma \omega_s}{L_\rho^2} \|\nabla_{a_s} u_s\|_0 \right], \tag{4.21}$$

where $h_s^{(n)}$ is given by

$$h_s^{(n)} = u_s^{(n)} + \tfrac{1}{SL_\rho^2} A^* f - \tfrac{1}{SL_\rho^2} A^* A u_s^{(n)} - \sum_{s':s' \neq s} \tfrac{\rho_{s,s'}}{L_\rho^2} (u_s^{(n)} - u_{s'}^{(n)}), \qquad \text{for all } s \in \{1, \dots, S\}. \tag{4.22}$$

In Section 4.2.3 below, we explain how to compute a minimizer of (4.21) efficiently. Now we assume that a small deviation between the u_s is acceptable, that is,

$$\|u_s - u_{s'}\|_2^2 = \sum_{i,j} |(u_s)_{ij} - (u_{s'})_{ij}|^2 < \tfrac{\varepsilon^2}{c_{s,s'}} \tag{4.23}$$

for $\varepsilon > 0$ and all indices s, s' with $c_{s,s'} \neq 0$. Then the following algorithm computes a result for the quadratic penalty relaxation (4.7) which satisfies (4.23).

Algorithm 4.1. *We consider the quadratic penalty relaxed Potts problem* (4.7) *and a tolerance* ε *for the targets* u_s. *The following algorithm approaches the relaxed Potts problem* (4.7) *and produces a result for which the* u_s *deviate from each other by at most* $\varepsilon/\sqrt{c_{s,s'}}$.

- *Set* ρ *according to* (4.37), *set* L_ρ *according to* (4.15) *(in the special cases of* (4.9), *set* L_ρ *according to* (4.16) *or* (4.17), *respectively.)*

  Set $n = 0$ *and initialize* $u_s^{(0)}$ *as discussed in the corresponding paragraph below, (e.g.,* $u_s^{(0)} = 0$ *for all* s*).*

- *Iterate until convergence:*

  1. $h_s^{(n)} = u_s^{(n)} + \tfrac{1}{SL_\rho^2} A^* f - \tfrac{1}{SL_\rho^2} A^* A u_s^{(n)} - \sum_{s':s' \neq s} \tfrac{\rho_{s,s'}}{L_\rho^2} (u_s^{(n)} - u_{s'}^{(n)}), \quad s = 1, \dots, S,$

  2. $$\left(u_1^{(n+1)}, \dots, u_S^{(n+1)}\right) \in \operatorname*{argmin}_{u_1,\dots,u_S} \sum_{s=1}^{S} \left[\left\| u_s - h_s^{(n)} \right\|_2^2 + \tfrac{\gamma \omega_s}{L_\rho^2} \|\nabla_{a_s} u_s\|_0 \right]. \tag{4.24}$$

By Theorem 4.5 below, Algorithm 4.1 converges to a local minimizer of the quadratic penalty relaxation (4.7) and the components u_s are ε-close, i.e., (4.23) is fulfilled.

A projection procedure for solutions of the quadratic penalty relaxation. As stated above, Algorithm 4.1 produces a local minimizer of the quadratic penalty relaxation (4.7) (Theorem 4.5) of the Potts problem (4.6) and the variables of the produced result are close up to an a priori chosen tolerance. (In practice, this may be already sufficient.) However, a local minimizer of the quadratic penalty relaxation (4.7) is not a feasible solution of the Potts problem (4.6).

In the following, we explain a projection procedure which produces a feasible solution for the Potts problem (4.6) from a local minimizer of (4.7) with variables u_s close to each other (as produced by Algorithm 4.1). Related theoretical results are stated as Theorem 4.6. In particular, it will turn out that in case of a lower bounded imaging operator A, the projection procedure yields a feasible point. Furthermore, the hereby obtained result is close to a local minimizer of the original Potts problem (4.6).

In order to explain the procedure, we need some notions on partitionings of a discrete image domain Ω. We recall that a partitioning $\mathcal{P}$ consists of a finite number $N_{\mathcal{P}}$ of segments $\mathcal{P}_i$ which are pairwise disjoint sets of pixel coordinates whose union equals the image domain Ω, i.e.,

$$\cup_{i=1}^{N_{\mathcal{P}}} \mathcal{P}_i = \Omega, \qquad \mathcal{P}_i \cap \mathcal{P}_j = \emptyset \quad \text{for all } i, j = 1, \ldots, N_{\mathcal{P}}. \tag{4.25}$$

Here, we assume that each segment $\mathcal{P}_i$ is connected w.r.t. the neighborhood system $a_1, \ldots, a_S$, i.e., for any two elements in $\mathcal{P}_i$ there is a path with steps in $a_1, \ldots, a_S$ which connects them.

We will need the following proposed notion of a *directional partitioning*.

Definition 4.2. A *directional partition* $\mathcal{I}$ of a discrete domain Ω w.r.t. a set of S directions $a_1, \ldots, a_S$ is a set of (discrete) intervals I. Each interval I is associated with exactly one of the directions $a_s \in \{a_1, \ldots, a_S\}$ and is of the form $I = \{(i, j) + k a_s : k = 0, \ldots, K-1\}$, where $K \in \mathbb{N}$ and we require $I \subset \Omega$. Further, for each (single) direction a_s, the corresponding intervals have to form an ordinary partition.

We note that a result $u = (u_1, \ldots, u_S) : \Omega \to \mathbb{R}^s$ of Algorithm 4.1 induces a directional partitioning: each variable u_s is associated with a direction a_s and for any $s \in \{1, \ldots, S\}$, we let each (maximal) interval of constance of u_s be an interval in $\mathcal{I}$ associated with a_s.

Each partitioning $\mathcal{P}$ induces a directional partitioning $\mathcal{I}$ by letting the intervals I of $\mathcal{I}$ be the stripes with direction a_s obtained from segment $\mathcal{P}_i$ for each direction $s = 1, \ldots, S$ and each segment $\mathcal{P}_i$, $i = 1, \ldots, N_{\mathcal{P}}$.

Conversely, each directional partitioning $\mathcal{I}$ induces a partitioning by the following merging process.

Definition 4.3. Two pixels x, y are related, in symbols, $x \sim y$, if they are connected by a path $x_0 = x, \ldots, x_N = y$ such that for any consecutive members x_i, x_{i+1}, $i = 1, \ldots, N-1$, of the path there is an interval I of the directional partitioning $\mathcal{I}$ which contains x_i and x_{i+1}.

Obviously, the relation $x \sim y$ is an equivalence relation and the equivalence classes $\mathcal{P}_i$ form a partitioning on Ω. We denote by

$$\mathcal{I}(\mathcal{P}) = \mathcal{I}^{\mathcal{P}}, \qquad \mathcal{P}(\mathcal{I}) = \mathcal{P}_{\mathcal{I}}, \tag{4.26}$$

the mappings which assign a directional partitioning to a partitioning and vice versa, respectively.

Finally, we define for a function $u = (u_1, \ldots, u_S) : \Omega \to \mathbb{R}^s$ and a partitioning $\mathcal{P}$ of Ω, the projection to a function $\pi_{\mathcal{P}}(u) : \Omega \to \mathbb{R}$ by

$$\pi_{\mathcal{P}}(u)|_{\mathcal{P}_i} = \frac{\sum_{x \in \mathcal{P}_i} \sum_{s=1}^{S} u_s(x)}{S \, \#\mathcal{P}_i}. \tag{4.27}$$

Here, $\#\mathcal{P}_i$ denotes the number of elements in the segment $\mathcal{P}_i$. The projection π defined in (4.27) averages the components u_s of u and all members of the segment $\mathcal{P}_i$ and thereby produces a piecewise constant function w.r.t. the partitioning $\mathcal{P}$.

We now propose the following projection procedure.

Procedure 4.1 (Projection Procedure). *We consider output* $u = (u_1, \dots, u_S) : \Omega \to \mathbb{R}^s$ *of Algorithm 4.1 together with its induced directional partitioning* $\mathcal{I}$*.*

1. *Compute the partitioning* $\mathcal{P}(\mathcal{I}) = \mathcal{P}_{\mathcal{I}}$ *induced by the directional partitioning* $\mathcal{I}$ *as explained above* (4.26)*.*
2. *Project* $u = (u_1, \dots, u_S) : \Omega \to \mathbb{R}^s$ *to* $\pi_{\mathcal{P}_{\mathcal{I}}}(u)$ *using* (4.27) *for the partitioning* $\mathcal{P}(\mathcal{I}) = \mathcal{P}_{\mathcal{I}}$*, and return* $\pi_{\mathcal{P}_{\mathcal{I}}}(u)$ *as output.*

We remark that the second step might be replaced by solving the normal equations w.r.t. A and f in the space of functions constant on $\mathcal{P}_{\mathcal{I}}$ which, however, might be more expensive.

An approach to the Potts problem based on the majorization-minimization approach to its quadratic penalty relaxation. The coupling parameter ρ controls the similarity of the components u_s, i.e., a large value of ρ enforces the u_s to be close to each other and in the limit case $\rho \to \infty$, they become equal and yield a solution to the Potts problem (4.4). As a consequence, increasing the parameters ρ during the iterations should tie the u_s closer together such that the constraint of (4.5) should be ultimately fulfilled and we obtain an approach for the initial Potts problem (4.4). Recall that $\rho_{s,s'} = \rho\, c_{s,s'}$ was defined by (4.8), where the $c_{s,s'}$ are nonnegative numbers weighting the constraints. The coupling parameter ρ is increased in the course of the iterations, while the $c_{s,s'}$ are fixed during this process. We sum up the approach in the following algorithm.

Algorithm 4.2. *We consider the Potts problem* (4.5) *in* S *variables (which is equivalent to* (4.4) *as explained above). We propose the following algorithm for the Potts problem* (4.5)*.*

Let $\rho^{(k)}$ *be a strictly increasing sequence (e.g.,* $\rho^{(k)} = \tau^k \rho^{(0)}$*, with* $\rho_0, \tau > 1$*) and* $\delta_k \to 0$ *be a strictly decreasing sequence converging to zero (e.g.,* $\delta_k = \delta_0/\tau^k$*.) Further, let*

$$t > 2\sigma_1^{-1/2} S^{-1/2} \|A\| \, \|f\|, \tag{4.28}$$

where σ_1 *is the smallest non-zero eigenvalue of* $C^T C$ *with* C *given by* (4.52)*. For the particular choice of coupling given by the left-hand and right hand side of* (4.9) *we let*

$$t > \tfrac{2}{S} \|A\| \, \|f\|, \qquad \textit{and} \qquad t > 2(2 - 2\cos(2\pi/S))^{-1/2} S^{-1/2} \|A\| \, \|f\|, \qquad \textit{respectively.} \tag{4.29}$$

- *Initialize* $u_s^{(0)} := u_s^{(0,0)}$ *as discussed in the corresponding paragraph below (e.g.,* $u_s^{(0)} = 0$ *for all* s*).*

  Set $\rho = \rho^{(0)}$, $\rho_{s,s'} = \rho^{(0)} c_{s,s'}$, $\delta = \delta_0$, $k, n = 0$; *set* L_ρ *according to* (4.15) *(or, in the special cases of* (4.9)*, as explained below* (4.15)*).*

(i) While

$$\left\|u_s^{(k,n)} - u_{s'}^{(k,n)}\right\| > \frac{t}{\rho\sqrt{c_{s,s'}}}, \quad \textit{or} \quad \left\|u_s^{(k,n)} - u_s^{(k,n-1)}\right\| > \frac{\delta}{L_\rho} \tag{4.30}$$

do

1. $h_s^{(k,n)} = u_s^{(k,n)} + \frac{1}{SL_\rho^2}A^* f - \frac{1}{SL_\rho^2}A^* A u_s^{(k,n)} - \sum_{s':s' \neq s} \frac{\rho_{s,s'}}{L_\rho^2}(u_s^{(k,n)} - u_{s'}^{(k,n)}), \quad s = 1, \dots, S,$

2. $$\left(u_1^{(k,n+1)}, \dots, u_S^{(k,n+1)}\right) \in \operatorname*{argmin}_{u_1,\dots,u_S} \sum_{s=1}^{S} \left[\left\|u_s - h_s^{(k,n)}\right\|_2^2 + \frac{\gamma\omega_s}{L_\rho^2} \|\nabla_{a_s} u_s\|_0\right], \tag{4.31}$$

and set $n = n + 1$.

(ii) Set

$$u_s^{(k+1)} = u_s^{(k+1,0)} = u_s^{(k,n)}, \tag{4.32}$$

set $k = k + 1, n = 0$, and let $\rho = \rho^{(k)}, \rho_{s,s'} = \rho^{(k)} c_{s,s'}, \delta = \delta_k$; set L_ρ according to (4.15) *(or, in the special cases of* (4.9)*, as below* (4.15)*) and goto (i).*

This approach is inspired by the quadratic penalty methods of [142] for sparsity problems, that is, searching for a solution with only a few nonzero entries. There, the corresponding prior is separable, while we consider a non-separable prior.

Initialization. Although the initialization of Algorithm 4.1 and of Algorithm 4.2 is not relevant for its convergence properties (cf. Section 4.3), the initialization influences the final result. We discuss different strategies. The simplest choice is the all-zero initialization $(u_1^{(0)}, \dots, u_s^{(0)}) = (0, \dots, 0)$. Furthermore, one can select the right hand side of the normal equations of the underlying least squares problem, that is $A^T f$. A third reasonable choice is the solution of the normal equations itself or an approximation of it. Using an approximation might especially be reasonable to receive a regularized approximation of the normal equations. A possible strategy to obtain such a regularized initialization is to apply a fixed number of Landweber iterations [129] or of the conjugate gradient method to the underlying least square problem. (In our experiments, we initialized Algorithm 4.1 with the result of 1000 Landweber iterations and Algorithm 4.2 with $A^T f$.)

4.2.3. Solving the subproblems

In both proposed algorithms we have to solve the Potts subproblem (4.21) in the backward step, see (4.24) and (4.31), respectively. We first observe that (4.21) can be solved for each of the u_s separately and the corresponding s minimization problems have the prototypical form

$$\operatorname*{argmin}_{u_s:\Omega\to\mathbb{R}} \|u_s - f\|_2^2 + \gamma_s' \|\nabla_{a_s} u\|_0 \tag{4.33}$$

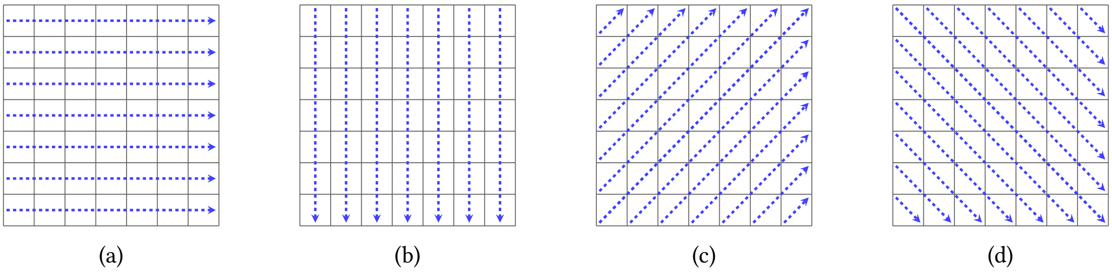

(a) (b) (c) (d)

Figure 4.2.: Decomposition of the surrogate iteration (4.21) into univariate Potts problems along the paths induced by th directions a_s. The specific paths depend on the directions used for discretizing the gradient. (a)-(b): anisotropic discretization. (a)-(d): near-isotropic discretization.

for given data f, the jump penalty $\gamma_s' = \frac{\gamma \omega_s}{L_\rho^2} > 0$ and the direction $a_s \in \mathbb{Z}^2$. Next, we see that (4.33) decomposes into univariate Potts problems for data along the paths in f induced by a_s. For example, the direction $a_s = e_1$ induces paths which correspond to the rows of f and a minimizer u_s^* of (4.33) is obtained by determining each of its rows individually. An illustration of the paths for the 8-neighborhood discretization is given in Figure 4.2. The univariate Potts problem corresponds to

$$P_\gamma^{\mathrm{id},1\mathrm{d}}(x) = \|x - g\|_2^2 + \gamma \|\nabla x\|_0 \to \min, \tag{4.34}$$

where the data g is given by the restriction of f to the pixels in Ω of the form $v + a_s\mathbb{Z}$, i.e., $g = f(v + a_s \cdot)$. Here the offset v is fixed when solving each univariate problem, but varied afterwards to get all lines in the image with direction a_s. The target to optimize is denoted by $x \in \mathbb{R}^n$ and, in the resulting univariate situation, $\|\nabla x\|_0 = |\{i : x_i \neq x_{i+1}\}|$ denotes the number of jumps of x. The univariate Potts problem can be solved with Algorithm 3.1 of Chapter 3, so that the subproblems of the proposed algorithms are solved exactly and efficiently.

4.3. Analysis

4.3.1. Analytic results

We started by considering the quadratic penalty relaxation (4.7) of the multivariate Potts problem when we derived the proposed algorithms. Even though it is more straight-forward to access algorithmically via our approach, we first note that this problem is still NP-hard (as is the original problem).

Theorem 4.4. *Finding a (global) minimizer of the quadratic penalty relaxation* (4.7) *of the multivariate Potts problem is an NP-hard problem.*

The proof is given in Section 4.3.3 below. In Section 4.2.2, we have proposed Algorithm 4.1 to approach the quadratic penalty relaxation of the multivariate Potts problem. We show that Algorithm 4.1 converges to a local minimizer of the quadratic penalty relaxation and that a feasible point of the original multivariate Potts problem is nearby.

Theorem 4.5. *We consider the iterative Potts minimization Algorithm 4.1 for the quadratic penalty relaxation* (4.7) *of the multivariate Potts problem.*

(i) Algorithm 4.1 computes a local minimizer of the quadratic penalty relaxation (4.7) *of the multivariate Potts problem for any starting point. The convergence rate is linear.*

(ii) We have the following relation between local minimizers $\mathcal{L}$, global minimizers $\mathcal{G}$ and the fixed points $\mathrm{Fix}(\square)$ *of the iteration of Algorithm 4.1,*

$$\mathcal{G} \subset \mathrm{Fix}(\square) \subset \mathcal{L}. \tag{4.35}$$

(iii) Assume a tolerance ε we are willing to accept for the distance between the u_s, i.e.,

$$\sum_{s,s'} c_{s,s'} \|u_s - u_{s'}\|_2^2 = \sum_{s,s'} c_{s,s'} \sum_{i,j} |(u_s)_{ij} - (u_{s'})_{ij}|^2 \leq \varepsilon^2. \tag{4.36}$$

Running Algorithm 4.1 with the choice of the parameter ρ by

$$\rho > 2\varepsilon^{-1}\, \sigma_1^{-1/2} S^{-1/2} \|A\| \|f\| \tag{4.37}$$

(where σ_1 is the smallest non-zero eigenvalue of $C^T C$ with C given by (4.52)*; for the particular choice of the coupling given by* (4.9)*, $\sigma_1 = S$ and $\sigma_1 = (2 - 2\cos(2\pi/S))$, respectively) yields a local minimizer of the quadratic penalty relaxation* (4.7) *such that the u_s are close up to ε, i.e.,* (4.36) *is fulfilled.*

The proof is given in Section 4.3.5 below. We recall that a solution of Algorithm 4.1 is not a feasible point for the initial Potts problem (4.5). However, it produces a δ-approximative solution u^*, i.e., there is μ^* and a partitioning $\mathcal{P}^*$ such that

$$\sum_{s,s'} c_{s,s'} \|u_s^* - u_{s'}^*\|_2^2 < \delta, \qquad \text{and} \qquad L(\mu^*) < \delta, \tag{4.38}$$

where $L(\mu^*)$ is given by (4.56) below. In this context, we note that the conditions for a local minimizer of (4.5) are given by $\sum_{s,s'} c_{s,s'} \|u_s^* - u_{s'}^*\|_2^2 = 0$ and the Lagrange multiplier condition $L(\mu^*) = 0$. Hence, (4.38) intuitively means that both the constraint and the Lagrange multiplier condition are approximately fulfilled for the partitioning induced by u^*. Further, a feasible point for the Potts problem (4.5) (or, equivalently, (4.4)) is nearby which is detailed in the following theorem.

Theorem 4.6. *We consider the iterative Potts minimization Algorithm 4.1 for the quadratic penalty relaxation* (4.7) *in connection with the (non-relaxed) Potts problem* (4.5).

(i) Algorithm 4.1 produces an approximative solution w.r.t. (4.38) *of the Potts problem* (4.5).

(ii) The projection procedure of Section 4.2.2 (which was formulated as Procedure 4.1) applied to the solution $u' = (u'_1, \ldots, u'_S)$ of Algorithm 4.1 produces a feasible image $\hat{u}$ (and a valid partitioning) for the Potts problem (4.5) *which is close to u' in the sense that*

$$\|u'_s - \hat{u}\| \leq C_1 \varepsilon \qquad \text{for all} \quad s \in \{1, \ldots, S\}, \tag{4.39}$$

where $\varepsilon = \max_{s,s'} \|u'_s - u'_{s'}\|$ *quantifies the deviation between the components* u_s *and* $C_1 = \#\Omega/4$. *If the imaging operator* A *is lower bounded, i.e., there is a constant* $c > 0$ *such that* $\|Au\| \geq c\|u\|$, *a local minimizer* u^* *of the Potts problem* (4.5) *is nearby, i.e.,*

$$\|u^* - \hat{u}\| \leq \frac{\sqrt{\eta}}{c} \tag{4.40}$$

where

$$\eta := \left(\|A\|^2 \varepsilon C_1^2 + 2\|A\| C_1 \|f\|_2\right) \varepsilon. \tag{4.41}$$

The proof of Theorem 4.6 can be found at the end of Section 4.3.4, where most relevant statements are shown before in Section 4.3.3. Theorem 4.6 is a theoretical justification that in practice we may use Algorithm 4.1 for the Potts problem (4.5) (for an arbitrary small tolerance we may fix in advance).

In Section 4.2.2, we have proposed Algorithm 4.2 to approach the Potts problem (4.5). We first show that Algorithm 4.2 is well-defined.

Theorem 4.7. *Algorithm 4.2 is well-defined, that is, the inner iteration governed by* (4.30) *terminates. More precisely, for any* $k \in \mathbb{N}$, *there is* $n \in \mathbb{N}$ *such that the termination criterium given by* (4.30) *holds.*

The proof of Theorem 4.7 is given in Section 4.3.6. We obtain the following results concerning the convergence properties of Algorithm 4.2.

Theorem 4.8. *We consider the iterative Potts minimization algorithm (Algorithm 4.2) for the Potts problem* (4.5).

- *Any cluster point of the sequence* $u^{(k)}$ *is a local minimizer of the Potts problem* (4.5) *(which implies that the components of each limit* u^* *are equal, i.e.,* $u^*_s = u^*_{s'}$ *for all* s, s'.*)*
- *If* A *is lower bounded, the sequence* $u^{(k)}$ *produced by Algorithm 4.2 has a cluster point and the produced cluster points are local minimizers of the Potts problem* (4.5).

The proof of Theorem 4.8 can be found in Section 4.3.6.

4.3.2. Estimates on operator norms and Lagrange multipliers

Lemma 4.9. *The spectral norm of the block matrix* B *given by* (4.10) *satisfies*

$$\|B\|_2 \leq \left(\tfrac{1}{S}\|A\|_2^2 + 2 \max_{s \in \{1,\ldots,S\}} \sum_{s':s' \neq s}^{S} \rho_{s,s'}\right)^{\frac{1}{2}}. \tag{4.42}$$

For the particular choice of constant $\rho_{s,s'} = \rho$ *(independent of* s, s'*) as on the left-hand side of* (4.9) *we have the improved estimate*

$$\|B\|_2 \leq \left(\tfrac{1}{S}\|A\|_2^2 + S\rho\right)^{\frac{1}{2}}. \tag{4.43}$$

For only coupling neighboring u_s with the same constant ρ, i.e., the right-hand coupling of (4.9), *we have*

$$\|B\|_2 \leq \left(\tfrac{1}{S}\|A\|_2^2 + \alpha\rho\right)^{\frac{1}{2}}, \qquad \textit{where} \quad \alpha = \begin{cases} 4, & \textit{if } S \textit{ is even,} \\ 2 - 2\cos\left(\frac{\pi(S-1)}{S}\right), & \textit{if } S \textit{ is odd.} \end{cases} \tag{4.44}$$

Proof. We write the matrix B in block form $B = \begin{pmatrix} S^{-1/2}\tilde{A} \\ \tilde{P} \end{pmatrix}$. Here $\tilde{A}$ denotes an $S \times S$- block diagonal matrix with each diagonal entry being equal to A, where A is the matrix representing the forward/imaging operator; see (4.10). The matrix $\tilde{P}$ is given as the lower $\binom{S}{2} \times S$- block in (4.10) which represents the soft constraints.

Using this decomposition of B, we may decompose the symmetric and positive (semidefinite) matrix $B^T B$ according to

$$B^T B = \tfrac{1}{S}\tilde{A}^T\tilde{A} + \tilde{P}^T\tilde{P}, \tag{4.45}$$

where $\tilde{A^T}\tilde{A}$ is an $S \times S$- block diagonal matrix with each diagonal entry being equal to $A^T A$, and $\tilde{P}^T\tilde{P}$ is an $S \times S$- block diagonal matrix with block entries given by

$$\tilde{P}^T\tilde{P} = \begin{pmatrix} \sum_{k=2}^{S}\rho_{1,k}I & -\rho_{1,2}I & -\rho_{1,3}I & \dots & -\rho_{1,S}I \\ -\rho_{1,2}I & \sum_{k=1,k\neq 2}^{S}\rho_{2,k}I & -\rho_{2,3}I & \dots & -\rho_{2,S}I \\ & \vdots & & & \vdots \\ -\rho_{1,S}I & -\rho_{2,S}I & -\rho_{3,S}I & \dots & \sum_{k=1}^{S-1}\rho_{S,k}I \end{pmatrix}, \tag{4.46}$$

with $\rho_{l,k} := \rho_{k,l}$ for $l > k$. Using Gerschgorin's Theorem (see for instance [178]), the eigenvalues of $\tilde{P}$ are contained in the union of the balls with center $x_r = \sum_{k=1,k\neq r}^{S}\rho_{r,k}$ and radius $x_r = \sum_{k=1,k\neq r}^{S}|-\rho_{r,k}|$. These balls are all contained in the larger ball with center 0 and radius $2 \cdot \max_r x_r$. This implies (4.42).

For seeing (4.43) we decompose an argument $u = (u_1, \dots, u_S)$ according to $u = \bar{u} + u^0$ with an "average" part $\bar{u} = (\frac{1}{S}\sum_{i=1}^{S}u_i, \dots, \frac{1}{S}\sum_{i=1}^{S}u_i)$ and $u^0 := u - \bar{u}$ such that u^0 has average 0, i.e., $\sum_{i=1}^{S}u_i^0 = 0$, where 0 denotes the vector containing only zero entries here. In the situation of (4.43), the matrix $\tilde{P}^T\tilde{P}$ has the form $\tilde{P}^T\tilde{P} = \rho(S \cdot I - (1, \dots, 1)(1, \dots, 1)^T)$ We have $\tilde{P}^T\tilde{P}\bar{u} = 0$. Further, $\tilde{P}^T\tilde{P}u^0 = \rho S u^0$. Hence, the largest modulus of an eigenvalue of $\tilde{P}^T\tilde{P}$ equals ρS which in turn shows the estimate (4.43).

For seeing (4.44), we notice that in case of (4.44), the matrix $\tilde{P}^T\tilde{P}$ has cyclic shift structure with three nonzero entries in each line. The discrete Fourier matrix w.r.t. the cyclic group of order S diagonalizes $\tilde{P}^T\tilde{P}$. The corresponding eigenvalues are given by $\lambda_k = \rho\left(2 - 2\cos\left(2\pi\frac{k}{S}\right)\right)$, where $k = 0, \dots, S-1$. Thus, the largest modulus of an eigenvalue given by 4 ρ, if S is even, and by $\rho \cdot \left(2 - 2\cos\left(\frac{\pi(S-1)}{S}\right)\right)$ otherwise. □

We recall that we have defined the notion of a directional partitioning $\mathcal{I}$ (cf. Definition 4.2) and discussed its relation with the concept of a partitioning near (4.26). For a function $f : \Omega \to \mathbb{R}^S$ (representing S component functions $f_1, \dots, f_S : \Omega \to \mathbb{R}$) defined on a discrete domain Ω we consider the orthogonal projection $P_{\mathcal{I}}$ associated with a directional partition $\mathcal{I}$. First we sort the intervals $\mathcal{I}$ into $\mathcal{I}_1, \ldots, \mathcal{I}_S$ according to their associated directions a_s, $s = 1, \ldots, S$, and then we let

$$P_{\mathcal{I}} f = \begin{pmatrix} P_{\mathcal{I}_1} f_1 \\ \vdots \\ P_{\mathcal{I}_S} f_S \end{pmatrix}, \qquad \text{where} \quad P_{\mathcal{I}_s} f_s|_I = \frac{\sum_{x \in I} f_s(x)}{\# I}, \tag{4.47}$$

i.e., the function $P_{\mathcal{I}_s} f_s$ on the interval I is given by the arithmetic mean of f_s on the interval I for all intervals $I \in \mathcal{I}_s$, and for all $s = 1, \ldots, S$. Here, $\# I$ denotes the number of elements in I. We note that $P_{\mathcal{I}}$ defines an orthogonal projection on the corresponding ℓ_2-space of discrete functions $f : \Omega \to \mathbb{R}^S$ with the norm $\|f\|^2 = \sum_{s,i} |(f_s)_i|^2$ where i iterates through all the indices of f_s.

We consider a partitioning $\mathcal{P}$ of Ω, its induced directional partitioning $\mathcal{I}^{\mathcal{P}}$ w.r.t. a set of S directions $a_1, \ldots, a_S$, and define the subspace

$$\mathcal{A}^{\mathcal{P}} = P_{\mathcal{I}^{\mathcal{P}}}(\ell_2(\Omega, \mathbb{R}^S)) \tag{4.48}$$

of functions which are constant on the intervals of the induced directional partitioning $\mathcal{I}^{\mathcal{P}}$ (which are invariant w.r.t. the orthogonal projection $P_{\mathcal{I}^{\mathcal{P}}}$).

The functions $g : \Omega \to \mathbb{R}$ which are piecewise constant w.r.t. a partitioning $\mathcal{P}$, i.e., they are constant on each segment $\mathcal{P}_i$, are in one-to-one correspondence with the linear subspace $\mathcal{B}^{\mathcal{P}}$ of $\mathcal{A}^{\mathcal{P}}$ given by

$$\mathcal{B}^{\mathcal{P}} = \left\{ f \in \mathcal{A}^{\mathcal{P}} : f_1 = \ldots = f_S \right\} \tag{4.49}$$

which we show in the following lemma.

Lemma 4.10. *There is a one-to-one correspondence between the linear space of piecewise constant functions w.r.t. the partitioning $\mathcal{P}$, and the subspace $\mathcal{B}^{\mathcal{P}}$ of $\mathcal{A}^{\mathcal{P}}$ via the mapping $\iota : g \mapsto (g, \ldots, g)$.*

Proof. Let g be a piecewise constant mapping w.r.t. the partitioning $\mathcal{P}$. Then $(g, \ldots, g)$ is constant on each interval I of the induced directional partitioning $\mathcal{I}^{\mathcal{P}}$, and $(g, \ldots, g) \in \mathcal{B}^{\mathcal{P}}$. This shows that ι is well-defined in the sense that its range is contained in $\mathcal{B}^{\mathcal{P}}$. Obviously, ι is an injective linear mapping so that it remains to show that any $f \in \mathcal{B}^{\mathcal{P}}$ is the image under ι of some $g : \Omega \to \mathbb{R}$ which is piecewise constant w.r.t. the partitioning $\mathcal{P}$. To this end, let $f \in \mathcal{B}^{\mathcal{P}}$. By definition, f has the form $f = (g, \ldots, g)$ for some $g : \Omega \to \mathbb{R}$. Now, towards a contradiction, assume there is a segment $\mathcal{P}_i$ and points $x, y \in \mathcal{P}_i$ with $g(x) \neq g(y)$. Since there is a path $x_0 = x, \ldots, x_N = y$ connecting x, y in $\mathcal{P}_i$ with steps in $a_1, \ldots, a_S$, we have that for any i there is an interval I in the induced partitioning $\mathcal{I}^{\mathcal{P}}$ containing x_i together with x_{i+1}. Since g is constant on each I in $\mathcal{I}^{\mathcal{P}}$ we get $g(x_i) = g(x_{i+1})$ for all i which implies $g(x) = g(y)$. This contradicts our assumption and shows the lemma. □

We use the identification given by Lemma 4.10 to define, for a given partitioning $\mathcal{P}$, the projection $Q_{\mathcal{P}}$ onto $\mathcal{B}^{\mathcal{P}}$ by

$$Q_{\mathcal{P}} f = \begin{pmatrix} \pi_{\mathcal{P}} f \\ \vdots \\ \pi_{\mathcal{P}} f \end{pmatrix}, \qquad \text{where} \quad \pi f|_{\mathcal{P}_i} = \frac{\sum_{s=1}^{S} \sum_{x \in \mathcal{P}_i} f_s(x)}{\# \mathcal{P}_i \, S}, \tag{4.50}$$

i.e., taking the average w.r.t. the segment and all component functions as given by (4.27). As the components of $Q_{\mathcal{P}} f$ are all identical, we will not distinguish between $Q_{\mathcal{P}}$ and $\pi_{\mathcal{P}}$ in the following. Hence, we also use the symbol $Q_{\mathcal{P}} f$ to denote the scalar-valued function which is piecewise constant on the partitioning $\mathcal{P}$.

On the space $\mathcal{A}^{\mathcal{P}}$, we consider the following problem

$$\operatorname*{argmin}_{u_1,\ldots,u_S} \sum_{s=1}^{S} \frac{1}{S} \|Au_s - f\|_2^2 \qquad \text{subject to} \qquad Cu = 0, \tag{4.51}$$

which corresponds to searching for a solution which belongs to $\mathcal{B}^{\mathcal{P}}$ given the directional partitioning. Here, C denotes the constraint matrix

$$C = \begin{pmatrix} c_{1,2}I & -c_{1,2}I & 0 & \ldots & 0 & 0 \\ c_{1,3}I & 0 & -c_{1,3}I & \ldots & 0 & 0 \\ & \vdots & & & \vdots & \\ c_{1,S}I & 0 & 0 & \ldots & 0 & -c_{1,S}I \\ 0 & c_{2,3}I & -c_{2,3}I & \ldots & 0 & 0 \\ & \vdots & & & \vdots & \\ 0 & c_{2,S}I & 0 & \ldots & 0 & -c_{2,S}I \\ & & & \vdots & & \\ 0 & 0 & 0 & \ldots & c_{S-1,S}I & -c_{S-1,S}I \end{pmatrix}, \tag{4.52}$$

where the $c_{s,s'}$ are as in (4.7); in particular, if $c_{s,s'} = 0$, the corresponding row is removed from the constraint matrix C. For the special choices of (4.9), we have

$$C = \begin{pmatrix} I & -I & 0 & \ldots & 0 & 0 \\ I & 0 & -I & \ldots & 0 & 0 \\ & \vdots & & & \vdots & \\ I & 0 & 0 & \ldots & 0 & -I \\ 0 & I & -I & \ldots & 0 & 0 \\ \vdots & & & & \vdots & \\ 0 & I & 0 & \ldots & 0 & -I \\ & & & \vdots & & \\ 0 & 0 & 0 & \ldots & I & -I \end{pmatrix}, \quad \text{and} \quad C = \begin{pmatrix} I & -I & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & I & -I & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & I & -I & \ldots & 0 & 0 & 0 \\ & \vdots & & & & \vdots & & \\ 0 & 0 & 0 & 0 & \ldots & I & -I & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & I & -I \\ -I & 0 & 0 & 0 & \ldots & 0 & 0 & I \end{pmatrix} \tag{4.53}$$

which reflects the constraints $u_1 = \ldots = u_S$. We recall that $\mu_{\mathcal{P}}$ is a Lagrange multiplier of the problem in (4.51) if

$$\min_{u \in \mathcal{B}^{\mathcal{P}}} \sum_{s=1}^{S} \frac{1}{S} \|Au_s - f\|_2^2 = \min_{u \in \mathcal{A}^{\mathcal{P}}} \sum_{s=1}^{S} \frac{1}{S} \|Au_s - f\|_2^2 + \mu_{\mathcal{P}}^T Cu. \tag{4.54}$$

We note that for quadratic problems such as in (4.51) Lagrange multipliers always exist [15]. We have the identity

$$\tfrac{2}{S} P_{\mathcal{I}^{\mathcal{P}}} \tilde{A}^T \tilde{A} P_{\mathcal{I}^{\mathcal{P}}} u^*_{\mathcal{P}} - \tfrac{2}{S} P_{\mathcal{I}^{\mathcal{P}}} \tilde{A}^T \tilde{f} = P_{\mathcal{I}^{\mathcal{P}}} C^T \mu_{\mathcal{P}} = C^T \mu_{\mathcal{P}}, \tag{4.55}$$

or, equivalently,

$$L(\mu_{\mathcal{P}}) := \left\| \tfrac{2}{S} P_{\mathcal{I}^{\mathcal{P}}} \tilde{A}^T \tilde{A} P_{\mathcal{I}^{\mathcal{P}}} u^*_{\mathcal{P}} - \tfrac{2}{S} P_{\mathcal{I}^{\mathcal{P}}} \tilde{A}^T \tilde{f} - P_{\mathcal{I}^{\mathcal{P}}} C^T \mu_{\mathcal{P}} \right\| = 0, \tag{4.56}$$

where $\tilde{A}$ is the block diagonal matrix with constant entry A on each diagonal component, $\tilde{f} = (f^T, \ldots, f^T)^T$ is a block vector of corresponding dimensions with entry f in each component, and $u^*_{\mathcal{P}}$ is a minimizer of the constraint problem on $\mathcal{B}^{\mathcal{P}}$. We note that the last equality $C^T \mu_{\mathcal{P}} = P_{\mathcal{I}^{\mathcal{P}}} C^T \mu_{\mathcal{P}}$ in (4.55) holds as the left-hand side of (4.55) is contained in the image of $P_{\mathcal{I}^{\mathcal{P}}}$.

The following lemma gives estimates on the norm of the Lagrange multiplier $\mu_{\mathcal{P}}$.

Lemma 4.11. *We consider a partitioning $\mathcal{P}$ of the discrete domain Ω, and the corresponding quadratic problem* (4.51)*. There is a Lagrange multiplier $\mu_{\mathcal{P}}$ for* (4.51) *with*

$$\|\mu_{\mathcal{P}}\| \leq 2\sigma_1^{-1/2} S^{-1/2} \|A\| \|f\|, \tag{4.57}$$

where σ_1 is the smallest nonzero eigenvalue of $C^T C$ for C given by (4.52)*. For the particular choice of C given by the left-hand side of* (4.53) *we have*

$$\|\mu_{\mathcal{P}}\| \leq \tfrac{2}{S} \|A\| \|f\|; \tag{4.58}$$

and, for the particular choice of C given by the right-hand side of (4.53) *we have*

$$\|\mu_{\mathcal{P}}\| \leq 2(2 - 2\cos(2\pi/S))^{-1/2} S^{-1/2} \|A\| \|f\|, \tag{4.59}$$

(e.g., for $S = 4$ we have $2 - 2\cos(2\pi/S) = 2$). In particular, the right-hand side and the constants in all these estimates are independent of the particular partitioning $\mathcal{P}$.

Proof. For any minimizer $u^*_{\mathcal{P}}$ of the constraint problem in $\mathcal{B}^{\mathcal{P}}$, we have that

$$\|\tfrac{2}{S} P_{\mathcal{I}^{\mathcal{P}}} \tilde{A}^T \tilde{A} P_{\mathcal{I}^{\mathcal{P}}} u^*_{\mathcal{P}} - \tfrac{2}{S} P_{\mathcal{I}^{\mathcal{P}}} \tilde{A}^T \tilde{f}\| \leq \|\tfrac{2}{S} \tilde{A}^T \tilde{A} u^*_{\mathcal{P}} - \tfrac{2}{S} \tilde{A}^T \tilde{f}\| \leq \tfrac{2}{S} \|A\| \|\tilde{f}\| \leq \tfrac{2\sqrt{S}}{S} \|A\| \|f\|, \tag{4.60}$$

where we recall that $\tilde{A}$ is the block diagonal matrix with constant entry A, and $\tilde{f}$ is a block vector with entry f in each component. The first inequality is a consequence of the fact that $P_{\mathcal{I}^{\mathcal{P}}}$ is an orthogonal projection. The second inequality may be seen by evaluating the functional for the constant zero function (which always belongs to $\mathcal{B}^{\mathcal{P}}$) as a candidate and by noting that $\|A^T\| = \|A\|$.

Using (4.55), it follows from (4.55) that $\|C^T \mu_{\mathcal{P}}\| \leq \frac{2}{\sqrt{S}} \|A\| \|f\|$. Choosing $\mu_{\mathcal{P}}$ in the complement of the zero space of C^T, we get

$$\|C^T \mu_{\mathcal{P}}\| \geq \inf_{x \in \left(\ker\left(C^T\right)\right)^{\perp},\, \|x\|=1} \|C^T x\| \; \|\mu_{\mathcal{P}}\|. \tag{4.61}$$

We observe that finding the infimum in (4.61) corresponds to finding the square root of the smallest nonzero eigenvalue of $C^T C$. This is because (i) the nonzero eigenvalues of $C^T C$ equal the nonzero eigenvalues of CC^T, i.e.,

$$\min\left\{\sigma : \sigma \in \text{spectrum}(CC^T) \setminus \{0\}\right\} = \min\left\{\sigma : \sigma \in \text{spectrum}(C^T C) \setminus \{0\}\right\} = \sigma_1, \tag{4.62}$$

where σ_1 is the smallest nonzero eigenvalue of $C^T C$. Further, (ii) for $x \in \left(\ker\left(C^T\right)\right)^{\perp}$, we have $\|C^T x\|^2 = \langle x, CC^T x\rangle \geq \min\left\{\sigma : \sigma \in \text{spectrum}(CC^T) \setminus \{0\}\right\} \cdot \|x\|^2$. Hence, using (4.62) in (4.61) we get that $\|C^T \mu_{\mathcal{P}}\| \geq \sqrt{\sigma_1}\|\mu_{\mathcal{P}}\|$, and together with (4.55) and (4.60), we obtain

$$\|\mu_{\mathcal{P}}\| \leq \sigma^{-1/2}\|C^T \mu_{\mathcal{P}}\| \leq 2\sigma^{-1/2}S^{-1/2}\|A\|\|f\| \tag{4.63}$$

which shows (4.57).

Now we consider the particular choice of C given by the left-hand side of (4.53). Similar to the derivation in (4.46), we have that $C^T C = S \cdot I - (1, \dots, 1)(1, \dots, 1)^T)$. Further, the constants constitute the kernel of $C^T C$ and any vector u in its orthogonal complement is mapped to Su. Hence, $\sigma_1 = S$ which shows (4.58).

Finally, we consider the particular choice of C given by the right-hand side of (4.53). As already explained in the proof of Lemma 4.9, the discrete Fourier transform shows that the corresponding eigenvalues are given by $\lambda_k = \rho\left(2 - 2\cos\left(2\pi\frac{k}{S}\right)\right)$, where $k = 0, \dots, S-1$. The smallest nonzero eigenvalue is thus given by $2 - 2\cos(2\pi/S)$. This shows (4.59) which completes the proof of the lemma. □

4.3.3. The quadratic penalty relaxation of the Potts problem and its relation to the Potts problem

In this section, we point out several relations between the Potts problem and its quadratic penalty relaxation. In particular, we show Theorem 4.4 and parts of Theorem 4.6. We start out by showing that the quadratic penalty relaxation of the Potts problem is still NP-hard. This was formulated as Theorem 4.4.

Proof of Theorem 4.4. We consider the quadratic penalty relaxation (4.7) of the multivariate Potts problem in its formulation (4.13) which reads

$$P_{\gamma,\rho}(u_1, \dots, u_S) = \left\|B(u_1, \dots, u_S)^T - g\right\|_2^2 + \gamma \left\| D(u_1, \dots, u_S) \right\|_{0,\omega},$$

where B and g are given by (4.10) and D is given by (4.11). We recast $u : (u_1, \dots, u_S) : \Omega \to \mathbb{R}^S$ as a function $\hat{u} : X \to \mathbb{R}$ on the discrete interval $X \subset \mathbb{Z}$ of size $S\#\Omega$ as follows: for u_s, we consider the discrete lines in the image with direction a_s and consider u on these lines as a vector; next we concatenate these vectors starting with the one corresponding to the leftmost upper line to obtain a vector of length $\#\Omega$; for each s we obtain such a vector and we again concatenate these vectors starting with index $s = 1, 2, \dots, S$ to obtain the resulting vector which we denote by $\hat{u}$. With this serialization at hand, we may arrange B, g and D accordingly and obtain the *univariate* Potts problem

$$\hat{P}_{\gamma,\rho}(\hat{u}) = \left\|\hat{B}\hat{u} - \hat{g}\right\|_2^2 + \gamma \left\|\hat{\omega}\nabla\hat{u}\right\|_0,$$

where $\hat{\omega} : X \to [0,\infty)$ is a weight vector, $\omega \nabla \hat{u}$ denotes pointwise multiplication, and $\hat{B}, \hat{g}$ are the matrix and the vector corresponding to B, g w.r.t. the serialization. The weight vector can be zero which happens at the line breaks, i.e., those indices where two vectors have been concatenated in the above procedure. More precisely, constant data induce a directional partitioning on Ω and the image of the directional partitioning under the above serialization procedure induces a partitioning of the univariate domain D; precisely between the segments, the weight vectors equals zero. Now, for each segment $[d_1, \dots, d_r]$ in D, we transform the basis $\delta_{d_1}, \dots, \delta_{d_r}$ to the basis $\delta_{d_2} - \delta_{d_1}, \dots, \delta_{d_r} - \delta_{d_{r-1}}, \frac{1}{r} \sum_{l=1}^{r} \delta_{d_l}$ obtained by neighboring differences and the average. As a result (which is in detail elaborated in [182]), we obtain a problem of the form

$$\hat{P}_{\gamma,\rho}(\hat{u}) = \left\| \tilde{B}\tilde{u} - \tilde{b} \right\|_2^2 + \gamma \left\| \hat{\omega}\tilde{u} \right\|_0, \tag{4.64}$$

where $\hat{\omega} : D \to [0,\infty)$. Problem (4.64) is a sparsity problem which is known to be NP-hard; see, for instance, [182]. This shows the assertion. □

We give characterizations of the local minimizers of the relaxed Potts problem (4.7) and of the Potts problem (4.4), respectively.

Lemma 4.12. *A local minimizer $u = (u_1, \dots, u_S)$ of the quadratic penalty relaxation* (4.7) *is characterized as follows. Let $\mathcal{I}$ be the directional partitioning induced by the minimizer u and $\mathcal{P} = \mathcal{P}_{\mathcal{I}}$ be the induced partitioning, then u is a minimizer of the problem*

$$\min_{u \in \mathcal{A}^{\mathcal{P}}} F_\rho(u), \qquad \text{where} \qquad F_\rho(u) = \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \|Au_s - f\|_2^2 + \rho \|Cu\|^2. \tag{4.65}$$

Conversely, if u minimizes (4.65) *, then u is a minimizer of the relaxed Potts problem* (4.7).

Proof. Let $u = (u_1, \dots, u_S)$ be a local minimizer of the quadratic penalty relaxation (4.7). Then there is a neighborhood $\mathcal{U}$ of u such that, for any $v \in \mathcal{U}$, $P_{\gamma,\rho}(v) \geq P_{\gamma,\rho}(u)$. Now if $v \in \mathcal{A}^{\mathcal{P}}$ and $\|v - u\|$ is small, then $\sum_{s=1}^{S} \omega_s \|\nabla_{a_s} u_s\|_0 = \sum_{s=1}^{S} \omega_s \|\nabla_{a_s} v_s\|_0$ which implies that

$$F_\rho(u) = P_{\gamma,\rho}(u) - \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{a_s} u_s\|_0 \leq P_{\gamma,\rho}(v) - \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{a_s} v_s\|_0 = F_\rho(v). \tag{4.66}$$

This shows that u is a local minimizer of (4.65). Conversely, we assume that u minimizes (4.65). If the directional partitioning $\mathcal{I}'$ induced by u is coarser than $\mathcal{I}$ consider the coarser directional partitioning $\mathcal{I}'$ instead of $\mathcal{I}$. Let the maximum norm of $h = (h_1 \dots, h_S)$ be smaller than the height of the smallest jump of u, then, for $u + h$,

$$\sum_{s=1}^{S} \omega_s \|\nabla_{a_s}(u_s + h_s)\|_0 \geq \sum_{s=1}^{S} \omega_s \|\nabla_{a_s} u_s\|_0 \,. \tag{4.67}$$

If inequality holds in (4.67), then the continuity of F_ρ implies that $F_\rho(u + h) \geq F_\rho(u) - \varepsilon$ for small enough h and arbitrary ε. Hence,

$$\begin{aligned} P_{\gamma,\rho}(u) &= F_\rho(u) + \gamma \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s} u_s\right\|_0 \leq F_\rho(u+h) - \gamma \min_s \omega_s + \gamma \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s}(u_s+h_s)\right\|_0 + \varepsilon \\ &\leq F_\rho(u+h) + \gamma \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s}(u_s+h_s)\right\|_0 = P_{\gamma,\rho}(u+h), \end{aligned} \tag{4.68}$$

if we choose ε small enough. If equality holds in (4.67) we have that $u+h \in \mathcal{A}^{\mathcal{P}}$ which implies $F_\rho(u) \leq F_\rho(u+h)$ since u is a minimizer of F_ρ on $\mathcal{A}^{\mathcal{P}}$. This in turn implies $P_{\gamma,\rho}(u) \leq P_{\gamma,\rho}(u+h)$ by the assumed equality in (4.67). Together, in any case, $P_{\gamma,\rho}(u) \leq P_{\gamma,\rho}(u+h)$ for any small h. This shows that u is a local minimizer of $P_{\gamma,\rho}$ which completes the proof. □

Lemma 4.13. *We consider a function $u^* : \Omega \to \mathbb{R}$ and its induced partitioning $\mathcal{P}$. Then u^* is a local minimizer of the Potts problem* (4.4), *if and only if $(u^*, \dots, u^*)$ minimizes* (4.51) *w.r.t. $\mathcal{P}$.*

Proof. Since the proof of this statement is very similar to the proof of Lemma 4.12 we keep it rather short and refer to the proof of Lemma 4.12 if more explanation is necessary. Let u be a minimizer of (4.4) which is equivalent to $\bar{u} = (u, \dots, u)$ being a minimizer of (4.6). There is a neighborhood $\mathcal{U}$ of $\bar{u}$ such that, for any $\bar{v} = (v, \dots, v) \in \mathcal{U}$, $P_\gamma(v) \geq P_\gamma(u)$. For $\bar{v} \in \mathcal{B}^{\mathcal{P}}$ with small $\|\bar{v} - \bar{u}\|$, we have $\sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s} u\right\|_0 = \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s} v\right\|_0$. Hence, by the definition of P_γ in (4.6), $\|Au - f\|_2^2 \leq \|Av - f\|_2^2$ which shows that $(u^*, \dots, u^*)$ minimizes (4.51).

Conversely, let $\bar{u} = (u, \dots, u)$ be a minimizer of (4.51) with the partitioning $\mathcal{P}$ induced by u. For $\bar{h} = (h, \dots, h)$ with absolute value smaller than the minimal height of a jump of u, we have the estimate $\sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s}(u+h)\right\|_0 \geq \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s} u\right\|_0$. If inequality holds in this estimate, the continuity of F_ρ implies that $\|A(u+h) - f\|_2^2 \geq \|Au - f\|_2^2 - \varepsilon$ for small enough h and arbitrary ε. Hence, $P_\gamma(\bar{u}) \leq \|A(u+h) - f\|_2^2 - \gamma \min_s \omega_s + \gamma \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s}(u+h)\right\|_0 + \varepsilon \leq P_{\gamma,\rho}(\bar{u} + \bar{h})$ if ε is small. If equality holds above, i.e, $\sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s}(u+h)\right\|_0 = \sum_{s=1}^{S} \omega_s \left\|\nabla_{a_s} u\right\|_0$, then $\bar{u} + \bar{h} \in \mathcal{B}^{\mathcal{P}}$ which implies that $\|Au - f\|_2^2 \leq \|A(u+h) - f\|_2^2$ since $\bar{u}$ is a minimizer of the corresponding functional on $\mathcal{B}^{\mathcal{P}}$. As a consequence, $P_\gamma(\bar{u}) \leq P_\gamma(\bar{u} + \bar{h})$ for any small perturbation h. This shows that u is a local minimizer of P_γ which completes the proof. □

We prove that local minimizers of the quadratic penalty relaxation of the Potts problem are approximate local minimizers of the Potts problem.

Proposition 4.14. *Any local minimizer of the quadratic penalty relaxation* (4.7) *is an approximate local minimizer in the sense of* (4.38) *of the Potts problem* (4.5).

Proof. By Lemma 4.12, a local minimizer $u = (u_1, \dots, u_S)$ of the quadratic penalty relaxation (4.7) is a minimizer of the problem (4.65). Thus, we consider a local minimizer u of (4.7) with induced partitioning $\mathcal{P} = \mathcal{P}_{\mathcal{I}}$. As u minimizes (4.65), we have

$$\tfrac{1}{S} P_{\mathcal{I}} \tilde{A}^T \tilde{A} P_{\mathcal{I}} u - \tfrac{1}{S} P_{\mathcal{I}} \tilde{A}^T \tilde{f} + \rho P_{\mathcal{I}} C^T C P_{\mathcal{I}} u = 0 \tag{4.69}$$

since the gradient projected to $\mathcal{A}^{\mathcal{P}}$ equals zero for any local minimizer of the restricted problem on the subspace $\mathcal{A}^{\mathcal{P}}$. (The notation is as in (4.56) above.) We define μ by $\mu = \rho C P_{\mathcal{I}} u$ and obtain

$$L(\mu) = \|\tfrac{1}{S} P_{\mathcal{I}} \tilde{A}^T \tilde{A} P_{\mathcal{I}} u - \tfrac{1}{S} P_{\mathcal{I}} \tilde{A}^T \tilde{f} + P_{\mathcal{I}} C^T \mu\| = 0 \tag{4.70}$$

by (4.69). It remains to show that $\|Cu\|$ becomes small. To this end, we observe that, by Lemma 4.15, for arbitrary $v = (v_1, \ldots, v_S) \in \mathcal{A}^{\mathcal{P}}$,

$$\|Cv\| = \|CP_{\mathcal{I}} v\| \leq \tfrac{1}{\rho}\|\mu^*\| + \sqrt{\tfrac{F_\rho(v) - \min_{x \in \mathcal{A}^{\mathcal{P}}} F_\rho(x)}{\rho}},$$

where μ^* is an Lagrange multiplier of (4.51). Plugging in the minimizer u for v yields $\|Cu\| < \frac{1}{\rho}\|\mu^*\|$. Thus, by letting $\delta = \frac{1}{\rho}\|\mu^*\|$, we obtain

$$\sum_{s,s'} c_{s,s'} \|u_s^* - u_{s'}^*\|_2^2 = \|Cu\|^2 < \delta, \tag{4.71}$$

and $L(\mu) = 0$ by (4.70) which by (4.38) shows the assertion and completes the proof. □

For the proof of Proposition 4.14 and for subsequent proofs, we need the next lemma. Similar statements are [128, Proposition 13] and [142, Lemma 2.5]. However, since there are differences concerning the precise estimate in these references, and the setup here is slightly different, we provide a brief proof.

Lemma 4.15. *Let $\mathcal{P}$ be a partitioning and $\mathcal{I} = \mathcal{I}_{\mathcal{P}}$ be the corresponding induced partitioning. For arbitrary $v = (v_1, \ldots, v_S) \in \mathcal{A}^{\mathcal{P}}$ it holds that*

$$\|Cv\| = \|CP_{\mathcal{I}} v\| \leq \tfrac{1}{\rho}\|\mu^*\| + \sqrt{\tfrac{F_\rho(v) - \min_{x \in \mathcal{A}^{\mathcal{P}}} F_\rho(x)}{\rho}}, \tag{4.72}$$

where μ^* is an arbitrary Lagrange multiplier of (4.51).

Proof. By [128, Corollary 2], we have for arbitrary $v = (v_1, \ldots, v_S) \in \mathcal{A}^{\mathcal{P}}$ that

$$\sum\nolimits_{s=1}^{S} \tfrac{1}{S} \|Av_s - f\|_2^2 - \min_{(y,\ldots,y) \in \mathcal{B}^{\mathcal{P}}} \|Ay - f\|_2^2 \geq -\|\mu^*\| \, \|Cv\|. \tag{4.73}$$

Then,

$$\begin{aligned} F_\rho(v) - \min_{x \in \mathcal{A}^{\mathcal{P}}} F_\rho(x) &\geq \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \|Av_s - f\|_2^2 + \rho\|Cv\|^2 - \min_{(y,\ldots,y) \in \mathcal{B}^{\mathcal{P}}} F_\rho(y, \ldots, y) \\ &= \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \|Av_s - f\|_2^2 + \rho\|Cv\|^2 - \min_{(y,\ldots,y) \in \mathcal{B}^{\mathcal{P}}} \|Ay - f\|_2^2 \\ &\geq \rho\|Cv\|^2 - \|\mu^*\| \, \|Cv\|. \end{aligned} \tag{4.74}$$

For the first inequality we used the definition of F_ρ and restricted the set with respect to which the minimum is formed which results in a potentially larger functional value. For the equality we notice that, for $(y, \ldots, y) \in \mathcal{B}^{\mathcal{P}}$, we have $C(y, \ldots, y) = 0$, and for the last inequality we employed (4.73). Now, by writing $z^2 - \frac{\|\mu^*\|}{\|\rho\|} z = z^2 - \frac{\|\mu^*\|}{\rho} z + \left(\frac{\|\mu^*\|}{2\rho}\right)^2 - \left(\frac{\|\mu^*\|}{2\rho}\right)^2 = (z - \frac{\|\mu^*\|}{2\rho})^2 - \left(\frac{\|\mu^*\|}{2\rho}\right)^2$ and plugging this into (4.74) with $z := \|Cv\|$ we obtain

$$\tfrac{F_\rho(v) - \min_{x \in \mathcal{A}^{\mathcal{P}}} F_\rho(x)}{\rho} \geq \left(\|Cv\| - \tfrac{\|\mu^*\|}{2\rho}\right)^2 - \left(\tfrac{\|\mu^*\|}{2\rho}\right)^2, \tag{4.75}$$

and consequently

$$\left| \|Cv\| - \tfrac{\|\mu^*\|}{2\rho} \right| \leq \sqrt{\tfrac{F_\rho(v) - \min_{x \in \mathcal{A}^\mathcal{P}} F_\rho(x)}{\rho} + \left(\tfrac{\|\mu^*\|}{2\rho} \right)^2} \leq \sqrt{\tfrac{F_\rho(v) - \min_{x \in \mathcal{A}^\mathcal{P}} F_\rho(x)}{\rho}} + \tfrac{\|\mu^*\|}{2\rho}, \tag{4.76}$$

where the last inequality is a consequence of the fact that the unit ball w.r.t. the ℓ_1-norm is contained in the the unit ball w.r.t. the ℓ_2-norm. As a consequence, $\|Cv\| \leq \sqrt{\frac{F_\rho(v) - \min_{x \in \mathcal{A}^\mathcal{P}} F_\rho(x)}{\rho}} + \frac{\|\mu^*\|}{2\rho} + \frac{\|\mu^*\|}{2\rho}$ which completes the proof. ☐

Next, we see that for any local minimizer of the quadratic penalty relaxation (4.7), a nearby feasible point can be found by using the projection procedure (Procedure 4.1) of Section 4.2.2. What is more, if the imaging operator A is lower bounded, we find a nearby minimizer.

Proposition 4.16. *Applying Procedure 4.1 to a local minimizer* $u' = (u'_1, \ldots, u'_S)$ *of the quadratic penalty relaxation* (4.7) *produces a feasible image* $\hat{u}$ *(together with a valid partitioning) for the Potts problem* (4.5) *which is close to* u'*,i.e.,*

$$\|u'_s - \hat{u}\| \leq C_1 \varepsilon \qquad \text{for all} \quad s \in \{1, \ldots, S\}, \tag{4.77}$$

where $\varepsilon = \max_{s,s'} \|u'_s - u'_{s'}\|$ *quantifies the deviations between the* u_s*. We denote* $C_1 = \#\Omega/4$*, for the number of elements* $\#\Omega$ *in* Ω*. Further, if the imaging operator* A *is lower bounded, i.e., there is a constant* $c > 0$ *such that* $\|Au\| \geq c\|u\|$*, a local minimizer* u^* *of the Potts problem* (4.5) *is nearby in the sense that*

$$\|u^* - \hat{u}\| \leq \frac{\sqrt{\eta}}{c}, \quad \text{where} \tag{4.78}$$

$$\eta := \left(\|A\|^2 \varepsilon C_1^2 + 2\|A\| C_1 \|f\|_2 \right) \varepsilon. \tag{4.79}$$

Proof. We let the directional partitioning induced by u' be $\mathcal{I}$ and denote the corresponding induced partitioning by $\mathcal{P} = \mathcal{P}_\mathcal{I}$. We note that Procedure 4.1 applied to u' precisely produces

$$(\hat{u}, \ldots, \hat{u}) = Q_\mathcal{P} u', \tag{4.80}$$

for the projection $Q_\mathcal{P}$ defined in (4.50). First, we see that the average $(\bar{u})_{ij} = \frac{1}{S} \sum_{s=1}^S (u'_s)_{ij}$ fulfills $|(\bar{u})_{ij} - (u'_s)_{ij}| < \varepsilon$. Further, the function value of $\hat{u}$ which is piecewise constant w.r.t. $\mathcal{P}$ is obtained by $\hat{u}|_{\mathcal{P}_i} = \sum_{x \in \mathcal{P}_i} \bar{u}(x) / \#\mathcal{P}_i$. Hence, we have the estimate

$$\|u'_s - \hat{u}\|_2^2 \leq \varepsilon L, \tag{4.81}$$

where L is the maximal length of a path connecting any two pixels as given by Definition 4.3. As a worst case estimate, we get $L \leq C_1$ where we define C_1 as one fourth of the number of elements in Ω, i.e., $C_1 = \frac{\#\Omega}{4}$. This shows (4.77).

For F_ρ given by (4.65), we have

$$\begin{aligned} F_\rho(u') &\leq F_\rho(\hat{u},\dots,\hat{u}) = \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \|A\hat{u}-f\|_2^2 \\ &\leq \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \left(\|A\hat{u}-Au'_s\|_2 + \|Au'_s - f\|_2\right)^2 \\ &\leq \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \left(\|A\|\varepsilon C_1 + \|Au'_s - f\|_2\right)^2 \\ &\leq \|A\|^2\varepsilon^2 C_1^2 + 2\|A\|\varepsilon C_1 \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \|Au'_s - f\|_2 + \sum\nolimits_{s=1}^{S} \tfrac{1}{S} \|Au'_s - f\|_2^2 \\ &\leq \eta + F_\rho(u'), \end{aligned} \tag{4.82}$$

with

$$\eta = \left(\|A\|^2\varepsilon C_1^2 + 2\|A\|C_1\|f\|_2\right)\varepsilon, \tag{4.83}$$

as given in (4.79). The first inequality holds as a local minimizer of the quadratic penalty relaxation (4.7), u' is the global minimizer of F_ρ on $\mathcal{A}^{\mathcal{P}}$ by Lemma 4.12 and since $(\hat{u},\dots,\hat{u}) \in \mathcal{A}^{\mathcal{P}}$ by construction. The next inequalities apply the triangle inequality and estimates on matrix norms. The last inequality is a consequence of the fact that $\sum_{s=1}^{S} \frac{1}{S} \|Au'_s - f\|_2 \leq \|f\|_2$. Otherwise, if $\|Au'_s - f\|_2 > \|f\|_2$, choosing $u'_s = 0$ would yield a lower functional value which would contradict the minimality of u'.

Now we consider the partitioning $\mathcal{P}'$ induced by $\hat{u}$, and the corresponding minimizer u^*, i.e.,

$$(u^*,\dots,u^*) = \operatorname*{argmin}_{u\in\mathcal{B}^{\mathcal{P}'}} F_\rho(u), \tag{4.84}$$

where, for $(u,\dots,u) \in \mathcal{B}^{\mathcal{P}'}$, we have $F_\rho(u,\dots,u) = \|Au - f\|_2^2$. By Lemma 4.13, u^* is a local minimizer of the Potts problem (4.4). On the other hand, by orthogonality we have

$$Au^* = P_{A(\mathcal{B}^{\mathcal{P}'})}f, \qquad \text{and} \qquad \|f - P_{A(\mathcal{B}^{\mathcal{P}'})}f\|^2 = \min_{u\in\mathcal{B}^{\mathcal{P}'}} F_\rho(u), \tag{4.85}$$

where $P_{A(\mathcal{B}^{\mathcal{P}'})}$ denotes the orthogonal projection onto the image of $\mathcal{B}^{\mathcal{P}'}$ under the linear mapping A. Thus,

$$\begin{aligned} \|A\hat{u} - Au^*\|^2 &= \|A\hat{u} - P_{A(\mathcal{B}^{\mathcal{P}'})}f\|^2 \\ &= \|A\hat{u} - f\|^2 - \|f - P_{A(\mathcal{B}^{\mathcal{P}'})}f\|^2 = \|A\hat{u} - f\|^2 - \|Au^* - f\|^2. \end{aligned} \tag{4.86}$$

Inserting u^* in the estimate (4.82), we obtain

$$F_\rho(u') \leq F_\rho(u^*,\dots,u^*) \leq F_\rho(\hat{u},\dots,\hat{u}) \leq \eta + F_\rho(u') \leq \eta + F_\rho(u^*,\dots,u^*). \tag{4.87}$$

This allows us to further estimate

$$\|A\hat{u} - Au^*\|^2 = \|A\hat{u} - f\|^2 - \|Au^* - f\|^2 \leq \|Au^* - f\|^2 + \eta - \|Au^* - f\|^2 = \eta. \tag{4.88}$$

If the operator A is lower bounded, then

$$\|\hat{u} - u^*\|^2 < \frac{1}{c^2}\|A\hat{u} - Au^*\|^2 \leq \frac{\eta}{c^2} \tag{4.89}$$

which completes the proof. □

4.3.4. Majorization-minimization for multivariate Potts problems

In this section, we lay the foundation of the convergence analysis of Algorithm 4.1 and Algorithm 4.2. To this end, we first recall basics on surrogate functionals. We consider functionals $F(u)$ of the form $F(u) = \|Xu - z\|^2 + \gamma J(u)$, where X is a given (measurement) matrix with operator norm $\|X\| < 1$ (the operator norm is formed w.r.t. the ℓ_2-norm), z is a given vector (of data), J is an arbitrary (not necessarily convex) lower semicontinuous functional, and $\gamma > 0$ is a parameter. The *surrogate functional* $F^{\mathrm{surr}}(u, v)$ of $F(u)$ is given by

$$F^{\mathrm{surr}}(u, v) = F(u) + \|u - v\|^2 - \|Xu - Xv\|^2. \tag{4.90}$$

Lemma 4.17. *Consider the functionals* $F(u) = \|Xu - z\|^2 + \gamma J(u)$ *as above with* $\|X\| < 1$. *(For our purposes,* J *is the regularizer* $\|D(u)\|_{0,\omega}$ *given by* (4.12).*) Then, for the associated surrogate functional* F^{surr} *given by* (4.90) *(with* J *as regularizer), it holds that*

(i) the inequality

$$F^{\mathrm{surr}}(u, v) \geq F(u)$$

holds for all v; *and* $F^{\mathrm{surr}}(u, v) = F(u)$ *if and only if* $u = v$;

(ii) the functional values $F(u^k)$ *of the sequence* u^k *defined by the* surrogate iteration $u^{k+1} = \operatorname{argmin}_u F^{\mathrm{surr}}(u, u^k)$ *are non-increasing, i.e.,*

$$F(u^{k+1}) \leq F(u^k); \tag{4.91}$$

(iii) the distances between consecutive members of the sequence u^k *converge to zero, i.e.,*

$$\lim_{k\to\infty} \|u^{k+1} - u^k\| = 0. \tag{4.92}$$

We note that for minimizing F the condition $\|X\| < 1$ can always be achieved by rescaling, i.e., by dividing the functional F by a number larger than $\|X\|^2$. Proofs of the general statements *(i)-(iii)* above can be found in [59, 22, 75].

We now employ properties of the quadratic penalty relaxation $P_{\gamma,\rho}(u_1, \ldots, u_S)$ of the Potts functional which was given by (4.7). The strategy is similar to the approach to the univariate case which was considered in [201]. We first show that the minimizers of $P_{\gamma,\rho}(u_1, \ldots, u_S)$ with $B = \mathrm{id}$ in (4.13) (which are precisely the solutions of (4.21)) have a minimal directional jump height that depends on the scale parameter γ, the directional weights ω_s and the constant L_ρ only, but not on the input data. For the multivariate discrete function $u = (u_1, \ldots, u_S)$ (and the directional system a_s, $s = 1, \ldots, S$), a *directional jump* is a jump of the s-th component u_s in direction a_s. In particular, jumps of the components u_s in directions $a_{s'}$, $s' \neq s$ are *not* considered.

Lemma 4.18. *We consider the functional* $P_{\gamma,\rho}(u_1, \ldots, u_S)$ *of* (4.13) *for* $B = \mathrm{id}$ *and data* $h = (h_1, \ldots, h_S)$. *Equivalently, we consider the problem* (4.21) *for arbitrary data* $h = (h_1, \ldots, h_S)$. *Then there is a constant* $c > 0$ *which does not depend on the minimizer* $u^* = (u_1^*, \ldots, u_S^*)$ *of*

(4.21) *and the data h such that the minimal directional jump height $j_{\min}(u^*)$ (w.r.t. the directional system a_s, $s = 1, \dots, S$,) of a minimizer u^* fulfills*

$$j_{\min}(u^*) \geq c, \tag{4.93}$$

where the constant c depends on γ, the directional weights ω_s and the constant L_ρ.

Proof. By writing $u = (u_1, \dots, u_S)$ we restate (4.21) as minimizing

$$P^{\mathrm{id}}_{\gamma/L_\rho^2}(u_1, \dots, u_S) = \|u - h\|_2^2 + \frac{\gamma}{L_\rho^2} \left\| D(u_1, \dots, u_S) \right\|_{0,\omega}, \tag{4.94}$$

where we use the notation $\|D(u_1, \dots, u_S)\|_{0,\omega} = \sum_{s=1}^S \omega_s \left\|\nabla_{a_s} u_s \right\|_0$ introduced in (4.12). We let

$$c = \sqrt{\frac{\gamma \ \min_{s \in \{1,\dots,S\}} \omega_s}{L_\rho^2 W}}, \tag{4.95}$$

where W denotes the maximal length of the signal u per dimension (to fix ideas, for an $l \times b$ image, we have $W = \max(l, b)$). Towards a contradiction, we assume that $h_{\min}(u^*) < c$, which means that the minimizer u^* has a directional jump of height smaller than c. We construct an element u' with a smaller $P^{\mathrm{id}}_{\gamma/L_\rho^2}$ value than u^* (which yields a contradiction as u^* is a minimizer of $P^{\mathrm{id}}_{\gamma/L_\rho^2}$). To this end, we let a_s be a direction such that the component u_s^* of u^* has a jump with height smaller than c. We denote the two (discrete) directional intervals in direction a_s which border the directional jump by I_1, I_2. We let m_1, m_2 and m be the mean of h_s on I_1, I_2 and $I_1 \cup I_2$, respectively. We define

$$u'_{s'} = u^*_{s'} \quad \text{if} \quad s' \neq s, \qquad \text{and} \qquad u'_s(x) = \begin{cases} m & \text{if } x \in I_1 \cup I_2, \\ u^*_s(x) & \text{otherwise.} \end{cases} \tag{4.96}$$

By construction, $\|\nabla_{a_s} u'_s\|_0 = \|\nabla_{a_s} u^*_s\|_0 - 1$ and thus

$$\|D(u'_1, \dots, u'_S)\|_{0,\omega} = \|D(u^*_1, \dots, u^*_S)\|_{0,\omega} - \omega_s \leq \|D(u^*_1, \dots, u^*_S)\|_{0,\omega} - \min_{s \in \{1,\dots,S\}} \omega_s. \tag{4.97}$$

Since u^* is a minimizer of $P^{\mathrm{id}}_{\gamma/L_\rho^2}$, its s-th component u^*_s equals m_1 on I_1 and m_2 on I_2. Further, as $u'_{s'} = u^*_{s'}$ if $s' \neq s$ and u^*_s and u'_s only differ on $I_1 \cup I_2$, we have that

$$\begin{aligned} \|u' - h\|^2 = \sum_{s'=1}^S \|u'_{s'} - h_{s'}\|^2 &= \sum_{s'=1, s' \neq s}^S \|u^*_{s'} - h_{s'}\|^2 + \|u^*_s - h_s\|^2 + l_1 |m_1 - m|^2 + l_2 |m_2 - m|^2 \\ &< \|u^* - h\|^2 + W c^2, \end{aligned} \tag{4.98}$$

where l_1, l_2 denote the length of I_1 and I_2, respectively. Employing (4.97) together with (4.98) we get

$$\begin{aligned} P^{\mathrm{id}}_{\gamma/L_\rho^2}(u'_1, \dots, u'_S) &= \|u' - h\|_2^2 + \frac{\gamma}{L_\rho^2} \left\| D(u'_1, \dots, u'_S) \right\|_{0,\omega} \\ &< \|u^* - h\|^2 + W c^2 + \frac{\gamma}{L_\rho^2} \|D(u^*_1, \dots, u^*_S)\|_{0,\omega} - \frac{\gamma}{L_\rho^2} \min_{s \in \{1,\dots,S\}} \omega_s \\ &\leq \|u^* - h\|^2 + \frac{\gamma}{L_\rho^2} \|D(u^*_1, \dots, u^*_S)\|_{0,\omega} = P^{\mathrm{id}}_{\gamma/L_\rho^2}(u^*_1, \dots, u^*_S). \end{aligned}$$

The validity of the last inequality follows by (4.95). Together, u' has a smaller functional value than u^* which is a contradiction to u^* being a minimizer which shows the assertion. □

We have gathered all ingredients to prove the following proposition which states the convergence of Algorithm 4.1 to a local minimizer of the quadratic penalty relaxation.

Proposition 4.19. *The iteration* (4.24) *of Algorithm 4.1 converges to a local minimizer of the quadratic penalty relaxation* $P_{\gamma,\rho}$ *of the Potts functional given by* (4.7). *The convergence rate is linear.*

Proof. We divide the proof into three parts. First, we show that the directional partitionings induced by the iterates $u^{(n)}$ become fixed after sufficiently many iterations. In the second part, we derive the convergence of Algorithm 4.1 and in the third part, we show that the limit point is a local minimizer of $P_{\gamma,\rho}$.

(i) We first show that the directional partitioning $\mathcal{I}^n$ induced by the iterates $u^{(n)}$ gets fixed for large n. For every $n \in \mathbb{N}$, the iterate $u^{(n)}$ of Algorithm 4.1 is a minimizer of the functional $P_{\gamma,\rho}$ of (4.13) for the choice $B = \mathrm{id}$ as it appears in (4.21) and data $h = (h_1, \ldots, h_S)$ given by (4.22). By Lemma 4.18 there is a constant $c > 0$ which is independent of the particular $u^{(n)} = (u_1^{(n)}, \ldots, u_S^{(n)})$ of (4.21) and the data h such that the minimal directional jump height $j_{\min}(u^{(n)})$ fulfills

$$j_{\min}(u^{(n)}) \geq c \quad \text{for all} \quad n \in \mathbb{N}. \tag{4.99}$$

We note that the parameter γ, the directional weights ω_s and the constant L_ρ –which the constant c depends on by Lemma 4.18– are fixed in the iterations of Algorithm 4.1.

If two iterates $u^{(n)}, u^{(n+1)}$ induce different directional partitionings $\mathcal{I}^n$, $\mathcal{I}^{n+1}$, their ℓ_∞-distance satisfies $\|u^{(n)} - u^{(n+1)}\|_\infty > c/2$ since both $u^{(n)}, u^{(n+1)}$ have minimal jump height of at least c and the induced directional partitionings are different. This implies $\|u^{(n)} - u^{(n+1)}\|_2 > c/2$ for the ℓ_2-distance as well. This can only happen for small n, since by Lemma 4.17 we have $\|u^{(n)} - u^{(n+1)}\|_2 \to 0$ as n increases. Hence, there is an N such that, for all $n \geq N$, the directional partitionings $\mathcal{I}^n$ are identical.

(ii) We use the previous observation to show the convergence of Algorithm 4.1. We consider the iterates $u^{(n)}$ with $n \geq N$. These have the same induced directional partitionings which we denote by $\mathcal{I}'$ and all jumps have minimal jump height c. Hence, for $n \geq N$, the iteration of (4.21) can be written as

$$u^{(n+1)} = P_{\mathcal{I}'}(h^{(n)}), \tag{4.100}$$

where $P_{\mathcal{I}'}$ is the orthogonal projection onto the space $\mathcal{A}^{\mathcal{P}}$ which consists of the functions which are piecewise constant w.r.t. the directional partitioning $\mathcal{I}'$, and where $h^{(n)}$ depends on $u^{(n)}$ via

$$h_s^{(n)} = u_s^{(n)} + \tfrac{1}{SL_\rho^2} A^* f - \tfrac{1}{SL_\rho^2} A^* A u_s^{(n)} - \sum_{s':s' \neq s} \tfrac{\rho_{s,s'}}{L_\rho^2} (u_s^{(n)} - u_{s'}^{(n)}), \qquad \text{for all } s \in \{1, \ldots, S\}, \tag{4.101}$$

as given by (4.22). As introduced before, we use the symbols $\tilde{A}$ to denote the block diagonal matrix with constant entry A on each diagonal component, and $\tilde{f}$ for the block vector of corresponding dimensions with entry f in each component. With this notation we may write

(4.100) as

$$u^{(n+1)} = P_{\mathcal{I}'}\Big((I - \tfrac{1}{SL_\rho^2}(\tilde{A})^T\tilde{A} - \tfrac{1}{SL_\rho^2}\rho\, C^TC)u^{(n)} + \tfrac{1}{SL_\rho^2}\tilde{A}^T\tilde{f}\Big). \tag{4.102}$$

Since $u^{(n)}$ is piecewise constant w.r.t. the directional partitioning $\mathcal{I}'$, we have $u^{(n)} = P_{\mathcal{I}'}u^{(n)}$. By using this fact and the fact that $P_{\mathcal{I}'}$ is an orthogonal projection, we obtain

$$u^{(n+1)} = \left(I - \left(\left(\tfrac{\tilde{A}P_{\mathcal{I}'}}{\sqrt{S}L_\rho}\right)^T\left(\tfrac{\tilde{A}P_{\mathcal{I}'}}{\sqrt{S}L_\rho}\right) + \left(\tfrac{\sqrt{\rho}CP_{\mathcal{I}'}}{\sqrt{S}L_\rho}\right)^T\left(\tfrac{\sqrt{\rho}CP_{\mathcal{I}'}}{\sqrt{S}L_\rho}\right)\right)\right)u^{(n)} + \left(\tfrac{\tilde{A}P_{\mathcal{I}'}}{\sqrt{S}L_\rho}\right)^T\tfrac{\tilde{f}}{\sqrt{S}L_\rho}. \tag{4.103}$$

Since $C\tilde{A}^T\tilde{f} = 0$, the iteration (4.103) can be interpreted as the Landweber iteration for the block matrix consisting of the upper block $(\tilde{A}P_{\mathcal{I}'})/(\sqrt{S}L_\rho)$ and the lower block $(\sqrt{\rho}CP_{\mathcal{I}'})/(\sqrt{S}L_\rho)$ and data $\tilde{f}/(\sqrt{S}L_\rho)$ extended by zeros. The Landweber iteration converges at a linear rate; see, e.g., [72]. Thus, the iteration (4.100) convergences and further Algorithm 4.1 converges at a linear rate to some limit u^*.

(iii) We show that u^* is a local minimizer. Since u^* is the limit of the iterates $u^{(n)}$, the jumps of u^* also have minimal height c, the number of jumps are equal to those of the $u^{(n)}$ for all $n \geq N$, and the induced directional partitioning $\mathcal{I}^*$ equals the partitioning $\mathcal{I}'$ of the $u^{(n)}$ for $n \geq N$. Since u^* equals the limit of the above Landweber iteration, u^* minimizes F_ρ given by (4.65) on $\mathcal{A}^{\mathcal{P}_{\mathcal{I}'}}$. Then by Lemma 4.12, u^* is a local minimizer of the relaxed Potts functional $P_{\gamma,\rho}$ which completes the proof. □

After having shown the convergence of Algorithm 4.1 to a local minimizer, we can show Theorem 4.6.

Proof of Theorem 4.6. Assertion *(i)* was stated and shown as Proposition 4.14 in Section 4.3.3. By Proposition 4.19, Algorithm 4.1 produces a local minimizer. Then, assertion *(ii)* is a consequence of Proposition 4.16. □

4.3.5. Estimating the distance between the objectives

The following lemma is a preparation for the proof of item *(iii)* of Theorem 4.5.

Lemma 4.20. *We consider Algorithm 4.1 for the quadratic penalty relaxation* (4.7) *of the multivariate Potts problem. For any output $u = (u_1, \ldots, u_S)$ of Algorithm 4.1, the deviations between the components are bounded by*

$$\left(\sum\nolimits_{s,s'} c_{s,s'}\|u_s - u_{s'}\|_2^2\right)^{\frac{1}{2}} \leq 2\sigma_1^{-1/2}S^{-1/2}\|A\|\|f\|/\rho, \tag{4.104}$$

where σ_1 denotes the smallest non-zero eigenvalue of C^TC for C given by (4.52).

Proof. Since $u = (u_1, \ldots, u_S)$ is the output of Algorithm 4.1, it is a local minimizer of the relaxed Potts problem (4.7). In particular, there is a directional partitioning $\mathcal{I}$ on which u is piecewise constant. We denote the induced partitioning by $\mathcal{P} = \mathcal{P}_{\mathcal{I}}$. By Lemma 4.15 we have

$$\left(\sum\nolimits_{s,s'} c_{s,s'}\|u_s - u_{s'}\|_2^2\right)^{\frac{1}{2}} = \|Cu\| = \|CP_{\mathcal{I}}u\| \leq \tfrac{1}{\rho}\|\mu^*\| + \sqrt{\tfrac{F_\rho(u) - \min_{x\in\mathcal{A}^{\mathcal{P}}} F_\rho(x)}{\rho}}, \tag{4.105}$$

where μ^* is a Lagrange multiplier of (4.51). We have that $\|\mu^*\| \leq 2\sigma_1^{-1/2} S^{-1/2} \|A\| \|f\|$ for any partitioning of the discrete domain Ω by Lemma 4.11 and in particular for the partitioning $\mathcal{P} = \mathcal{P}_\mathcal{I}$. This shows that

$$\|Cu\| \leq 2\sigma_1^{-1/2} S^{-1/2} \|A\| \|f\| / \rho + \sqrt{\frac{F_\rho(u) - \min_{x \in \mathcal{A}^\mathcal{P}} F_\rho(x)}{\rho}}.$$

Since u is a local minimizer of the relaxed Potts problem (4.7), it is a minimizer of F_ρ on $\mathcal{A}^\mathcal{P}$ by Lemma 4.12 and the second summand on the right hand side equals zero. This shows (4.104) which completes the proof. □

We have now gathered all ingredients to show Theorem 4.5.

Proof of Theorem 4.5. *(i)* This is shown by Proposition 4.19. *(ii)* We first show that any global minimizer of the relaxed Potts functional $P_{\gamma,\rho}$ given by (4.7) appears as a stationary point of Algorithm 4.1. To this end, we initialize Algorithm 4.1 with a global minimizer $\bar{u}^* = (u_1^*, \ldots, u_S^*)$. Then, we have for all $\bar{v} = (v_1, \ldots, v_S)$ with $\bar{v} \neq \bar{u}^*$,

$$\begin{aligned} P_{\gamma,\rho}^{\mathrm{surr}}\left(v_1, \ldots, v_S, u_1^*, \ldots, u_S^*\right) &= P_{\gamma,\rho}(\bar{v}) - \|B\bar{v} - B\bar{u}^*\|^2 + \|\bar{v} - \bar{u}^*\|^2 \\ &> P_{\gamma,\rho}(\bar{v}) \geq P_{\gamma,\rho}(\bar{u}^*) = P_{\gamma,\rho}^{\mathrm{surr}}(\bar{u}^*, \bar{u}^*), \end{aligned} \tag{4.106}$$

where B is given by (4.10). The estimate (4.106) states that $\bar{u}^*$ is the minimizer of the surrogate functional w.r.t. the first component, i.e., it is the minimizer of the mapping $\bar{v} \mapsto P_{\gamma,\rho}^{\mathrm{surr}}(\bar{v}, \bar{u}^*)$. Thus, the iterate $\bar{u}^{(1)} = (u_1^{(1)}, \ldots, u_S^{(1)})$ of Algorithm 4.1 equals $\bar{u}^*$ when the iteration is started with $\bar{u}^*$. Thus, a global minimizer $\bar{u}^*$ is a stationary point of Algorithm 4.1. It remains to show that each stationary point of Algorithm 4.1 is a local minimizer of the relaxed Potts functional $P_{\gamma,\rho}$. This was essentially done in the proof of Proposition 4.19: start the iteration (4.21) with a stationary point u'; its limit equals u' and thus is a local minimizer by Proposition 4.19.

(iii) We estimate by Lemma 4.20

$$\left(\sum\nolimits_{s,s'} c_{s,s'} \|u_s - u_{s'}\|_2^2\right)^{\frac{1}{2}} \leq 2\sigma_1^{-1/2} S^{-1/2} \|A\| \|f\| / \rho < \varepsilon. \tag{4.107}$$

The second inequality follows by the choice of ρ in (4.37) as $\rho > 2\varepsilon^{-1}\, \sigma_1^{-1/2} S^{-1/2} \|A\| \|f\|$. This shows *(iii)* which completes the proof. □

4.3.6. Convergence analysis of Algorithm 4.2

We begin with showing that Algorithm 4.2 is well-defined, that is, the inner iteration (4.30) terminates which was formulated as Theorem 4.7.

Proof of Theorem 4.7. The assertion we have to show is that for any $k \in \mathbb{N}$ there is $n \in \mathbb{N}$ such that

$$\left\|u_s^{(k,n)} - u_{s'}^{(k,n)}\right\| \leq \frac{t}{\rho_k \sqrt{c_{s,s'}}}, \qquad \text{and} \qquad \left\|u_s^{(k,n)} - u_s^{(k,n-1)}\right\| \leq \frac{\delta_k}{L_\rho}. \tag{4.108}$$

Concerning the right-hand side of (4.108), we note that by Proposition 4.19, the iteration (4.24) converges to a local minimizer of the quadratic penalty relaxation $P_{\gamma,\rho}(u_1, \ldots, u_S)$ of the Potts functional. The inner loop of Algorithm 4.2 corresponds to the iteration (4.24) (for the current penalty parameter ρ which increases with k). As a consequence, the distance between consecutive iterates $u_s^{(k,n)}, u_s^{(k,n-1)}$ of the inner loop converges to zero for increasing n which proves the right-hand side of (4.108) for sufficiently large n and all $k \in \mathbb{N}$.

Concerning the left-hand inequality in (4.108), we notice that the inner loop of Algorithm 4.2 would converge to a minimizer $\bar{u}^{(k),*} = (u_1^{(k),*}, \ldots, u_S^{(k),*})$ if it was not terminated by (4.108) for all $k \in \mathbb{N}$. As $\bar{u}^{(k),*}$ is a local minimizer of the relaxed Potts problem (4.7) for the parameter ρ_k, it is a minimizer of F_{ρ_k} on $\mathcal{A}^{\mathcal{P}}$ (where $\mathcal{P}$ denotes the partitioning induced by $\bar{u}^{(k),*}$) by Lemma 4.12. Thus, for any $k \in \mathbb{N}$ and any $\xi > 0$ there is $\bar{u}^{(k,n)} = (u_1^{(k,n)}, \ldots, u_S^{(k,n)})$ such that $F_{\rho_k}(\bar{u}^{(k,n)}) - F_{\rho_k}(\bar{u}^{(k),*}) < \xi$. We let $\tau = (t - 2\sigma_1^{-1/2} S^{-1/2} \|A\| \, \|f\|)/\rho_k$, and choose $\xi = \rho_k \tau^2$. Then by using Lemma 4.15 we obtain

$$\begin{aligned}\sqrt{c_{s,s'}} \left\| u_s^{(k,n)} - u_{s'}^{(k,n)} \right\|_2 = \|C\bar{u}^{(k,n)}\| &\leq \tfrac{1}{\rho_k}\|\mu^*\| + \sqrt{\tfrac{F_{\rho_k}(\bar{u}^{(k,n)}) - F_{\rho_k}(\bar{u}^{(k),*})}{\rho_k}} \\ &\leq \tfrac{1}{\rho_k}\|\mu^*\| + \sqrt{\tfrac{\xi}{\rho_k}} \leq \tfrac{1}{\rho_k}\|\mu^*\| + \tau \leq \tfrac{t}{\rho_k}, \end{aligned} \tag{4.109}$$

where μ^* is a Lagrange multiplier of (4.51). Here, the last inequality holds since by Lemma 4.11 we have $\|\mu^*\| \leq 2\sigma_1^{-1/2} S^{-1/2} \|A\| \|f\|$ which implies $\tau \leq (t - \mu^*)/\rho_k$. The estimate (4.109) shows the left-hand inequality in (4.108) which completes the proof. □

We have gathered all ingredients to prove Theorem 4.8 which stated the convergence properties of Algorithm 4.2.

Proof of Theorem 4.8. We first show that any accumulation point of the sequence $u^{(k)}$ produced by Algorithm 4.2 is a local minimizer of the Potts problem (4.5). Let u^* be such an accumulation point and let $\mathcal{I}^*$ be its induced directional partitioning. We can extract a subsequence $u^{(k_l)}$ of the sequence $u^{(k)}$ such that $u^{(k_l)}$ converges to u^* as $l \to \infty$ and such that the directional partitionings $\mathcal{I}^{k_l}$ induced by the $u^{(k_l)}$ all equal the directional partitioning $\mathcal{I}^*$, i.e., $\mathcal{I}^{k_l} = \mathcal{I}^*$ for all $l \in \mathbb{N}$. We let

$$\mu^{k_l} = -2\rho_{k_l} \, C u^{k_l} \tag{4.110}$$

with C given by (4.52) and obtain the following estimation

$$\begin{aligned}\|\tfrac{2}{S}\tilde{A}^T\tilde{A}u^{k_l} - \tfrac{2}{S}\tilde{A}^T\tilde{f} - C^T\mu^{k_l}\| &= \|\tfrac{2}{S}\tilde{A}^T\tilde{A}u^{k_l} - \tfrac{2}{S}\tilde{A}^T\tilde{f} + 2\rho_{k_l}C^T \, Cu^{k_l}\| \\ &= \|\nabla F_{\rho_{k_l}}(u^{k_l})\| \leq \tfrac{\delta_{k_l}}{L_{\rho_{k_l}}} \leq \delta_{k_l}. \end{aligned} \tag{4.111}$$

We recall that $\tilde{A}$ was the block diagonal with the matrix A as the diagonal components and that $F_{\rho_{k_l}}$ was given by (4.65). The second before last inequality follows from the right-hand side of (4.108). We further estimate

$$\|\mu^{k_l}\| = \rho_{k_l}\|Cu^{k_l}\| < \rho_{k_l}\tfrac{St}{\rho_{k_l}} = St$$

which follows from the left-hand side of (4.108). Hence, the sequence μ^{k_l} is bounded and by the Bolzano-Weierstraß-theorem it has a cluster point μ^*. By restricting to a further subsequence (we omit the new indexation for readability and continue using l for the index) we get that

$$\mu^{k_l} \to \mu^* \quad \text{for} \quad l \to \infty. \tag{4.112}$$

On this subsequence, we have that $u^{(k_l)} \to u^*$ and that $\mu^{k_l} \to \mu^*$. Taking limits on both sides of (4.111) gives

$$\tfrac{2}{S}\tilde{A}^T\tilde{A}u^* - \tfrac{2}{S}\tilde{A}^T\tilde{f} - C^T\mu^* = 0 \tag{4.113}$$

since $\delta_{k_l} \to 0$ as $l \to \infty$. Furthermore,

$$\|Cu^*\| \leq \lim_{l\to\infty} \tfrac{\|\mu^{k_l}\|}{\rho_{k_l}} \leq \|\mu^*\| \lim_{l\to\infty} \tfrac{1}{\rho_{k_l}} = 0 \tag{4.114}$$

which implies that the components of u^* are equal, i.e., $u_s^* = u_{s'}^*$ for all s, s'. Therefore, u^* is a feasible point for the Potts problem (4.5). Equivalently, letting $\mathcal{P}^*$ be the partitioning induced by u^*, we have that $u^* \in \mathcal{B}^{\mathcal{P}^*}$. Then (4.113) implies that u^* minimizes (4.51) which by Lemma 4.13 means that u^* is a local minimizer of (4.5). In other words, any component of u^* (which are all equal) minimizes the Potts problem (4.4). This shows the first assertion of Theorem 4.8.

We continue with the second assertion of Theorem 4.8, i.e., if A is lower bounded, then the sequence $u^{(k)}$ produced by Algorithm 4.2 has a cluster point. Then, in view of the above considerations, each cluster point is a local minimizer which shows the assertion. To this end, we show that the sequence $u^{(k)}$ produced by Algorithm 4.2 is bounded if A is lower bounded. This in turn implies by the Heine-Borel property of finite dimensional Euclidean spaces that it has a cluster point. We assume that A is lower bounded and consider the sequence $u^{(k)} = (u_1^{(k)}, \dots, u_S^{(k)})$ produced by Algorithm 4.2. As in the proof of Theorem 4.7 we see that, for any $k \in \mathbb{N}$, there is a local minimizer $u^{(k),*} = (u_1^{(k),*}, \dots, u_S^{(k),*})$ of (4.7) such that

$$\|u^{(k)} - u^{(k),*}\| \leq C_2\delta_k, \tag{4.115}$$

where C_2 is a constant which does not depend on k. By Lemma 4.12, $u^{(k),*}$ is a minimizer of F_ρ on $\mathcal{A}^{\mathcal{P}}$ (where $\mathcal{P}$ is the partitioning induced by $u^{(k),*}$). Thus,

$$\tfrac{1}{S}\sum\nolimits_{s=1}^{S} \|Au_s^{(k),*} - f\|^2 \leq F_\rho(u^{(k),*}) \leq \|f\|^2$$

which follows from choosing the zero function as a candidate. This implies

$$\tfrac{1}{S}\sum\nolimits_{s=1}^{S} \|Au_s^{(k),*}\|^2 \leq 4\|f\|^2. \tag{4.116}$$

Now as A is lower bounded, there is a constant $c > 0$ such that

$$\|u^{(k),*}\|^2 = \tfrac{1}{S}\sum\nolimits_{s=1}^{S} \|u_s^{(k),*}\|^2 \leq \tfrac{1}{S}\sum\nolimits_{s=1}^{S} c^2\|Au_s^{(k),*}\|^2 \leq 4c^2\|f\|^2, \tag{4.117}$$

where we used (4.116) for the last inequality. By combining this estimate with (4.115) we obtain

$$\|u^{(k)}\| \leq \|u^{(k)} - u^{(k),*}\| + \|u^{(k),*}\| \leq C_2\delta_k + 2c\|f\|. \tag{4.118}$$

Since δ_k was chosen as a sequence converging to zero, (4.118) shows that the sequence $u^{(k)}$ is bounded which implies that it has cluster points which completes the proof. □

4.4. Experimental results

In this section, we illustrate the potential of our methods for different imaging tasks. We start out by providing the necessary implementation details. Next, we compare the results of the quadratic relaxation (4.7), i.e., Algorithm 4.1, and of the Potts problem (4.4), i.e., Algorithm 4.2. We continue with applying Algorithm 4.2 to blurred image data and to the reconstruction from incomplete Radon data. Finally, we consider the image partitioning problem which corresponds to the classical Potts model.

Implementation details. We implemented Algorithm 4.1 and Algorithm 4.2 for the coupling schemes in (4.9) and the 8-neighborhood discretization, i.e., the set of compass and diagonal directions $(1,0),(0,1),(1,1),(1,-1)$ with weights $\omega_{1,2}=\sqrt{2}-1$ and $\omega_{3,4}=1-\frac{\sqrt{2}}{2}$.

Concerning Algorithm 4.1, we observed both visually and quantitatively appealing results for relaxed step-sizes $L_\rho^\lambda = L_\rho\big(\lambda + (1-(n+1)^{-1/2})(1-\lambda)\big)$ where the parameter $0<\lambda\leq 1$ is chosen empirically and where L_ρ denotes the estimate in Lemma 4.9. We stopped the iterations when the closeness condition (4.23) was fulfilled and the iterates did not change anymore, i.e., when $\|u_1^{(n)}-u_1^{(n-1)}\|/(\|u_1^{(n)}\|+\|u_1^{(n-1)}\|)$ and $\|u_2^{(n)}-u_2^{(n-1)}\|/(\|u_2^{(n)}\|+\|u_2^{(n-1)}\|)$ became smaller than 10^{-6}. The result of Algorithm 4.1 was transformed into a feasible solution of (4.5) by applying the projection procedure described in Section 4.2.2 (Procedure 4.1). We initialized the algorithm with the result of 1000 Landweber iterations with step-size $1/\|A\|^2$ w.r.t. the least squares problem induced by the linear operator A and data f.

Concerning Algorithm 4.2, we initialized the coupling parameter by $\rho^{(0)}=10^{-3}$ in all experiments and incremented it by the factor $\tau=1.05$ in each outer iteration. The δ-sequence was chosen as $\delta^{(k)}=\frac{1}{\eta\rho^{(k)}}$ with $\eta=0.95$ for the full coupling scheme and $\eta=0.98$ for coupling consecutive variables only. Analogously to Algorithm 4.1, we performed step *(i)* of Algorithm 4.2 with the relaxed step sizes $L_\rho^\lambda = L_\rho\big(\lambda + (1-(n+1)^{-1/2})(1-\lambda)\big)$ for an application-dependent parameter $0<\lambda\leq 1$ and for the estimate L_ρ in Lemma 4.9. We stopped the iterations when the relative discrepancy between the first two splitting variables $\|u_1^{(k)}-u_2^{(k)}\|/(\|u_1^{(k)}\|+\|u_2^{(k)}\|)$ became smaller than 10^{-6}. We initialized Algorithm 4.2 with $A^T f$.

Comparison of Algorithm 4.1 and Algorithm 4.2. We compare Algorithm 4.1 and Algorithm 4.2 for blurred image data, that is, the linear operator A in (4.2) amounts to the convolution by a kernel K. In the present experiment, the kernel was a Gaussian kernel with standard deviation $\sigma=3$ and size $6\sigma+1$. We coupled all splitting variables and chose the step-size parameter $\lambda=0.4$ for Algorithm 4.1 and $\lambda=0.35$ for Algorithm 4.2, respectively. In Figure 4.3, we applied both methods to a blurred natural image. While both algorithms yield reasonable partitioning results, Algorithm 4.2 provides smoother edges than Algorithm 4.1 which further produces some smaller segments (at the treetops).

Blurred image data. For the following experiments, we focus on Algorithm 4.2. In case of motion blur, we set the step-size parameter to $\lambda=0.25$, while for Gaussian blur we used $\lambda=0.35$ as we did in Figure 4.3. We compare the proposed method with the Ambrosio-Tortorelli

(a) Original [111]

(b) Blurred data

(c) Algorithm 4.1, $\gamma = 0.1$, $\varepsilon = 2\|f\|$

(d) Algorithm 4.2, $\gamma = 0.1$

Figure 4.3.: Partitioning of a blurred image with Algorithm 4.1 and Algorithm 4.2. The image was blurred by a Gaussian kernel of standard deviation 3. We note that both methods provide reasonable partitionings. In general, Algorithm 4.2 provides smoother edges than Algorithm 4.1 (see, e.g., the boundary between meadow and forest or the cow's back). Further, Algorithm 4.1 produces some smaller segments around the treetops.

approximation [3] of the classical Mumford-Shah model (which tends to the Potts model for increasing smoothing penalty) which is given by

$$A_\varepsilon(u, v) = \gamma \int \varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon}\, dx + \alpha \int v^2 \|\nabla u\|^2\, dx + \frac{1}{2} \int (K * u - f)\, dx, \tag{4.119}$$

where v is an indicator function and $\varepsilon > 0$ is an edge smoothing parameter which must be chosen empirically in practice. The parameter $\gamma > 0$ controls the weight of the (smoothed) edge length penalty and the parameter $\alpha > 0$ controls the amount of smoothing. Thus, a higher value of α promotes solutions which are closer to being piecewise constant. In the limit $\alpha \to \infty$ minimizers of (4.119) are piecewise constant. The implementation follows the scheme of [10], i.e., the functional A_ε is alternately minimized w.r.t. u and v. This corresponds to iteratively solving the Euler-Lagrange equations

$$2\alpha v \|\nabla u\|_2^2 + \gamma \frac{v-1}{2\varepsilon} - 2\varepsilon\gamma \nabla^2 v = 0, \quad \text{and} \quad (K * u - f) * \widetilde{K} - 2\alpha \operatorname{div}(v^2 \nabla u) = 0, \tag{4.120}$$

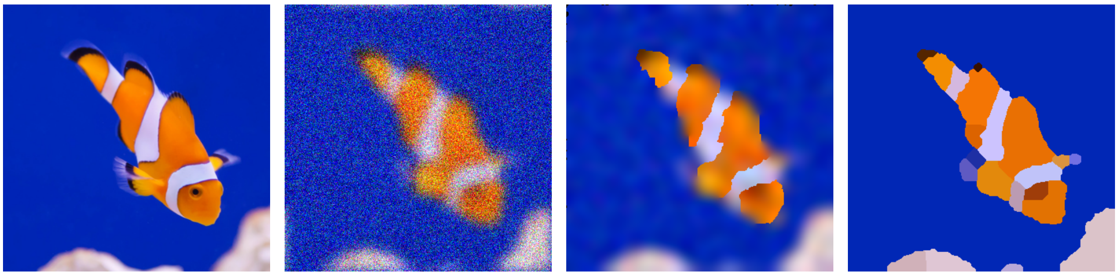
where $\widetilde{K}(x) := K(-x)$. The first equation is solved w.r.t. v using a minres solver and the second equation is solved by using the conjugate gradient method as described in [10]. The iterations were stopped when both variables did not change anymore, i.e., if both $\|u^{k+1} - u^k\|/(\|u^k\| + 10^{-6}) < 10^{-3}$ and $\|v^{k+1} - v^k\|/(\|v^k\| + 10^{-6}) < 10^{-3}$ hold.

In Figure 4.4, we restore a traffic sign from simulated horizontal motion blur. In the Ambrosio-Tortorelli approximation we set $\alpha = 10^5$ to obtain a piecewise constant result. We observe that both the Ambrosio-Tortorelli approximation and the proposed method restore the data to a human readable form. However, the Ambrosio-Tortorelli result shows clutter and blur artifacts. Our method provides sharp edges and shows less artifacts.

In Figure 4.5, we partition a natural image which was blurred by a Gaussian kernel and corrupted by Gaussian noise. We observed that the Ambrosio-Tortorelli result was heavily corrupted by artifacts for the previous choice $\alpha = 10^5$. This might be attributed to the underlying linear systems in scheme (4.120) which become severely ill-conditioned for large choices of the variation penalty α. Thus, we chose the moderate variation penalty $\alpha = 5$ which does

(a) Original (b) Blurred and noisy data (c) Ambrosio-Tortorelli, $\alpha = 10^5$, $\gamma = 0.8$, $\varepsilon = 1/2$ (d) Proposed, $\gamma = 0.15$

Figure 4.4.: Restoration of a traffic sign from simulated horizontal motion blur and Gaussian noise. The motion blur has 80 pixel length and the Gaussian noise has standard deviation $\sigma = 0.02$. The Ambrosio-Tortorelli result shows noise and blur artifacts as well as bumpy edges (see, e.g., the boundaries of the digits). The proposed result provides smoother edges and produces less clutter.



(a) Original (b) Blurred and noisy data (c) Ambrosio-Tortorelli, $\alpha = 5$, $\gamma = 0.03$, $\varepsilon = 1/30$ (d) Proposed, $\gamma = 0.2$

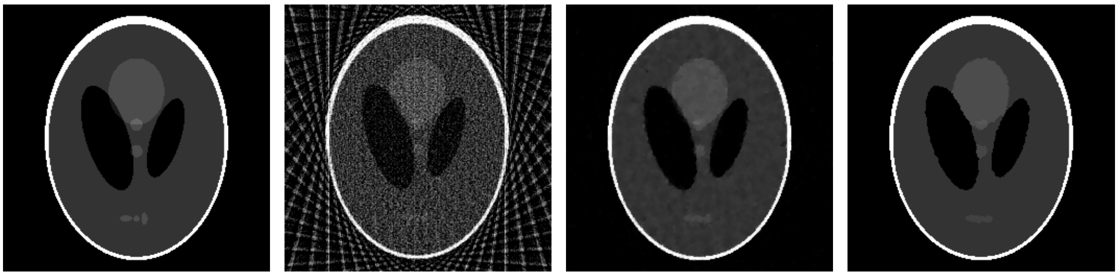
Figure 4.5.: Partitioning of an image blurred by a Gaussian kernel of standard deviation 7 and corrupted by Gaussian noise with $\sigma = 0.2$. The result of the Ambrosio-Tortorelli approximation does not yield a convincing partitioning of the scene, in particular many parts of the fish are merged with the background. The proposed approach provides a partitioning which reflects many parts of the fish.

only provide an approximately piecewise constant result. The result does not fully separate the background from the fish in terms of edges. On the other hand, the result of the proposed method yields a sharp boundary between the background and the fish. Further, it highlights various segments of the fish.

Reconstruction from undersampled Radon data. We consider the task of image reconstruction from (undersampled) Radon data which appears for example in computed tomography. The Radon transform of a bivariate function u is given by

$$Ru(\theta, s) = \int_{-\infty}^{\infty} u(s\theta + t\theta^{\perp})\mathrm{d}t, \tag{4.121}$$

where $s \in \mathbb{R}$, $\theta \in S^1$ and $\theta^{\perp} \in S^1$ is (counterclockwise) perpendicular to θ; see [150]. (We here denote by S^1 the unit sphere in $\mathbb{R}^2$.) In our experiments, we use a discrete variant of the



(a) Original

(b) FBP (Ram-Lak Filter)
MSSIM: 0.085

(c) TV, $\mu = 2.35$
MSSIM: 0.938

(d) Proposed, $\gamma = 3$
MSSIM: 0.984

Figure 4.6.: Reconstruction of the Shepp-Logan phantom from undersampled discrete Radon data (25 projection angles) corrupted by Gaussian noise with standard deviation $\sigma = 0.7$. The proposed method provides a genuine piecewise constant reconstruction and the MSSIM is improved by the factor 11.58 for filtered backprojection and by 1.05 for total variation, respectively.

Radon transform which was created with the AIR tools software package [91]. Concerning our method, we employed coupling of consecutive splitting variables only and used the step-size parameter $\lambda = 0.11$. We quantify the reconstruction quality with the mean structural similarity index (MSSIM) which is bounded from above by one, where higher values indicate better results (cf. Section 2.4).

We compare the proposed method to classical filtered back projection (FBP) which is the standard method in practice [159]. We used the Matlab implementation of the FBP together with the standard Ram-Lak filter. Furthermore, we compare our method with total variation (TV) regularization [169]. The TV result is obtained by minimizing the convex functional $\|Ru - f\|_2^2 + \mu\|\nabla u\|_1$ with parameter $\mu > 0$. We used the Chambolle-Pock algorithm for the minimization [51] and the corresponding model parameter μ was tuned w.r.t. the MSSIM index.

In Figure 4.6, we show the reconstruction results for the Shepp-Logan phantom from undersampled (25 angles) and noisy Radon data. The standard filtered back projection method (FBP) produces strong streak artifacts which are typical for angular undersampling, and the reconstruction suffers from noise. The TV result and the proposed method both provide considerably improved reconstruction results. Yet, the proposed method achieves a higher MSSIM value than TV and the result is less grainy.

Image partitioning. Finally, we consider the classical Potts problem which corresponds to $A = \mathrm{id}$ in (4.2). Our method uses the full coupling scheme and the step-size parameter was chosen as $\lambda = 0.55$.

To put our result in context we compared it with the results of the L_0 gradient smoothing method of Xu et. al [208], the state-of-the-art α-expansion graph cut algorithm based on max-flow/min-cut which uses the library GCOptimization 3.0 of Veksler and Delong [27, 26, 127] and the Ambrosio-Tortorelli model. The L_0 gradient smoothing [208] uses a parameter $\kappa > 1$ to control the convergence speed and a smoothing weight ν. In our experiments, we set $\kappa = 1.01$ and $\nu = 0.1$. For graph cuts we used the same neighborhood weights and jump penalty as for the proposed method. The discrete labels were computed via k-means.

(a) Original

(b) Noisy data

(c) Ambrosio-Tortorelli, $\gamma = 0.5$, $\alpha = 10^5$, $\varepsilon = 1/100$

(d) L_0 gradient smoothing, $\kappa = 1.01$, $\nu = 0.1$

(e) Graph cuts, $\gamma = 0.4$, 256 labels

(f) Proposed, $\gamma = 0.4$

Figure 4.7.: Partitioning of a natural image corrupted by Gaussian noise with standard deviation $\sigma = 0.2$. The Ambrosio-Tortorelli result is noisy and corrupted by clutter. The L_0 gradient smoothing oversegments the large window on the left-hand side, while it smooths out details of the cross on the right-hand side. The proposed result is visually competitive with the state-of-the-art graph cuts result.

In Figure 4.7, we partition a natural image which was corrupted by Gaussian noise. The Ambrosio-Tortorelli result shows clutter and remains noisy. The L_0 gradient smoothing oversegments the textured window area, while it smooths out details of the cross. The state-of-the-art graph cuts method and the proposed method both provide satisfying results which are visually comparable. Hence, while the proposed method can handle general linear operators A, the quality of the results for $A = \mathrm{id}$ is comparable with the state-of-the-art graph cut algorithm for the special case $A = \mathrm{id}$.

4.5. Summary of the chapter

In this chapter, we developed new iterative minimization strategies for the inverse multivariate Potts model and corresponding quadratic penalty relaxations. Our schemes are based on majorization-minimization using surrogate functionals. Compared to related schemes for sparsity problems, i.e., minimizing $\|u\|_0$ instead of $\|\nabla u\|_0$, [75, 22, 142, 143], the proposed schemes did not decompose into pointwise thresholding, but into non-separable linewise problems which could be solved by dynamic programming.

Subsequently, we gave a convergence analysis for the proposed algorithms. In particular, we showed convergence of the quadratic penalty relaxation scheme to a local minimizer of the (still NP-hard) relaxed problem. We further proposed a projection procedure to obtain a feasible solution for the Potts problem from a solution of the quadratic penalty relaxation such that a minimizer of the Potts problem is nearby. Regarding the scheme for the non-relaxed problem, we proved convergence towards local minimizers on subsequences. These results are comparable with the results of [142, 143] for sparsity problems, where we had to deal with the additional challenge of subproblems which did not pointwise decompose.

We illustrated the applicability of the proposed schemes in numerical experiments. This included deconvolution problems, joint reconstruction and segmentation from undersampled Radon data and the image partitioning problem.

5. The Multi-Channel Potts Prior for Multi-Spectral Computed Tomography

In this chapter, we consider image reconstruction from *multi-spectral* computed tomography (CT) data with the Potts prior. This chapter is based on the publication [116].

Organization of the chapter. First, in Section 5.1, we give an overview and discuss related work. In Section 5.2, we give a brief description of the measurement process in multi-spectral CT and formulate the corresponding reconstruction problem. In Section 5.3, we discuss the multi-channel Potts prior. We recall the multi-channel Potts model and briefly recall the ADMM approach of [183] in Section 5.4. In Section 5.5, we propose a new approach based on the superiorization of the conjugate gradient method (CG) with the multi-channel Potts prior, which we call Potts S-CG. A comparison of Potts ADMM and Potts S-CG is given in Section 5.6. In Section 5.7, we illustrate the potential of Potts ADMM and Potts S-CG by comparing them to existing TV-type approaches in numerical experiments. More precisely, we compare the reconstructions of two phantoms from simulated multi-spectral CT measurements. Section 5.8 concludes the chapter with a summary.

5.1. Overview and related work

In this chapter, we consider the (multi-channel) Potts prior for the multi-channel reconstruction problem in multi-spectral X-ray computed tomography (CT). X-ray imaging entails *polychromatic* X-ray sources, that is, the emitted photons have a spectrum of energies. Conventional CT detectors do not capture different energies. However, there are materials which may not be distinguished from another in conventional CT as their linear attenuation coefficients (LAC) are nearly equal. Yet, their LAC's might differ when the whole energy spectrum is considered. Thus, such materials may be distinguished, when measurements for multiple energy-levels are available. This phenomenon may be utilized by using *energy-discriminating* photon counting detectors [172].

As in conventional CT, due to shot noise, sampling effects and modeling effects, the reconstruction process in multi-spectral CT needs to be regularized. This is often done by imposing prior structural knowledge on the unknown function in terms of a penalty called prior. In principle, one could apply the existing priors and models for conventional CT to reconstruct each channel of the *multi-channel* images in multi-spectral CT separately. However, it is well-known that the channel images in multi-spectral CT are strongly correlated in the sense that region boundaries are spatially aligned across the channels. Therefore, an appropriate prior should exploit this property by enforcing the spatial correlation of the channels and aid the reconstruction of particularly noise-prone channels by employing the less problematic channels. We will see that the multi-channel version of the Potts prior has the appealing property that the edges of the channels are enforced to share spatial positions. This property is especially attractive for multi-spectral CT reconstruction as mentioned above, which we illustrate by comparing the multi-channel Potts prior to existing TV-based priors.

We investigate two reconstruction schemes for multi-spectral CT problems based on the Potts prior which are based on energy minimization and superiorization, respectively. In particular, we apply the multi-channel Potts prior both in terms of the Potts model and in terms of a new *Potts-based superiorization approach*. On the algorithmic side, we use an ADMM approach for the minimization problem corresponding to the Potts model. Regarding the superiorization approach, we choose the conjugate gradient method as the basic algorithm. We identify these approaches as suitable choices by comparing them to a penalty method for the Potts model and to a superiorization method using Landweber iterations.

To the author's knowledge the (multi-channel) Potts prior has neither been considered in the context of multi-spectral CT nor in the context of the superiorization methodology yet.

Related work. For related work concerning the Potts model, we refer to Chapter 4. We start with related work on multi-spectral CT reconstruction and material decomposition. Kazantsev et al. [114] consider energy-discriminating photon-counting detectors and propose a regularizer based on directional TV, which chooses new reference channels in each iteration in a probabilistic manner. They compare their method to other existing TV-based approaches by reconstructing the geocore phantom, which we also consider in our experiments. Toivanen et al. [187] studied reconstruction from three channels obtained from low dose acquisition protocols. A multi-channel reconstruction method which models the reconstruction as the sum of low-rank and a sparse matrix was proposed in [82]. In [168], the authors use total nuclear variation to regularize the reconstruction. Furthermore, tensor nuclear norm based regularizers to enforce channel-coupling were proposed in [171]. Material decomposition of multi-spectral CT images was considered in [141] and material decomposition of medical dual-energy CT images was considered in [65]. In [69], the authors propose a method to solve the nonlinear decomposition problem in multi-spectral X-ray imaging.

Next, we give related work on superiorization. The terms "superiorization" and "perturbation resilience" first appeared in [60]. Important notions were also introduced in [97]. A condensed introduction and examples can be found, e.g., in [45] and the references therein. The journal Inverse Problems had a special issue on superiorization [46]. Byrne [36] studied similarities between some superiorization algorithms and optimization methods. In [96], the (preconditioned) conjugate gradient algorithm and algebraic reconstruction techniques (ART) are perturbed by non-ascending directions w.r.t. the total variation prior. The methods were applied to (single-channel) tomographic image reconstruction problems. Perturbations by means of the TV prior have been considered in [218] as well. The potential of the superiorization methodology was illustrated in [47] by comparing it to convex optimization approaches. For an extensive overview, we refer the interested reader to the continuously updated bibliography on superiorization in [44].

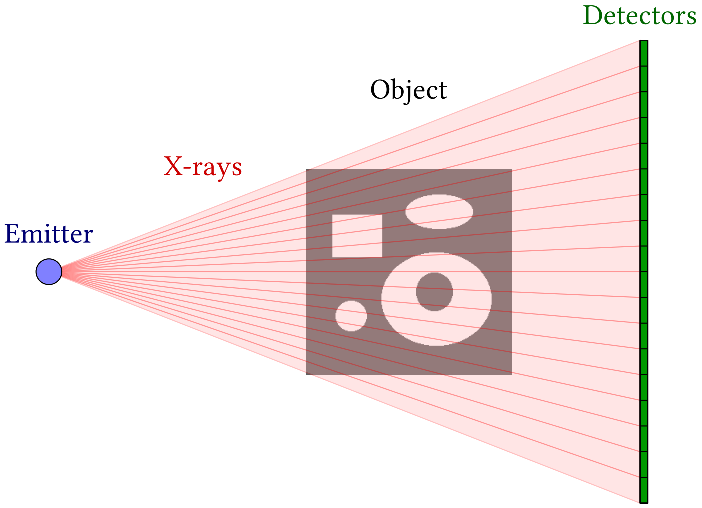


Figure 5.1.: Illustration of the (multi-spectral) CT measurement model for a single angle and the fan-beam geometry. The emitter (or, X-ray source) sends photons in the form of X-rays through the object of interest. The detectors count the number of incident photons, which were not absorbed by the object. Typically, dense materials absorb a higher amount of photons, so that the corresponding detectors count fewer photons. The emitter and detectors are rotated around the object to obtain measurements from multiple angles.

5.2. Multi-spectral CT imaging

In this section, we briefly explain the measurement process, i.e., the forward model of multi-spectral CT. In particular, we derive the mathematical formulation of the reconstruction problem for which we develop our methods. We start with conventional single-energy measurements and subsequently generalize this to multi-spectral measurements. For more details on multi-spectral CT imaging, including general measurement setups, we refer to [95, 146].

5.2.1. Forward model

Basic single-channel line integral model and Beer's law. The measurement process in CT is typically modeled in terms of line integrals. We provide a brief description of this model. To this end, we assume an X-ray source which emits a number of photons I_0, also called intensity flux, along a line $\mathcal{L}$ through an object. Further, we assume a detector placed behind the object, which counts the number of photons I_1 incident to the detector. As the photons pass through the object, a certain amount of them is absorbed by the materials they hit. This phenomenon is modeled by the attenuation coefficient function $\hat{u}$, that is, the relative change (here, decrease) of intensity of I satisfies $I'(z)/I(z) = -\hat{u}(z)$. (We note that $\hat{u}(z)$ is material-dependent as one material may absorb more photons than another one, e.g., bone tissue absorbs more photons than water.) Here, $z \in \Omega$ and $\Omega \subset \mathbb{R}^2$ denotes the domain of the object. Integration along $\mathcal{L}$ yields

$$\int_{\mathcal{L}} \hat{u}(z)\, dz = -\int_{\mathcal{L}} I'(z)/I(z)\, dz = \log I_0 - \log I_1, \tag{5.1}$$

which, after taking the exponential on both sides, corresponds to

$$I_1 = I_0 \exp\Big(\int_{\mathcal{L}} -\hat{u}(z)\, dz\Big). \tag{5.2}$$

Equation (5.2) is known as *Beer's law.* To sum up, the detector counts the number of incident photons I_1 and the number of emitted photons I_0 is known from calibration.

Typically, the above measurements are taken for many lines $\mathcal{L}$, i.e., many X-rays are shot through the object from multiple (equidistant) angles. The quantities of interest are the unknown *linear attenuation coefficients* (LAC)[1] $\hat{u}(z)$ which are indirectly given in terms of line integrals (5.2). If the attenuation coefficients $\hat{u}$ have been reconstructed, they allow us to obtain information from inside the object in a destruction-free manner. For instance, information of interest might be the location of bones in organic tissue or cavities in compound bodies. We give an illustration of the measurement process for a single angle and the so-called fan-beam geometry in Figure 5.1.

Multi-spectral version of Beer's law. The model (5.2) makes the implicit assumption that the photons emitted by the photon source all have the same energy. However, X-ray imaging entails *polychromatic* X-ray sources, that is, the emitted photons have a spectrum of energies. In conventional CT, the involved energies are accumulated and the photons are assumed to have a common energy. In contrast, energy-discriminating photon counting detectors provide multi-spectral measurements. To this end, we consider the multi-spectral version of Beer's law

$$I_1(\varepsilon) = I_0(\varepsilon) \exp\left(\int_{\mathcal{L}} -\hat{u}(z,\varepsilon)\, dz\right), \tag{5.3}$$

where ε denotes the energy, I_1 is the spectrum of the X-ray beam incident on the detector and $z \in \Omega$ denotes the spatial position. Further, $\hat{u}$ holds the *energy dependent* linear attenuation coefficients that need to be reconstructed and I_0 is the *energy dependent* intensity flux of the X-ray source corresponding to ray $\mathcal{L}$ from the source to a given detector element.

Discrete measurement model. In the following, we consider M detectors and assume that the measurements (5.3) are taken from p equidistant angles. Thus, a total number of $m = pM$ discrete measurements are available. Towards a discrete model, we discretize the (unknown) function $\hat{u}$ on the continuous domain Ω which results in a function u on an $n \times n$ pixel grid Ω'. It is given by

$$u(j,\varepsilon) = \sum_{z\in\Omega} \chi_j(z)\hat{u}(z,\varepsilon), \tag{5.4}$$

where χ_j denotes the characteristic function[2] corresponding to the pixel $j \in \Omega'$. From (5.3) and for pairs of source-detector positions indexed by i, we now obtain

$$I_{1,i}(\varepsilon) = I_{0,i}(\varepsilon) \exp\Big(-\sum_{j\in\Omega'} A_{ij}\, u(j,\varepsilon)\Big), \quad i = 1,\dots,m, \tag{5.5}$$

---

[1] The unit of the LAC is typically 1/cm.

[2] For a set B, $\chi_B(x) = \begin{cases} 1 & \text{if } x \in B, \\ 0 & \text{if } x \notin B. \end{cases}$

where we denote by

$$A_{ij} := \int_{\mathcal{L}_i} \chi_j(z)\, dz, \quad i = 1, \dots, m, \quad j \in \Omega', \tag{5.6}$$

the intersection length of ray i with pixel j. (In particular, $A_{ij} = 0$ if the ray i does not intersect pixel j.) We assume that each photon-counting detector acquires the measurements on an interval of energies $[\varepsilon_c, \varepsilon_{c+1})$. Hence, the detectors provide the measurements

$$Y_{i,c} = \int_{\varepsilon_c}^{\varepsilon_{c+1}} I_{0,i}(\varepsilon) \exp\Big(- \sum_{j \in \Omega'} A_{ij} u(j, \varepsilon)\Big)\, d\varepsilon, \quad i = 1, \dots, m,\ c = 1, \dots, C. \tag{5.7}$$

After discretizing the energy spectrum with step-size $\delta > 0$, we obtain

$$Y_{i,c} \approx \sum_{\varepsilon = \varepsilon_c, \varepsilon_c + \delta, \dots, \varepsilon_{c+1} - \delta} I_{0,i}(\varepsilon) \exp\Big(- \sum_{j \in \Omega'} A_{ij}\, u(j, \varepsilon)\Big), \quad i = 1, \dots, m,\ c = 1, \dots, C. \tag{5.8}$$

The following simplified measurement model becomes increasingly accurate for fine energy resolutions, i.e., if δ is small, [114]:

$$Y_{i,c} \approx I_{0,i}(\varepsilon_c) \exp\Big(- \sum_{j \in \Omega'} A_{ij}\, u(j, \varepsilon_c)\Big), \quad i = 1, \dots, m,\ c = 1, \dots, C. \tag{5.9}$$

After taking the logarithm, (5.9) corresponds to

$$f_{i,c} \approx \sum_{j \in \Omega'} A_{ij}\, u(j, \varepsilon_c), \quad i = 1, \dots, m,\ c = 1, \dots, C, \tag{5.10}$$

where we denote $f_{i,c} = -\log\Big(\frac{Y_{i,c}}{I_{0,i}(\varepsilon_c)}\Big)$. Towards a compact notation, we let $f_c \in \mathbb{R}^m$ hold the $f_{i,c}$ for all rays $i = 1, \dots, m = pM$ and define the linear operator $A = (A_{i,j})_{i=1,\dots,m,\, j \in \Omega'} : \mathbb{R}^{n \times n} \to \mathbb{R}^m$, which holds the *ray incidence geometry* of the measurement setup. Hence, we obtain the linear measurement model

$$f_c \approx A u_c \tag{5.11}$$

for the channels $c = 1, \dots, C$.

It is common to assume shot noise for photon counting detectors, that is, the measurements are Poisson distributed. Thus, in a setup incorporating noise, the corresponding variant of (5.9) is given by

$$Y_{i,c} \sim \mathrm{Poiss}\Big\{ I_{0,i}(\varepsilon_c) \exp\Big(- \sum_{j \in \Omega'} A_{ij}\, u(j, \varepsilon_c)\Big)\Big\}. \tag{5.12}$$

As a result, note that f_c in (5.11) is not deterministic due to (5.12). The multi-spectral CT reconstruction problem corresponds to determining $u_1, \dots, u_C$ from the measurements (5.11).

Remark 5.1. We note that the made assumptions guarantee a measurement model which is linear. This model is frequently used in the literature [89, 82, 171, 168, 114, 187]. We briefly recall its known limitations.

(i) For coarse energy resolutions, the step between (5.8) and (5.9) becomes increasingly inaccurate.

(ii) We made the implicit assumption that the energy-dependent detection answers in (5.7) are represented by characteristic functions $\chi_{[\varepsilon_c, \varepsilon_{c+1})}$ of the pairwise disjoint intervals $[\varepsilon_c, \varepsilon_{c+1})$. More realistic detector answer functions would not be constant on the corresponding energy spectrum nor would they have disjoint supports.

5.2.2. Reconstruction

In multi-spectral CT, it is common to use model-based reconstruction approaches as a direct inversion of (5.11) is not feasible due to the present shot noise (5.12), sampling effects, and necessary simplifications in the modeling. Instead, the reconstruction u is usually modeled as the minimizer of an energy function. Typically, such an energy function corresponds to the (weighted) sum of a data term $\mathcal{D}$ and a regularizer $\mathcal{R}$, i.e., the reconstruction is modeled as the solution of a minimization problem of the form

$$\operatorname*{argmin}_{u \in \mathbb{R}^{n \times n \times C}} \sum_{c=1}^{C} \mathcal{D}(Au_c, f_c) + \gamma\, \mathcal{R}(u). \tag{5.13}$$

The data term $\mathcal{D}$ promotes closeness to the data f and $\mathcal{R}$ imposes regularity on u by penalizing deviations from a priori fixed assumptions on u. The parameter $\gamma > 0$ balances the two terms.

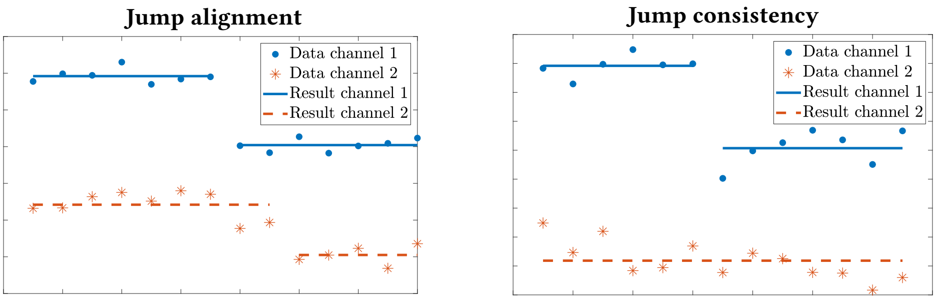
Data term. Concerning the data term $\mathcal{D}$, it is common in CT to employ the penalized weighted least squares model (PWLS) [65, 114, 171]. The PWLS serves as a (computationally more tractable) quadratic approximation of the log-likelihood function associated with the Poisson distribution in (5.12); see [170]. The PWLS is given by

$$\mathcal{D}(Au_c, f_c) = \|Au_c - f_c\|_{W_c}^2 = \|W_c^{\frac{1}{2}} Au_c - W_c^{\frac{1}{2}} f_c\|^2, \tag{5.14}$$

where each $W_c \in \mathbb{R}^{m \times m}$ is a diagonal matrix which weighs the measurements within the respective channel. More precisely, the entries of W_c are chosen as the number of detected photons in the c-th energy bin (5.9). (Please note that $\|\cdot\|$ corresponds to the Frobenius norm, i.e., $\|u\|^2 = \sum_{i,j,c} u_{ijc}^2$.)

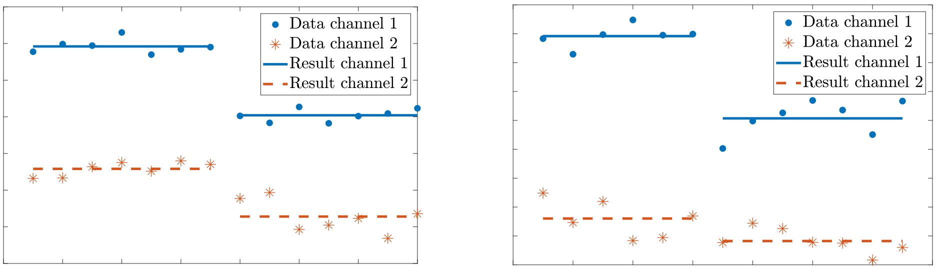
Regularizing term. In general, an appropriate regularizing term $\mathcal{R}$ in (5.13) should enforce prior knowledge on the unknown result u. For example, compound solid bodies are (approximately) piecewise homogeneous so that an adequate regularizer may enforce piecewise constancy on u. We note that it is generally assumed that the channels $u_1, \ldots, u_C$ of multi-spectral CT images are strongly correlated [114, 187, 65, 171, 115]. In particular, it was observed that the edges in the channels are spatially correlated, i.e., they are located at the same positions across the channels. As a result, a regularizer $\mathcal{R}$ in (5.13) should enforce prior structural knowledge on each channel *as well as* strong correlation between the channels.

In this chapter, we propose to use the (multi-channel) Potts prior for $\mathcal{R}$, which combines structural knowledge and strong channel correlation as explained in the next section.



(a) Channel-wise Potts prior

(b) Channel-wise Potts prior



(c) Multi-channel Potts prior

(d) Multi-channel Potts prior

Figure 5.2.: Channel-coupling with the multi-channel Potts prior. *First column*: (a) the channel-wise Potts prior tends to produce jumps at shifted positions in the channels. (c) The multi-channel Potts prior enforces the jumps to be spatially aligned across channels (called the *jump alignment* property of the multi-channel Potts prior). *Second column*: (b) the channel-wise Potts prior may open jumps in a single channel only. (d) The jumps produced by the multi-channel Potts prior are typically present in all channels. Hence, the multi-channel Potts prior provides *jump consistency*.

Remark 5.2. An instance of (5.13) (for single-channel signals) is given by the inverse Potts model (4.4) in Chapter 4. There, $\mathcal{D}$ was given by the squared ℓ_2-norm and $\mathcal{R}$ was given by the Potts prior (as defined below (4.4)), i.e., the results are piecewise constant. Unfortunately, the methods of Chapter 4 are not well-suited for multi-spectral CT: the methods depend on the spectral norm of the forward operator A (cf. Section 2.2), which becomes very large for realistic multi-spectral measurements. Further, due to the Poisson noise in (5.12), the ℓ_2-data term in (4.4) is not an appropriate choice; see above.

5.3. The multi-channel Potts prior

In the following, we describe the multi-channel Potts prior and explain how it enforces strong correlation between channels. This property is especially beneficial for the reconstruction of multi-channel images from multi-spectral CT measurements (5.11) as stated in Section 5.2. Recall that the single-channel Potts prior was defined below (4.4).

We consider a multi-channel image $\hat{u} : \Omega \to \mathbb{R}^C$ with open domain $\Omega \subset \mathbb{R}^2$ and codimension $C \in \mathbb{N}$. The c-th component function of $\hat{u}$ is denoted by $\hat{u}_c$. Further, we denote the (distributional) Jacobian of $\hat{u}$ analogously to the single-channel case by $\nabla \hat{u}$. Similarly to the single-channel Potts prior, the multi-channel Potts prior penalizes the length of the support of the (distributional) Jacobian of $\hat{u}$, i.e.,

$$\|\nabla\hat{u}\|_0 = \text{length}\left(\left\{x \in \Omega : \nabla\hat{u}(x) \neq 0\right\}\right), \tag{5.15}$$

where length is understood as the one-dimensional Hausdorff measure as in (4.3). The discrete counterpart of (5.15) for the function $u : \Omega' \to \mathbb{R}^C$ on the discrete lattice $\Omega' = \{1, \dots, n\} \times \{1, \dots, n\}$ is now given by

$$\|\nabla\hat{u}\|_0 \approx \sum_{s=1}^{S} \omega_s \|\nabla_{d_s} u\|_0. \tag{5.16}$$

Here, the right-hand side of (5.16) counts the number of finite difference vectors, which have at least one non-zero entry, i.e.,

$$\|\nabla_{d_s} u\|_0 = \left|\left\{x \in \Omega' : x + d_s \in \Omega',\ u_c(x) \neq u_c(x + d_s) \text{ for at least one } c \in \{1, \dots, C\}\right\}\right|. \tag{5.17}$$

The integer vectors $d_s \in \mathbb{Z}^2$ and weights $\omega_s > 0$ are chosen as described in (4.4). Due to (5.17), the value of the multi-channel Potts prior for a jump between x and $x + d_s$ in all channels is the same as for opening a jump in a single channel only. On the one hand, this enforces potential jumps at close positions to be spatially aligned across channels in view of the lower costs compared with multiple jumps in different channels (*jump alignment*). On the other hand, jumps are always introduced for all channels simultaneously as higher closeness to the data can be ensured (*jump consistency*). Thus, the edges of multi-channel images regularized by the multi-channel Potts prior are spatially aligned across all channels, or, in other words, the spatial locations of the edges in the channels are enforced to be completely correlated.

In Figure 5.2, we illustrate the jump alignment and the jump consistency provided by the multi-channel Potts prior by comparing it to the channel-wise Potts prior for dual-channel one-dimensional data.

In the following two sections, we work out two approaches to the numerical reconstruction of piecewise constant multi-channel volume functions by means of the (multi-channel) Potts prior. Firstly, we consider the multi-spectral Potts model, which corresponds to a minimization problem of the form (5.13) with the Potts prior as the regularizer $\mathcal{R}$. We approach the Potts model numerically with the ADMM scheme of [183]. Secondly, we derive a new approach based on the Potts prior, which is *not* of the form (5.13), i.e., it is not based on an energy minimization model. Rather, we use the Potts prior to *superiorize* an existing iterative solver for (weighted) least squares problems – the conjugate gradient algorithm (CG). In particular, the Potts prior is viewed as a target function which *perturbs* the iterations of CG towards Potts superiorized results (cf. Section 2.2).

5.4. Potts-based ADMM

The multi-spectral Potts model for the (linear) forward operator $A : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ and multi-spectral data $f \in \mathbb{R}^{m\times C}$ is given by the following minimization problem

$$u^* = \operatorname*{argmin}_{u:\Omega\to\mathbb{R}^C} \sum_{c=1}^{C} \mathcal{D}(Au_c, f_c) + \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{d_s} u\|_0. \tag{5.18}$$

Here, $\mathcal{D}$ is the PWLS data fidelity term from (5.14), which ensures channel-wise closeness of u to the data. The nonnegative parameter γ balances the data fidelity and the Potts prior. It is worth considering the limit cases of γ: for $\gamma \to 0$ the minimizer of (5.18) has minimal data fidelity and no regularization takes places. For $\gamma \to \infty$ the minimizer of (5.18) will be a constant function (with minimal data fidelity).

Algorithmic approach with the ADMM. To obtain good approximate solutions of (5.18), we follow the approach first presented in [181] and extended to inverse problems in [183], which is based on the ADMM. We briefly recall the scheme for the reader's convenience and provide the necessary adaptations to multi-spectral CT.

The approach uses the ADMM to approach the multi-channel Potts problem in (5.18). As a first step, we recast (5.18) as a constrained problem. To this end, we introduce splitting variables $u_1, \ldots, u_S$ (each corresponding to a direction d_s of the discrete gradient) and the variable v (corresponding to the data term) under the restriction that they are equal. Hence, we consider the constrained problem

$$\begin{aligned} \min_{v, u_1, \ldots, u_S} \sum_{c=1}^{C} \mathcal{D}(Av_c, f_c) + \sum_{s=1}^{S} \gamma\omega_s \|\nabla_{d_s} u_s\|_0 & \\ \text{subject to } u_s - u_t = 0 \text{ for all } 1 \leq s < t \leq S, & \\ v - u_s = 0 \text{ for all } s = 1, \ldots, S. & \end{aligned} \tag{5.19}$$

It is a basic but important observation that we did not alter the original problem, i.e., (5.18) and (5.19) are equivalent. The ADMM approach requires to form the augmented Lagrangian of (5.19) which is given by

$$\begin{aligned} \mathcal{L}_{\mu,\rho}\Big(v, \{u^s\}, \{\lambda^{s,t}\}, \{\tau^s\}\Big) &= \sum_{c=1}^{C} \mathcal{D}(Av_c, f_c) \\ &+ \sum_{s=1}^{S} \left\{ \gamma\omega_s \|\nabla_{d_s} u^s\|_0 + \tfrac{\rho}{2}\|v - u^s + \tfrac{\tau^s}{\rho}\|^2 - \tfrac{1}{2\rho}\|\tau^s\|^2 + \sum_{t=s+1}^{S} \|u^s - u^t + \tfrac{\lambda^{s,t}}{\mu}\|^2 - \tfrac{1}{2\mu}\|\lambda^{s,t}\|^2 \right\}. \end{aligned} \tag{5.20}$$

The constraints in (5.19) are now part of the functional in the form of the corresponding squared deviations weighted by the nonnegative parameters μ, ρ. The variables $\lambda_{s,t}, \tau_s \in \mathbb{R}^{n \times n \times C}$ are the corresponding Lagrange multipliers. In each ADMM iteration, the Lagrangian $\mathcal{L}$ is sequentially minimized w.r.t. v and $u_1, \ldots, u_S$. Further, a gradient ascent step is performed on the multipliers.

After some algebraic manipulation (see [183]; we also conduct a very similar derivation later in Section 7.3), the minimization of the Lagrangian w.r.t. the data variable v reads

$$\operatorname*{argmin}_{v} \mathcal{L}_{\mu,\rho} = \operatorname*{argmin}_{v} \sum_{c=1}^{C} \mathcal{D}(Av_c, f_c) + \tfrac{\mu S}{2} \left\| v - \tfrac{1}{S} \sum_{s=1}^{S} \left(u_s - \tfrac{\tau_s}{\rho} \right) \right\|^2, \tag{5.21}$$

i.e., (5.21) is a PWLS Tikhonov problem in view of the additional squared ℓ_2-terms. Furthermore, the minimization of the Lagrangian w.r.t. u_s is given by

$$\operatorname*{argmin}_{u_s} \mathcal{L}_{\mu,\rho} = \operatorname*{argmin}_{u_s} \tfrac{2\gamma\omega_s}{\rho+\mu(S-1)} \|\nabla_{d_s} u_s\|_0 + \left\| u_s - \tfrac{\rho v+\tau_s+\sum_{t=s+1}^{S}(\mu u_t-\lambda_{s,t})+\sum_{r=1}^{s-1}(\mu u_r+\lambda_{r,s})}{\rho+\mu(S-1)} \right\|^2 . \tag{5.22}$$

We elaborate on efficiently solving the subproblems (5.21)-(5.22) in the subsequent paragraph. As it is common when dealing with non-convex problems, we employ monotonically increasing sequences $(\mu_k)_{k\in\mathbb{N}}$, $(\rho_k)_{k\in\mathbb{N}}$ as coupling parameters. This allows the splitting variables to develop rather independently in the first iterations and forces them to become equal in the later iterations. We summarize the Potts ADMM approach in Algorithm 5.1.

Algorithm 5.1: Potts ADMM

```
Input: Forward operator $A \in \mathbb{R}^{m\times n^2}$, multi-spectral sinogram $f \in \mathbb{R}^{m\times c}$, PWLS weights
       $W \in \mathbb{R}^{m\times m\times C}$, stopping parameter tol $> 0$, jump penalty $\gamma > 0$
Output: $u^* \in \mathbb{R}^{n\times n\times C}$
Initialize $v^0, u_1^0, \dots, u_S^0, \lambda_{s,t}^0, \tau_s^0$ by zero.
$k \leftarrow 0$
repeat
    /* Solve the channel-wise PWLS Tikhonov problems */
    $v^{k+1} = \operatorname{argmin}_v \|Av - f\|_{W_c}^2 + \frac{\mu_k S}{2}\left\| v - \frac{1}{S}\sum_{s=1}^{S}\left(u_s^k - \frac{\tau_s^k}{\rho_k}\right)\right\|^2$
    /* Solve linewise Potts problems along the directions $d_s$ */
    for $s = 1, \dots, S$ do
        $w_s^k = \frac{\rho_k v_k + \tau_s^k + \sum\limits_{t=s+1}^{S}(\mu_k u_t^k - \lambda_{s,t}^k) + \sum\limits_{r=1}^{s-1}(\mu_k u_r^{k+1} + \lambda_{r,s}^k)}{\rho_k + \mu_k(S-1)}$
        $u_s^{k+1} = \operatorname{argmin}_{u_s} \frac{2\gamma\omega_s}{\rho_k+\mu(S-1)} \|\nabla_{d_s} u_s\|_0 + \|u_s - w_s^k\|^2$
    end
    /* Update the multipliers */
    $\lambda_{s,t}^{k+1} = \lambda_{s,t}^k + \mu_k\left(u_s^{k+1} - u_t^{k+1}\right)$  for all $s \neq t$
    $\tau_s^{k+1} = \tau_s^k + \rho_k\left(v^{k+1} - u_s^{k+1}\right)$  for all $s$
    /* Increase the coupling parameters */
    $\mu_k \leftarrow \mu_{k+1}$, $\nu_k \leftarrow \nu_{k+1}$
    $k \leftarrow k + 1$
until $\|u_s^k - u_{s+1}^k\|_\infty <$ tol for all $s = 1, \dots, S-1$ and $\|u_s^k - v^k\|_\infty <$ tol for all $s$;
return $u^* = \frac{1}{S}\sum_{s=1}^{S} u_s^k$
```

Solving the subproblems in the ADMM scheme. We explain how to solve the subproblems (5.21) and (5.22) efficiently. We start with the first subproblem (5.21). For arbitrary data $f, z \in \mathbb{R}^{n\times n\times C}$ and $\mu > 0$ it is given by

$$v^* = \operatorname*{argmin}_{v\in\mathbb{R}^{n\times n\times C}} \sum_{c=1}^{C} \|W^{\frac{1}{2}} A v_c - W^{\frac{1}{2}} f_c\|^2 + \tfrac{\mu}{2}\|v - z\|^2. \tag{5.23}$$

The PWLS Tikhonov problem (5.23) decomposes channel-wise into C problems. Hence, the unique minimizer v^* is channel-wise determined by the weighted normal equations

$$\left(A^T W_c A + \frac{\mu}{2} I\right) v_c^* = A^T W_c f_c + \mu z_c. \tag{5.24}$$

(Note that I denotes the identity.) Consequently, solving (5.23) corresponds to solving the linear system (5.24) for each $c = 1, \dots, C$. We use the conjugate gradient method to solve (5.24).

The second subproblem has the generic form

$$\operatorname*{argmin}_{u_s} \|u - f\|^2 + \gamma \|\nabla_{d_s} u_s\|_0 \tag{5.25}$$

for some data $f \in \mathbb{R}^{n\times n\times C}$ and $\gamma > 0$. Analogously to the subproblem (4.33) in Chapter 4, (5.25) decomposes into univariate Potts problems along the paths in f induced by d_s (cf. Figure 4.2). In contrast to (4.33), the univariate Potts problem we have to solve here are given w.r.t. multi-channel data, i.e.,

$$\operatorname*{argmin}_{w\in\mathbb{R}^{n\times C}} \|w - g\|^2 + \gamma \|\nabla w\|_0. \tag{5.26}$$

We describe the extensions to solve (5.26) with the solver of Chapter 3. Now the symbol $\mathcal{E}^I$ in (3.14) denotes the sum of the channel-wise approximation errors $\mathcal{E}_c^I = \min_{\overline{g}} \sum_{i\in I} |g_{i,c} - \overline{g}|^2$, i.e.,

$$\mathcal{E}^I = \sum_{c=1}^{C} \mathcal{E}_c^I. \tag{5.27}$$

As a direct consequence of (5.27) and Theorem 3.10, we obtain:

Corollary 5.3. *Algorithm 3.1 solves the univariate multi-channel Potts problem* (5.26) *in* $O(n^2 C)$ *time.*

Hence, the theoretical computational costs grow only linearly in the number of channels. Thus, the subproblems may be efficiently solved for vector-valued data with a large number of channels C as it appears in multi-spectral CT.

5.5. Potts-based superiorization

We propose a superiorization approach to the reconstruction problem in multi-spectral CT. Recall that superiorization approaches alternate between basic algorithmic operations and target function reduction steps according to (2.12). In this section, we propose methods which take the CG method (the variant given by Algorithm 2.1) as the basic algorithm and the (multi-channel) Potts prior (5.18) as the target function to obtain Potts superiorized, i.e., piecewise constant, solutions. As a result, the iterates are not CG iterates but perturbed versions of them, so that the classical result that the CG method converges (theoretically) after a fixed number of iterations (which is equal to the number of columns of the underlying system matrix) cannot be expected to hold anymore. However, we will see that the termination of its perturbed counterpart may still be ensured.

We recall that both energy minimization methods and superiorization typically result in two steps within an algorithmic scheme: a forward step followed by a regularizing step. However, the interpretation of these steps is different. Please note that basics on the superiorization methodology are given in Section 2.2.

5.5.1. Perturbing the CG method with the block-wise Potts prior

To keep things focused, we begin with single-channel measurements and the simplest anisotropic neighborhood for the Potts prior. In a subsequent section, we provide the extensions to multi-channel measurements and the more isotropic neighborhood with diagonal directions.

As a starting point, we consider the penalized weighted least squares model (PWLS) corresponding to the forward operator A, weights W and (single-channel) data f, which is, analogously to (5.14), given by

$$u^* \in \operatorname*{argmin}_u \frac{1}{2} \left\| Au - f \right\|_W^2 = \operatorname*{argmin}_u \frac{1}{2} \left\| W^{\frac{1}{2}} Au - W^{\frac{1}{2}} f \right\|^2. \tag{5.28}$$

As the next step, we duplicate the objective variable u and incorporate the differences between the two emerging variables u_1, u_2 weighted by a parameter $\mu > 0$. This will allow us to apply the CG method to an overdetermined problem and will serve as the basis for using a block-wise (more accessible) version of the Potts prior. The modified problem is given by

$$(u_1^*, u_2^*) \in \operatorname*{argmin}_{u_1, u_2} \frac{1}{2} \left\| \begin{pmatrix} W^{\frac{1}{2}}A & 0 \\ 0 & W^{\frac{1}{2}}A \\ \mu I & -\mu I \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} - \begin{pmatrix} W^{\frac{1}{2}} f \\ W^{\frac{1}{2}} f \\ 0 \end{pmatrix} \right\|^2. \tag{5.29}$$

(Recall that I denotes the identity.) We remark that we considered such a modification also in Chapter 4. The problems (5.28) and (5.29) are equivalent in the following sense: a minimizer u^* of (5.28) yields via (u^*, u^*) a minimizer of (5.29). Conversely, a minimizer (u_1^*, u_2^*) of (5.29) satisfies $u_1^* = u_2^*$ and u_1^* is a minimizer of (5.28). We prove this in a more general situation in Proposition 5.8 below. Consequently, we did not alter the original PWLS problem (5.28).

To solve the weighted least squares problems (5.29), we consider the corresponding normal equations

$$\begin{pmatrix} A^T W A u_1 + \mu^2 (u_1 - u_2) \\ A^T W A u_2 + \mu^2 (u_2 - u_1) \end{pmatrix} = \begin{pmatrix} W A^T f \\ W A^T f \end{pmatrix}. \tag{5.30}$$

We note that for CT problems these linear systems are typically overdetermined. If the basic PWLS problem (5.28) is overdetermined, the linear system (5.30) can be algorithmically approached by the CG method (as given in Algorithm 2.1) which converges to the solution of

(5.30) and thus to the solution of (5.29). In contrast to the Landweber method –applying the gradient descent method to the least squares objective– the step sizes of the CG method do not depend on the operator norm of A which is particularly advantageous for CT problems, where the operator norm of A is typically large and the Landweber method becomes unfavorable (recall also Remark 5.2 and the paragraph above Algorithm 2.1).

Algorithm 2.1 provided the pseudocode of the used variant of the CG method. We recall that Algorithm 2.1 corresponds to a variant of the CG method which is *strongly bounded perturbation resilient* (cf. Definition 2.2 and Theorem 2.6). Hence, the CG method can be perturbed by bounded perturbations w.r.t. a target function, while the termination of the algorithm remains ensured.

Perturbations by steps towards the proximal mapping of the block-wise Potts prior. Towards Potts regularized solutions, we will use the (block-wise) Potts prior as the target function. The basic scheme is given as follows.

Algorithm 5.2 (Basic Potts superiorization scheme).

1. *Choose an initial perturbation parameter* $\beta_0 > 0$ *and an annealing parameter* $0 < a < 1$, *set* $k \leftarrow 0$. *Iterate until stopping criterion:*

2a. *Perform a CG step on* $(u_1, u_2)^k$ *w.r.t. the normal equations* (5.30) *and obtain* $(u_1, u_2)^{k+1/2}$.

2b. *Perturb the iterate proportionally to* β_k *with the Potts prior:* $(u_1, u_2)^{k+1} \leftarrow \mathcal{P}_{\beta_k}(u_1, u_2)^{k+1/2}$

2c. *Update* $\beta_{k+1} \leftarrow a\beta_k$, $k \leftarrow k+1$. *Go to 2a.*

Typically, in the context of superiorization, the perturbation $\mathcal{P}_\beta$ in step 2b corresponds to adding a *non-ascending direction* w.r.t. the target function (cf. Definition 2.7) to the current iterate with appropriate step-size as in Definitions 2.2 and 2.7. Thereby, the target function values of the iterates are decreased so that the final result should become more desirable in terms of the target function. For a smooth target function, a non-ascending direction may be obtained from the negative gradient. Decreasing the parameter β in step 2c ensures that the perturbations –denoted by $\mathcal{P}_\beta$ in step 2b– become smaller in the course of the iterations and $\sum_{k=0}^{\infty} \beta_k < \infty$.

As the Potts prior is non-smooth, we resort to the evaluation of proximal mappings – a common practice in non-smooth optimization. To this end, we define the *block-wise Potts prior*

$$F(u_1, u_2) = \|\nabla_1 u_1\|_0 + \|\nabla_2 u_2\|_0. \tag{5.31}$$

The block-wise Potts prior (5.31) corresponds to the sum of the row-wise jumps in u_1 and the column-wise jumps in u_2. The proximal mapping of F for $\beta > 0$ is given by

$$\operatorname{prox}_{\beta F}(u_1, u_2) = \operatorname*{argmin}_{v_1, v_2} F(w_1, w_2) + \tfrac{1}{2\beta} \left\| (u_1, u_2) - (w_1, w_2) \right\|^2. \tag{5.32}$$

(We recall that by Theorem 3.8 the right-hand side of (5.32) is unique for almost all u_1, u_2, so that using the equality in (5.32) is reasonable.) The proximal mapping (5.32) can be evaluated efficiently. To see this, we note that the minimization in (5.32) can be performed for each block w_1, w_2 separately. These smaller problems are instances of (5.25) for $\gamma = 2\beta$, which can be solved by dynamic programming as described in Section 5.4. Hence, (5.32) can be efficiently evaluated. In this context, we note that omitting the introduction of u_1, u_2 in (5.29) would lead to a proximal mapping in a single variable only whose evaluation corresponds to solving a two-dimensional Potts problem which is NP-hard.

It is well-known that the proximal mapping corresponding to a convex lower semicontinuous function can be written in additive form (2.12) using the generalized gradient [47]. This, however, is not feasible for the block-wise Potts prior as it is not convex. The following proposition ensures that we still can use (5.32) to obtain non-ascending directions (cf. Definition 2.7) for the block-wise Potts prior. To this end, we first note the following lemma.

Lemma 5.4. *Let $J_1(w)$ denote the positions of the horizontal jumps of w and $J_2(w)$ the positions of vertical jumps of w, i.e., $\{x, x + d_s\} \in J_s(w)$ if and only if $w(x) \neq w(x + d_s)$, $s = 1, 2$. Then we have the inclusions $J_1(\bar{u}_1) \subset J_1(u_1)$ and $J_2(\bar{u}_2) \subset J_2(u_2)$ for $(\bar{u}_1, \bar{u}_2) = \operatorname{prox}_{\beta F}(u_1, u_2)$. In other words, the proximal mapping of the block-wise Potts prior does not introduce jumps which were not already present in its arguments.*

Proof. We note that the statement follows from the analogous statement for the univariate Potts problem as the proximal mapping of the block-wise Potts prior corresponds to row- and column-wise univariate Potts problems (as discussed below (5.31)). For the univariate Potts problem, the statement was proven in [206, Lemma 4.2]. □

Lemma 5.5. *For any $u = (u_1, u_2)$ and $\beta \geq 0$, the vector given by*

$$v = (v_1, v_2) := \begin{cases} \frac{\operatorname{prox}_{\beta F}(u_1,u_2) - (u_1,u_2)}{\delta} & \text{if } \operatorname{prox}_{\beta F}(u_1, u_2) \neq (u_1, u_2), \\ 0 & \text{otherwise,} \end{cases} \tag{5.33}$$

where $\delta = \|\operatorname{prox}_{\beta F}(u_1, u_2) - (u_1, u_2)\|$, satisfies $\|v\| \leq 1$ and $F\Big(u + t \cdot v\Big) \leq F(u)$ for all $t \geq 0$ w.r.t. to the block-wise Potts prior F. Thus, (5.33) *yields a non-ascending direction for the block-wise Potts prior at $u = (u_1, u_2)$.*

Proof. The proof essentially follows from Lemma 5.4: starting from $u = (u_1, u_2)$, no additional jumps will be opened in $u + t \cdot v$ for any $t \geq 0$ as v_1 is constant on the (discrete) row intervals of constant value of u_1 and v_2 on the (discrete) column intervals of constant value of u_2. Consequently, $F\Big(u + t \cdot v\Big) \leq F(u)$ for all $t \geq 0$ which completes the proof. □

A perturbation strategy for step 2b of Algorithm 5.2 in terms of adding non-ascending directions is now given by

$$\begin{aligned} \mathcal{P}_{\beta_k}\Big((u_1, u_2)^{k+1}\Big) &= \begin{cases} (u_1, u_2)^{k+1/2} + \beta_k \frac{\operatorname{prox}_{\beta_k F}\left((u_1,u_2)^{k+1/2}\right) - (u_1,u_2)^{k+1/2}}{\delta} & \text{if } \delta > 0, \\ (u_1, u_2)^{k+1/2} & \text{otherwise} \end{cases} \\ &=: (u_1, u_2)^{k+1/2} + \beta_k \cdot \Big(v_1, v_2\Big)^k, \end{aligned} \tag{5.34}$$

where $\delta = \| \operatorname{prox}_{\beta_k F}(u_1, u_2)^{k+1/2} - (u_1, u_2)^{k+1/2}\|$. The summability of the parameters β_k by step 2c of Algorithm 5.2 ensures that (5.34) produces a sequence of bounded perturbations as defined in Definition 2.2. In particular, the sequence β_k of annealing parameters satisfies $\sum_k \beta_k < \infty$ and the sequence of additive perturbations $v^k = (v_1, v_2)^k$ is bounded as $\|v^k\| \leq 1$.

We summarize the above considerations in the following theorem.

Theorem 5.6. *Algorithm 5.2 with the perturbations given by* (5.34)*, i.e., adding non-ascending directions w.r.t. the block-wise Potts prior, and the stopping criterion*

$$\|W^{\frac{1}{2}}Au_1 - W^{\frac{1}{2}}f\|^2 + \|W^{\frac{1}{2}}Au_2 - W^{\frac{1}{2}}f\|^2 + \|\mu(u_1 - u_2)\|^2 < \varepsilon \tag{5.35}$$

terminates for every $\varepsilon > \varepsilon_0$ and initializations u_1, u_2, where ε_0 is the minimal value of the underlying (weighted) least squares problem (5.28) *or, equivalently,* (5.29).

Proof. The proof follows from Theorem 2.6, Lemma 5.5 and property (ii) in Definition 2.2. □

Concerning the parameters β_k, we used the scaling $\|A^T f\|_2/\|A\|_2^2$ to adjust β_k to the scale of the data which corresponds to employing the sequence $\beta_k = \frac{\|A^T f\|_2}{\|A\|_2^2}\tilde{\beta}_k$, where $\tilde{\beta}_k$ is summable, i.e., $\sum_k \tilde{\beta}_k < \infty$, e.g., $\tilde{\beta}_k = a^k \beta_0$ for $0 < a < 1$ and $\beta_0 > 0$ as in Algorithm 5.2.

Perturbations by the proximal mapping of the block-wise Potts prior. We derived perturbations in terms of the block-wise Potts prior by taking a step towards its proximal mapping. By Lemma 5.5 this approach yields a sequence of non-ascending directions w.r.t. the block-wise Potts prior and Theorem 5.6 ensures the termination of the corresponding instance of Algorithm 5.2. As (5.34) takes a step towards the proximal mapping of the block-wise Potts prior, we may consider perturbations which take the proximal mapping itself, that is,

$$\mathcal{P}_{\beta_k}\left((u_1, u_2)^{k+1}\right) = \operatorname{prox}_{\beta_k F}\left((u_1, u_2)^{k+1/2}\right). \tag{5.36}$$

It follows immediately from optimality in the proximal mapping that the block-wise Potts value of $\mathcal{P}_{\beta_k}\left((u_1, u_2)^{k+1}\right)$ is lower or equal than the one of $(u_1, u_2)^{k+1/2}$. (This also follows from Lemma 5.5.)

In our experiments (see Figure 5.3), we observed that this perturbation strategy improves upon the perturbations given by (5.34). Furthermore, the perturbation strategy (5.36) needs not to be scaled by some constant factor to bring $(u_1, u_2)^{k+1/2}$ and its perturbation $\mathcal{P}_{\beta_k}\left((u_1, u_2)^{k+1}\right)$ to the same scale. We remark that (5.36) may also be seen as an additive perturbation by defining $(v_1, v_2)^k = \frac{1}{\beta_k}\left(\operatorname{prox}_{\beta_k F}\left((u_1, u_2)^{k+1/2}\right) - (u_1, u_2)^{k+1/2}\right)$, so that (5.36) could be written as $(u_1, u_2)^{k+1/2} + \beta_k (v_1, v_2)^k$.

Remark 5.7. It is an open question whether (5.36) yields bounded perturbations in the sense of Definition 2.2. In [47], this was proven for the special case of (smoothed) total variation. The proof uses the Lipschitz continuity of the target function. Unfortunately, neither the Potts prior nor the block-wise Potts prior are even continuous. In general, the results for perturbations obtained from evaluating proximal mappings are rather rare in the literature. However, the practical results encourage future work on this topic.

5.5.2. Potts S-CG

In view of (5.29), a large choice of the parameter μ enforces the CG steps to put more emphasis on the deviations between u_1 and u_2. Thus, large values of μ lead to results u_1, u_2 which are closer to each other. Furthermore, the block-wise Potts prior (5.31) perturbs u_1, u_2 towards row-wise and column-wise piecewise constancy, respectively. In order to obtain solutions which satisfy equality, i.e., $u_1 = u_2$, and which are in addition genuinely piecewise constant, we propose a modified version of Algorithm 5.2 inspired by penalty methods for energy minimization: after the steps 2a-2b in Algorithm 5.2 have been conducted, we increase the coupling parameter μ. By starting with a low value μ_0, the CG steps give more weight to the data fidelity in the first iterations, while in the later iterations they give more weight to the discrepancies between u_1 and u_2. Thus, the variables become closer in the later iterations until they become (approximately) equal. This modified superiorization approach is summarized in the following scheme, which we call *Potts S-CG* (Potts superiorized conjugate gradient).

Algorithm 5.3 (Basic anisotropic Potts S-CG scheme)**.**

1. *Choose an initial perturbation parameter* $\beta_0 > 0$ *and coupling parameter* $\mu_0 > 0$. *Choose an annealing parameter* $0 < a < 1$, *set* $k \leftarrow 0$. *Iterate until* u_1, u_2 *become equal:*

2a. *Perform a CG step on* $(u_1, u_2)^k$ *w.r.t. the normal equations* (5.30) *for* $\mu = \mu_k$ *and obtain* $(u_1, u_2)^{k+1/2}$.

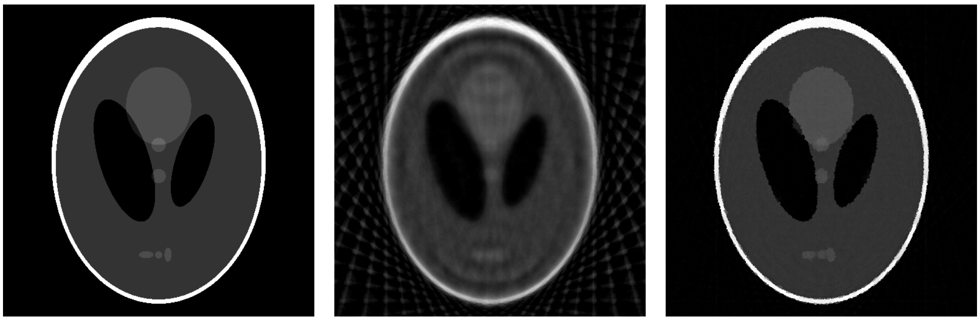
2b. *Perturb* $(u_1, u_2)^{k+1/2}$ *by the proximal mapping of the block-wise Potts prior for the current perturbation parameter* β_k, *i.e.,* $(u_1, u_2)^{k+1} \leftarrow \mathrm{prox}_{\beta_k F}(u_1, u_2)^{k+1/2}$.

2c. *Update* $\beta_{k+1} \leftarrow a\beta_k$, *increase the coupling parameter* $\mu_{k+1} \leftarrow (\mu_0\beta_0)/\beta_k$, $k \leftarrow k+1$.
Go to 2a.

In Figure 5.3, we compare the results of the perturbation strategies (5.34), (5.36) and the Potts S-CG approach for image reconstruction from undersampled noisy Radon data (measurements from 25 projection angles corrupted by Gaussian noise of variance $\sigma = 0.25$). For the perturbation strategies (5.34), (5.36), we used the coupling parameter $\mu = 10$ and the stopping rule (5.35) with $\varepsilon = m\sigma^2 + \mu^2$, where m denotes the total number of measurements. Potts S-CG used the initial coupling parameter $\mu_0 = 0.02$ and the iteration was stopped when u_1, u_2 became (approximately) equal. All three approaches used the annealing parameter $a = 0.99$ and $\beta_0 = 1$. We observe that all perturbation strategies improve upon unperturbed CG. Furthermore, we see that taking the proximal mapping (5.36) produces a more regularized solution than adding non-ascending directions (5.34). Finally, the result of Potts S-CG is genuinely piecewise constant and has the highest MSSIM value (cf. Section 2.4 for details on the MSSIM).

In the following, we give further details concerning the Potts S-CG approach. In particular, we describe the extensions for more isotropic solutions and multi-channel data.

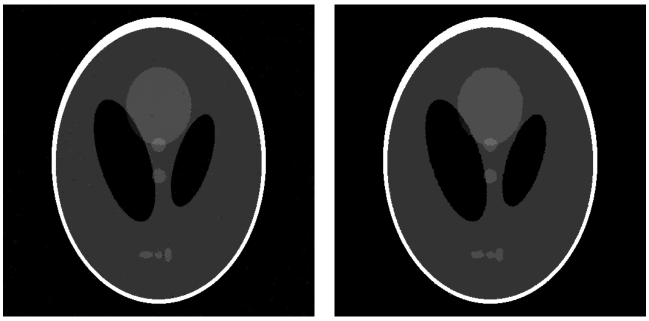
Extension to more isotropic discretizations and multi-channel measurements. We extend the Potts S-CG approach to more isotropic discretizations, i.e., directions $d_1, \ldots, d_S$, and



(a) Original

(b) Unperturbed CG, MSSIM=0.383

(c) Perturbed by non-ascending directions, μ=10, MSSIM=0.828



(d) Perturbed by proximal mappings, μ=10, MSSIM=0.965

(e) Potts S-CG, μ_0=0.02, MSSIM=0.988

Figure 5.3.: Reconstructions from undersampled Radon data (25 projection angles corrupted by Gaussian noise with $\sigma = 0.25$). (a) Original. (b) Unperturbed CG method. (c) Perturbing the CG iterations with non-ascending directions for the block-wise Potts prior improves upon the unperturbed CG method. (d) Perturbing with the proximal mapping of the block-wise Potts prior yields a further regularized result. (e) The proposed Potts S-CG method produces a less grainy and piecewise constant result and achieves the highest MSSIM value.

multi-channel measurements f. First, we note that the basic PWLS problem for multi-channel measurements is given by the channel-wise sum

$$\min_u \sum_{c=1}^{C} \frac{1}{2} \| W^{\frac{1}{2}} A u_c - W^{\frac{1}{2}} f_c \|^2. \tag{5.37}$$

As a preparation, we define the block-matrices $\overline{A}_{\mu,c}$ given by

$$\overline{A}_{\mu,c} = \begin{pmatrix} W_c^{\frac{1}{2}} A & & & & & \\ & W_c^{\frac{1}{2}} A & & & & \\ & & & \ddots & & \\ & & & & & W_c^{\frac{1}{2}} A \\ \mu I & -\mu I & 0 & \dots & 0 & 0 \\ \mu I & 0 & -\mu I & \dots & 0 & 0 \\ & \vdots & & & \vdots & \\ \mu I & 0 & 0 & \dots & & -\mu I \\ 0 & \mu I & -\mu I & \dots & 0 & 0 \\ & \vdots & & & \vdots & \\ 0 & \mu I & 0 & \dots & 0 & -\mu I \\ & & & \vdots & & \\ 0 & 0 & 0 & \dots & \mu I & -\mu I \end{pmatrix} \quad \text{and} \quad \overline{f}_c = \begin{pmatrix} W_c^{\frac{1}{2}} f_c \\ W_c^{\frac{1}{2}} f_c \\ \vdots \\ W_c^{\frac{1}{2}} f_c \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix}. \tag{5.38}$$

The upper block part of $\overline{A}_{\mu,c}$ realizes the data fidelities for the c-th channel and its lower part the mutual deviations between the variables $(u_1, \dots, u_S)$ in the c-th channel. By using this notation we may now formulate the corresponding PWLS problem, that is, the counterpart of (5.29); it is given by

$$(u_1^*, \dots, u_S^*) \in \operatorname*{argmin}_{u_1,\dots,u_S} \sum_{c=1}^{C} \frac{1}{2} \left\| \overline{A}_{\mu,c} \, (u_{1,c}, \dots, u_{S,c})^T - \overline{f}_c \right\|^2 \tag{5.39}$$

and the corresponding normal equations for each channel are given by

$$\overline{A}_{\mu,c}^T \overline{A}_{\mu,c} \, (u_{1,c}, \dots, u_{S,c})^T = \overline{A}_{\mu,c}^T \, \overline{f}_c. \tag{5.40}$$

After evaluating the multiplications in (5.40), we obtain the following more explicit equations

$$\begin{pmatrix} \sum_{t=2}^{S} \mu^2 (u_{1,c} - u_{t,c}) + A^T W_c A u_{1,c} \\ \vdots \\ \sum_{t \neq s}^{S} \mu^2 (u_{s,c} - u_{t,c}) + A^T W_c A u_{s,c} \\ \vdots \\ \sum_{t=1}^{S-1} \mu^2 (u_{S,c} - u_{t,c}) + A^T W_c A u_{S,c} \end{pmatrix} = \begin{pmatrix} A^T W_c f_c \\ \vdots \\ A^T W_c f_c \\ \vdots \\ A^T W_c f_c \end{pmatrix} \tag{5.41}$$

for each channel $c = 1, \dots, C$. The next proposition establishes the relation between the basic weighted least squares problem (5.37) and its augmented counterpart (5.39).

Algorithm 5.4: CG step for the augmented weighted least squares problem (5.39)

```
Input: Forward operator $A$, multi-spectral sinogram $f$, weights $W_c$, coupling parameter $\mu$,
       current iterates $u, p, h$
Output: Updated iterates $u, p, h$

for $c = 1, \dots, C$ do
    $g_{s,c} \leftarrow A^T W_c A u_{s,c} - A^T W_c f_c + \sum_{t \neq s} \mu^2 (u_{s,c} - u_{t,c})$   for all $s = 1, \dots, S$,
    $\alpha = \frac{\sum_s (g_{s,c})^T h_{s,c}}{\sum_s (p_{s,c})^T h_{s,c}}$,
    $p_{s,c} \leftarrow -g_{s,c} + \alpha p_{s,c}$   for all $s = 1, \dots, S$,
    $h_{s,c} \leftarrow A^T W_c A p_{s,c} + \sum_{t \neq s} \mu^2 \, (p_{s,c} - p_{t,c})$   for all $s = 1, \dots, S$,
    $\kappa = -\frac{\sum_s (g_{s,c})^T p_{s,c}}{\sum_s (h_{s,c})^T p_{s,c}}$,
    $u_{s,c} \leftarrow u_{s,c} + \kappa p_{s,c}$   for all $s = 1, \dots, S$
end
```

Proposition 5.8. *(i) A minimizer u^* of the least squares problem* (5.37) *induces a minimizer of the augmented least squares problem* (5.39) *via $(u^*, \dots, u^*)$. (ii) A minimizer $(u_1^*, \dots, u_S^*)$ of the augmented least squares problem* (5.39) *satisfies $u_1^* = \dots = u_S^*$. Further, u_1^* is a minimizer of* (5.37) *as well. (iii) If A has full column rank, then u^* is the (unique) minimizer of* (5.37) *if and only if $(u^*, \dots, u^*)$ is the (unique) minimizer of* (5.39).

Proof. (i) Let u^* be a minimizer of (5.37) and we define $(v_1, \dots, v_S) = (u^*, \dots, u^*)$. We show that $(v_1, \dots, v_S)$ satisfies the normal equations (5.41) which is sufficient for a minimizer of (5.39). It holds by definition that $v_s = v_t$ for all s, t. As u^* satisfies the normal equations of (5.37), we further obtain $A^T W A v_{s,c} = A^T W f_c$ for all channels $c = 1, \dots, C$. Together, $(v_1, \dots, v_S)$ satisfies (5.41) for all channels $c = 1, \dots, C$. (ii) We show that the equality of the block variables u_s^* of a minimizer follows from optimality. To this end, let $(u_1^*, \dots, u_S^*)$ be a minimizer of (5.39). Towards a contradiction, we assume that there are $s \neq t$ such that $u_s^* \neq u_t^*$ which in particular means $\sum_{s=1}^S \sum_{t=s+1}^S \frac{1}{2} \|u_{s,c}^* - u_{t,c}^*\| > 0$ for at least one channel c. For $(v_1, \dots, v_S) := (u^*, \dots, u^*)$ as above we have that $\sum_{s=1}^S \frac{1}{2} \|W^{\frac{1}{2}} A v_c^* - W^{\frac{1}{2}} f_c\|^2$ is minimal for each c in view of u^* being a minimizer of (5.37). Further, by definition the quadratic deviations between the block variables vanish, i.e., $\sum_{s=1}^S \sum_{t=s+1}^S \frac{1}{2} \|v_{s,c} - v_{t,c}\|^2 = 0$ for all c. As the sum of these two terms corresponds to the objective in (5.39), $(v_1, \dots, v_S)$ yields a lower objective value than $(u_1^*, \dots, u_S^*)$ which is a contradiction, so $u_1^* = \dots = u_S^*$. From the normal equations (5.40) follows immediately $A^T W A u_{1,c}^* = A^T W f_c$ for all c which corresponds to the normal equations of (5.37). (iii) The third assertion follows from (i) and (ii) together with the uniqueness of least squares problems for full-rank matrices. □

By following the lines of the CG algorithm (cf. Algorithm 2.1), a CG step on (5.39) is given by Algorithm 5.4.

Next, we derive the perturbation step by means of the proximal mapping of the block-wise Potts prior with S directions. We defined the block-wise Potts prior for the special case $S = 2$

Algorithm 5.5: Multi-channel Potts S-CG

```
Input: Forward operator $A \in \mathbb{R}^{m\times n^2}$, multi-spectral sinogram $f \in \mathbb{R}^{m\times c}$, PWLS weights
       $W \in \mathbb{R}^{m\times m}$, annealing parameter $0 < a < 1$, initial parameters $\beta_0, \mu_0 > 0$, stopping
       parameter tol $> 0$
Output: $u^* \in \mathbb{R}^{n\times n\times C}$
Initialize for all $s = 1, \dots, S$:
$u_s^0 = A^T W f,\ p_s^0 = A^T W f,\ \ h_s^0 = A^T W A p_s^0 + \sum_{t\neq s} \mu_0^2 (p_s^0 - p_t^0)$
$k \leftarrow 0$
repeat
    /* Increase the splitting penalty parameter proportionally to $\beta_k$ */
    $\mu_k \leftarrow (\mu_0\beta_0)/\beta_k$
    /* Compute solutions of linewise Potts problems with jump penalty $2\beta_k$: */
    $(u_1, \dots, u_S)^{k+1/2} = \operatorname{argmin}_{v_1,\dots,v_S} \sum_s \|u_s^k - v_s\|^2 + 2\beta_k \omega_s \|\nabla_{d_s} v_s\|_0$
    /* Peform a CG step on $(u_1, \dots, u_S)^{k+1/2}$ w.r.t. the weighted normal equations (5.41) : */
    for $c = 1, \dots, C$ do
        Compute $u_{s,c}^{k+1}, h_{s,c}^{k+1}, p_{s,c}^{k+1}$ from $u_{s,c}^{k+1/2}, h_{s,c}^k, p_{s,c}^k$ for all $s = 1, \dots, S$ by
          Algorithm 5.4 for $\mu = \mu_k$
    end
    $\beta_{k+1} \leftarrow a\beta_k$
    $k \leftarrow k + 1$
until $\|u_s^k - u_{s+1}^k\|_\infty <$ tol for all $s = 1, \dots, S - 1$;
return $u^* = \frac{1}{S} \sum_{s=1}^S u_s^k$
```

in (5.31). For the more general situation with S directions as above, it is given by

$$F(u_1, \dots, u_S) = \sum_{s=1}^{S} \omega_s \|\nabla_{d_s} u_s\|_0. \tag{5.42}$$

(Recall the definition of $\|\nabla_{d_s} u_s\|_0$ in (5.17) and the weights ω_s from Section 4.2.1.) To perturb the CG iterates of (5.41) by means of the block-wise Potts prior (5.42), we need to evaluate its proximal mapping as above. The proximal mapping of the block-wise Potts prior for S directions (5.42) is defined by

$$\operatorname{prox}_{\beta F}(u_1, \dots, u_S) = \operatorname*{argmin}_{w_1,\dots,w_S} F(w_1, \dots, w_S) + \frac{1}{2\beta} \sum_{s=1}^{S} \|u_s - w_s\|^2. \tag{5.43}$$

As seen in Section 5.4, the minimization in (5.43) decomposes into problems, which depend on one of the u_s only. These smaller problems are of the form (5.26), which can be efficiently solved with the solver developed in Chapter 3. In Algorithm 5.5, we provide the pseudocode of the Potts S-CG algorithm for more isotropic discretizations and multi-channel data.

5.6. Comparison of Potts ADMM and Potts S-CG

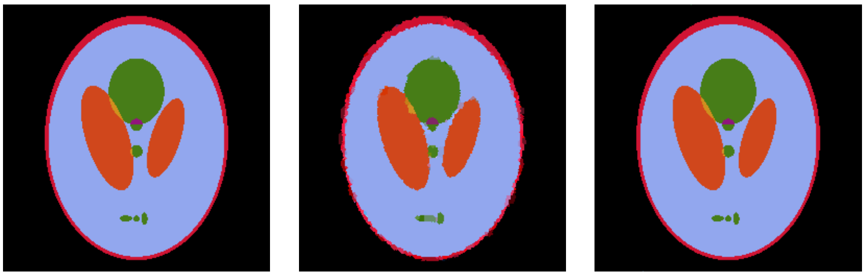
Handling the forward operator. Potts ADMM introduces an additional splitting variable v in (5.19) to cope with the forward operator A. Consequently, one has to solve a Tikhonov problem (5.21) w.r.t. A and v in each iteration of Potts ADMM. In contrast, Potts S-CG does not introduce such a splitting variable and only requires a CG step w.r.t. A in each iteration, i.e., it requires only applying A and A^T in each iteration instead of solving a Tikhonov problem. As a result, Potts S-CG needs more iterations, while the iterations are cheaper than in Potts ADMM (Figure 5.4: Potts S-CG 0.95 sec/iteration, Potts ADMM 2.35 sec/iteration).

Reconstruction from undersampled multi-channel Radon data. Next, we consider the reconstruction of a colored, i.e., three-channel, Shepp-Logan Phantom from undersampled multi-channel Radon data corrupted by additive Gaussian noise (20 angles, $\sigma = 0.35$). Further, we study the data deviation, $\frac{1}{S}\sum_{s,c}\|Au^k_{s,c}-f_c\|^2$, and the block-wise Potts prior, $\sum_s \omega_s\|\nabla_{d_s}u^k_s\|_0$, as a function of the iteration index. To put the results into context, we also include the reconstruction result and iteration data of a penalty method for the Potts model, which relaxes the constrained problem (5.19) by replacing the constraints by a sequence of softened constraints. To fix ideas, this method corresponds to Potts ADMM with the Lagrange multipliers set to zero. Furthermore, we include the Potts superiorized Landweber method (Potts S-Landweber), which corresponds to Potts S-CG with Landweber steps instead of CG steps. (Recall that a Landweber step corresponds to a gradient descent step w.r.t. the (weighted) least squares problem (5.37); see also Example 2.1.)

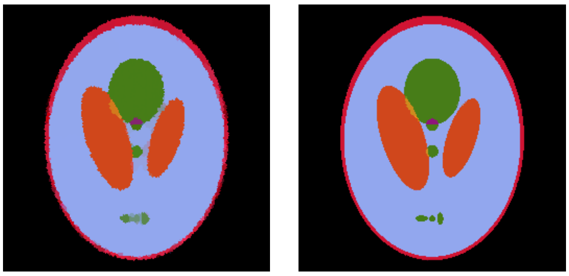
In Figure 5.4, we show the results. First, we note that the jumps of all results are spatially aligned across the channels which illustrates the considerations on the multi-channel Potts prior in Section 5.3. Concerning the reconstruction quality, we observe that the penalty method as well as Potts S-Landweber produce spurious artifacts near segment boundaries. Potts ADMM and Potts S-CG yield improved reconstruction results which are close to the groundtruth. This confirms the theoretical considerations of Section 5.5.1 that the Landweber method is not a favorable basic algorithm in the sense of superiorization, when large operator norms are involved. Moreover, the results confirm that the Lagrange multipliers in Potts ADMM (the "memory" of the iterations) play a crucial role for the reconstruction quality. These qualitative findings are reflected by the values of the block-wise Potts prior and the data deviations over the iterations: Potts ADMM and Potts S-CG arrive at values which are close to the values of the groundtruth. This illustrates that the results of Potts ADMM and Potts S-CG agree with the data and at the same time minimize the (block-wise) Potts prior.

5.7. Experimental results

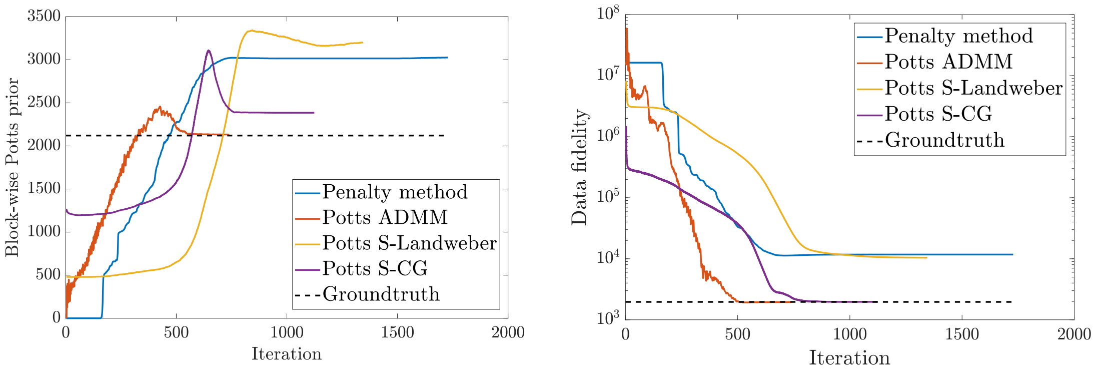
In this section, we illustrate the potential of the (multi-channel) Potts prior for multi-spectral computed tomography. More precisely, we consider the Potts S-CG algorithm (Algorithm 5.5) and the Potts ADMM algorithm (Algorithm 5.1) and compare them with existing TV-type approaches. First, we provide the necessary implementation details of Potts ADMM and Potts S-CG. Next, we briefly recall the TV, TNV and dTVp prior. To obtain a qualitative comparison,



(a) Original 256 × 256 (b) Penalty method (c) Potts ADMM

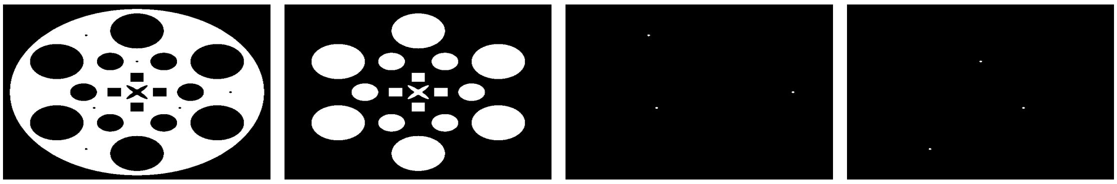


(d) Potts S-Landweber (e) Potts S-CG



(f) Block-wise Potts prior (g) Data fidelity

Figure 5.4.: Potts ADMM and Potts S-CG for undersampled noisy multi-channel Radon data (20 angles, $\sigma = 0.35$). *Top*: the jumps of all results are spatially aligned across the channels. However, the penalty method and Potts S-Landweber produce some spurious artifacts near the segment boundaries. Potts ADMM and Potts S-CG produce improved reconstruction results which are close to the groundtruth. On close inspection, we observe that some of the segment boundaries are briefly smoother in the Potts ADMM result than in the Potts S-CG result (see, e.g., the upper boundary of the right ellipse). *Bottom*: the values of the block-wise Potts prior and the data deviations over the iterations reflect the qualitative findings as Potts ADMM and Potts S-CG achieve values which are near the values of the groundtruth. This shows that the solutions of Potts ADMM and Potts S-CG simultaneously agree with the data and minimize the (block-wise) Potts prior.



(a) Quartz (b) Pyrite (c) Galena (d) Gold

Figure 5.5.: The geocore phantom consists of four inorganic materials: (a) quartz, (b) pyrite, (c) galena and (d) gold. The background is air.

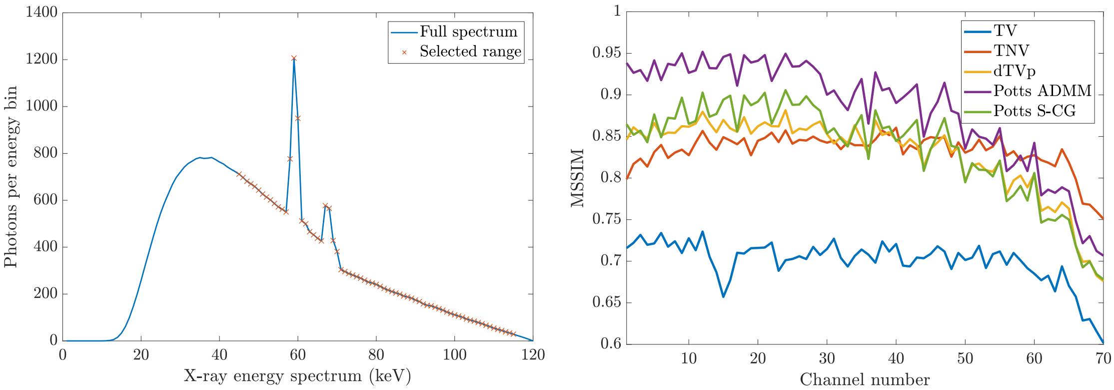


Figure 5.6.: *Left:* the full X-ray spectrum and the range of energies (45-114 keV) used for modeling the measurements of the geocore phantom. The spectrum is discretized into 70 energy bins. *Right:* MSSIM values of the reconstructions of the geocore phantom for all energy bins. The channel-coupling methods achieve larger MSSIM values than channel-wise TV. For the majority of the energy-bins, the result of Potts ADMM and Potts S-CG have higher MSSIM values than the TV-type regularizers dTVp and TNV (Potts ADMM: 57 channels, Potts S-CG: 38 channels).

we then consider the geocore phantom used in [114]. Finally, we give a quantitative comparison of the methods for a phantom which consists of few organic materials.

Implementation details. We provide the necessary implementation details. We begin with Potts ADMM (Algorithm 5.1). As coupling sequences, we employ $\rho_k = 10^{-7} \cdot k^{2.01}$, $\mu_k = \rho_k/S$. For the stopping criterion, we set `tol` $= 10^{-5}$. We solve the Tikhonov subproblems (5.21) using Matlab's built-in function `pcg` which is applied to the weighted normal equations (5.24) with standard tolerance 10^{-6} and a maximum number of 2000 iterations. We use the previous iteration v^{k-1} as initial guess, so that after the first ADMM iterations only few `pcg` iterations are needed. Regarding Potts S-CG (Algorithm 5.5), we used the annealing parameter $a = 0.999$ and the initial coupling $\mu_0 = 10^{-4}$. For the stopping criterion, we set `tol` $= 10^{-5}$.

Methods for comparison. We compare the Potts ADMM and the Potts S-CG method with classical channel-wise total variation (TV), channel-coupling total nuclear variation (TNV)

[168] and the probabilistic directional TV method (dTVp), which was proposed in [114].

Channel-wise TV corresponds to the ℓ_1-norm of the discrete gradient $\|\nabla u\|_1$ for each channel separately. Its application corresponds to solving the convex problem, for a parameter $\alpha > 0$, given by

$$\operatorname*{argmin}_u \sum_{c=1}^{C} \|Au_c - f_c\|_W + \alpha \|\nabla u_c\|_1. \tag{5.44}$$

The TNV regularizer uses the nuclear norm of the discrete Jacobian in each pixel of a multi-channel image to correlate the channels. Applying TNV corresponds to the convex problem

$$\operatorname*{argmin}_u \sum_{c=1}^{C} \|Au_c - f_c\|_W + \sum_{x \in \Omega} \alpha \|Du(x)\|_*, \tag{5.45}$$

where $\alpha > 0$ is a parameter, $Du(x)$ is the $2 \times C$ matrix of channel-wise finite differences of u at the point x and $\|\cdot\|_*$ denotes its nuclear norm, i.e., the sum of its singular vales.

The dTVp regularizer is a modification of the directional TV (dTV) regularizer. The dTV regularizer enforces correlation between a channel image w and a reference channel v by means of the directional diffusion of the channel given the known reference channel. To fix ideas, we denote by $\operatorname{vec}(w) \in \mathbb{R}^{n^2}$ the column-wise vectorization of w and by D_x the $2 \times n^2$-matrix such that $D_x \operatorname{vec}(w)$ is the finite differences vector of w at the point x. Then, dTV is given by

$$\operatorname{dTV}(w, v) = \sum_{x \in \Omega'} \|P_v D_x \operatorname{vec}(w)\|_2, \quad P_v = \begin{cases} I - \frac{D_x \operatorname{vec}(v) \operatorname{vec}(v)^T D_x^T}{\|D_x \operatorname{vec}(z)\|^2} & \text{if } D_x \operatorname{vec}(v) \neq 0, \\ I & \text{if } D_x \operatorname{vec}(v) = 0 \end{cases} \tag{5.46}$$

for a single-channel image w and reference v. The multi-channel version of (5.46) is given by

$$\sum_{c=1}^{C} \operatorname{dTV}(u_c, v_c), \tag{5.47}$$

where v_c denote the reference channel for u_c. In contrast to dTV, for dTVp the reference channels are not fixed, but chosen probabilistically in each iteration of the reconstruction process. The reference channels are chosen among the channels of the former iterate of u. This process uses a probability mass function, which is based on the channel-wise geometric mean of the estimated signal-to-noise ratios (SNR). Hence, channels with high SNR are more likely to be selected as a reference channel. We note that the dTVp regularizer is non-convex [114].

We used the implementation of [114], which employs the FISTA method (fast iterative shrinkage thresholding) [11] to approach the respective minimization problems. Thus, the FISTA method converges to the global minimizers of (5.45) and (5.44) as TV and TNV are convex. However, this does not necessarily hold for dTVp which is non-convex.

Geocore phantom. We start out with a qualitative comparison w.r.t. the geocore phantom of [114]. The phantom has size 512 × 512 and consists of four materials: quartz, pyrite, galena, and gold; see Figure 5.5. The width of the domain is set ro 1 cm, the distance from the X-ray source to the rotation center to 3 cm, the distance from the source to the detector is 5 cm and the width of the detector array is 2 cm. The X-ray spectrum was simulated with the spektr software package [173]. The energy spectrum of the emitted photons is shown on the left-hand side of Figure 5.6 for the used tube potential (120 kVp) and photon flux $I_0 = 4 \cdot 10^4$. The photon attenuation coefficients for the materials were obtained from the PhotonAttenuation software package [188]. The energy range is chosen as 45-114 keV and is discretized into 70 energy bins. The multi-spectral measurements were simulated with the Astra-Toolbox [189]. In particular, we set the fan-beam scanning geometry and obtained the measurements according to 120 projection angles from a larger version of the phantom (1024 × 1024). For TV, TNV and dTVp, we used the model parameters as given in [114], i.e., $\alpha_{TV} = 4.2 \cdot 10^{-4}$, $\alpha_{TNV} = 1.2 \cdot 10^{-3}$ and $\alpha_{dTVp} = 3.3 \cdot 10^{-3}$. Potts ADMM used the jump penalty $\gamma = 0.075$ and Potts S-CG the initial perturbation parameter $\beta_0 = 500$.

In Figure 5.7, we show the reconstructions for three depicted channels. We observe that Potts ADMM and Potts S-CG produce sharp boundaries across the channels, while also showing considerably fewer artifacts than the methods of comparison. These qualitative findings are confirmed by the corresponding MSSIM values (see also the right-hand side of Figure 5.6), which are higher for Potts ADMM and Potts S-CG for the majority of the channels.

Organic spheres phantom. In the last paragraph, we considered the geocore phantom, which consisted of inorganic materials, many energy channels and a moderate number of projections were available. Next, we conduct a quantitative comparison in terms of a phantom, which consists of organic materials and only few energy bins and projection angles are available. To this end, we consider the 256 × 256 phantom of Figure 5.8, which consists of homogeneous regions filled with muscle, fat and bone tissue. The domain and detector width as well as the distances involved in the measurement process are the same as for the geocore phantom. The X-ray spectrum as well as the LAC's of muscle, fat and bone were simulated with the spektr software package for the used tube potential (150 kVp) and photon flux $I_0 = 10^5$. In this context, we note that the measurements were simulated according to (5.8) to avoid the inverse crime. Then, for the reconstruction, the linear model (5.9) was used. We show the LAC's of the involved materials and the X-ray spectrum in Figure 5.9. The detectors bin the X-ray spectrum into three energy bins. We note that the LAC's of fat and muscle tissue are very close, especially for higher energies. Thus, the reconstruction approaches should aid the recovery of the third channel (the one which holds highest energies among the three channels) by employing the other two (less problematic) channels. Again, we simulate the measurements with the Astra-toolbox and use a fan-beam scanning geometry with 25 projection angles (obtained from a larger 512 × 512 version of the phantom). Hence, the reconstruction approaches have to deal with the additional challenge of highly undersampled measurements.

The individual model parameters were chosen empirically. More precisely, we determined the model parameters by choosing the optimal ones w.r.t. to the mean value of the channel-wise MSSIM's. In particular, for TV and TNV, the optimal parameter α was chosen among $\{10^{-4}, 1.5 \cdot 10^{-4}, \dots, 10^{-3}\}$. As dTVp is of probabilistic nature, we repeated the computation for each parameter α five times and chose the best result. Here, we let $\alpha \in \{10^{-4}, 2 \cdot 10^{-4}, \dots, 2 \cdot 10^{-3}\}$. The jump penalty of Potts ADMM was chosen from $\gamma \in \{10^{-5}, 2 \cdot 10^{-5}, \dots, 2 \cdot 10^{-4}\}$ and for Potts S-CG, the initial perturbation parameter was obtained from $\beta_0 \in \{1, \dots, 20\}$.

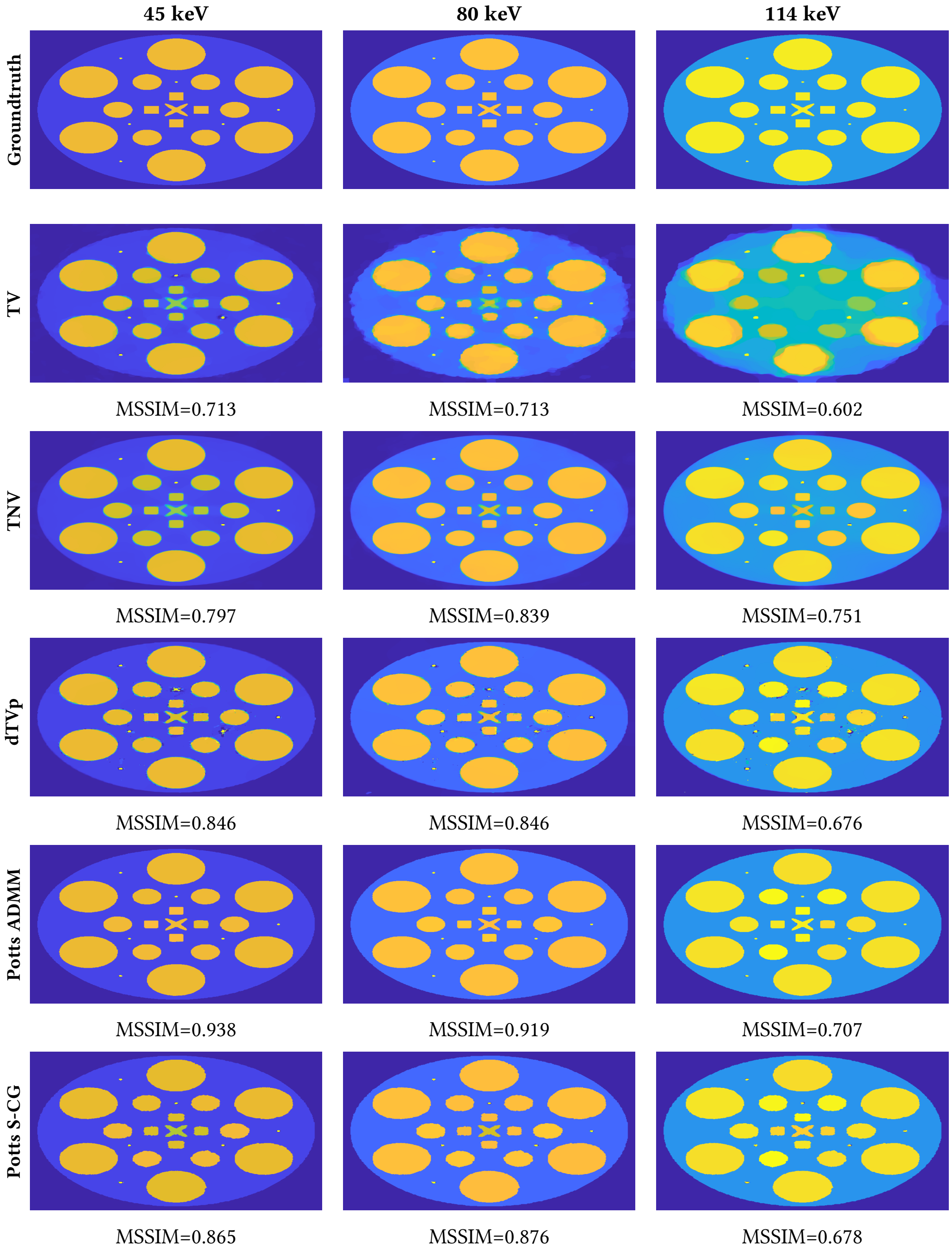
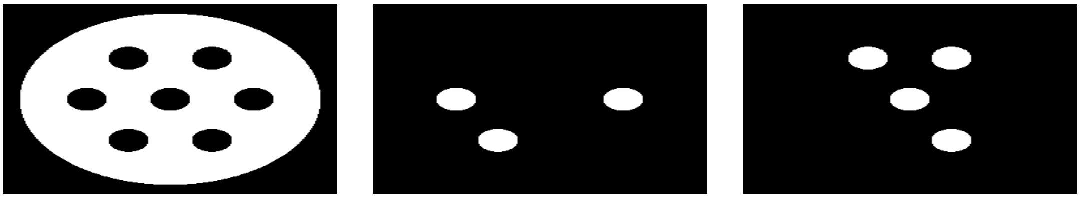


Figure 5.7.: Reconstruction results of the geocore phantom for three selected energy bins. Channel-wise TV exhibits artifacts at the segment boundaries in the second channel and does not recover the inner segments in the third channel. TNV yields an improved reconstruction, but still shows (blueish) artifacts at the segment boundaries. The dTVp result has sharper boundaries than TV and TNV, but the artifacts at the segment boundaries are still present. The Potts ADMM and the Potts S-CG results show less artifacts and the boundaries are sharp. Further, they achieve the highest MSSIM values for two out of the three depicted channels, which is representative for the full energy spectrum; see Figure 5.6.



(a) Muscle tissue

(b) Fat tissue

(c) Bone tissue

Figure 5.8.: The organic spheres phantom consists of three materials. (a) Muscle tissue, (b) fat tissue and (c) bone tissue. The background is air. The main challenge is to separate the regions with muscle tissue and those with fat tissue as fat and muscle have very similar LAC's throughout the considered energy spectrum; see Figure 5.9.

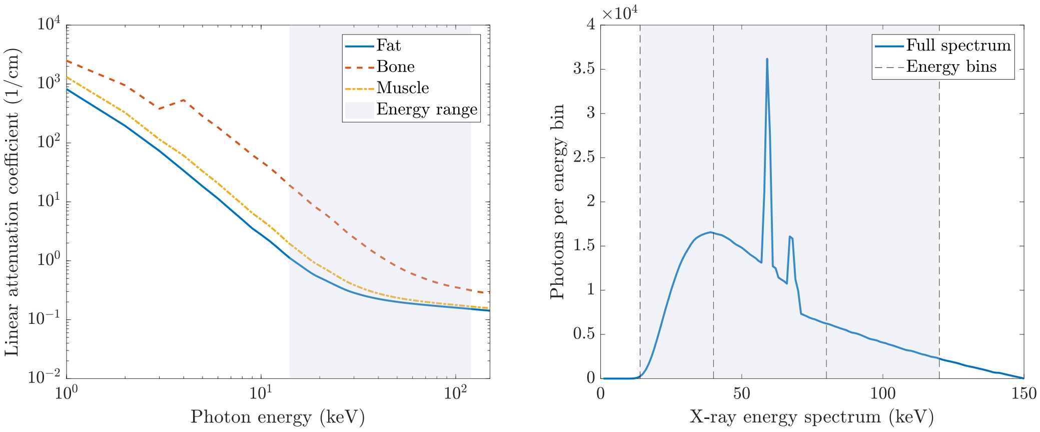


Figure 5.9.: Modeling the measurements of the organic spheres phantom. *Left:* the LAC's of fat, bone and muscle tissue. The highlighted area depicts the used energy range. The LAC curves of fat and muscle are close. *Right:* the full X-ray spectrum and the used energy range (15-120 keV) for modeling the data of the organic spheres phantom. The spectrum is binned into three energy bins by the detectors and their boundaries are depicted by the dashed vertical lines.

In Figure 5.10, we show the reconstructions together with the corresponding (optimal) MSSIM values. We observe that Potts ADMM and Potts S-CG achieve the highest MSSIM values for all three channels and the MSSIM values of Potts ADMM and Potts S-CG are rather close. Furthermore, the fact that Potts ADMM and Potts S-CG achieve higher MSSIM values than the other methods is reflected by the reconstructions: the result of channel-wise TV exhibits blurry boundaries and the segments corresponding to the fat tissue are not recovered in the third channel. TNV provides an improved reconstruction of the third channel. However, the segment boundaries remain diffuse. The dTVp method produces sharp boundaries in all three channels, but introduces some spurious artifacts near the boundaries. Potts ADMM and Potts S-CG recover the segments and provide sharp boundaries throughout the channels. Further, they show fewer artifacts.

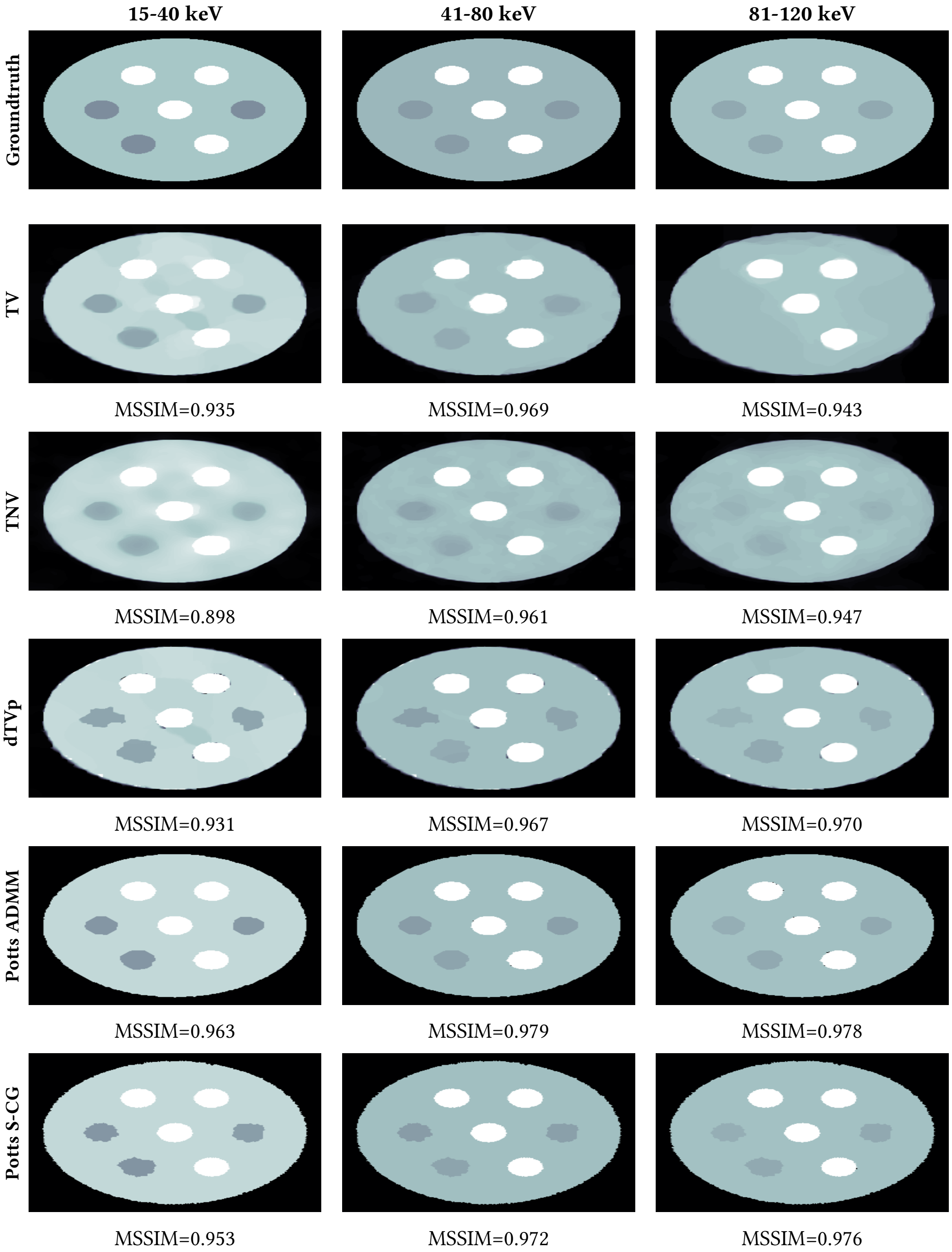


Figure 5.10.: Reconstructions of the organic spheres phantom. The channel-wise TV result exhibit blurry boundaries and smooths out the fat segments in the third channel. TNV yields an improved reconstruction for the third channel. However, the boundaries are still blurry. The dTVp method produces sharp boundaries in all three channels, but introduces some spurious artifacts near the boundaries. Potts ADMM and Potts S-CG recover the segments and provide sharp boundaries throughout the channels. Further, they show fewer artifacts and achieve the highest MSSIM values for all three channels.

5.8. Summary of the chapter

In this chapter, we considered the multi-channel reconstruction problem in multi-spectral CT. In particular, we focused on measurements obtained by energy-discriminating photon-counting detectors. We started out by briefly describing the corresponding linear measurement model. We noted that the edges in the different energy channels are typically spatially correlated. Hence, an appropriate regularizer incorporates this prior knowledge in the reconstruction process, i.e., such a regularizer should couple the channels. Furthermore, we found that the multi-channel Potts prior provides a strong channel coupling in the sense that the jumps of the results are enforced to be at the same spatial positions across the channels. This property is especially beneficial for the multi-channel images in multi-spectral CT as their channels have a strong structural correlation. We employ the multi-channel Potts prior by minimizing the multi-spectral Potts model. To this end, we adapted the ADMM strategy proposed in [183] to the multi-channel reconstruction problem in multi-spectral CT. Furthermore, we proposed new reconstruction approaches based on Potts superiorization of the conjugate gradients method (CG). Here, the iterates of the CG method were perturbed w.r.t. the (block-wise) Potts prior towards more desirable solutions. We have shown that one obtains non-ascending directions w.r.t. the block-wise Potts prior by taking steps towards its proximal mapping. The corresponding superiorization approach, which perturbs the iterates by adding these non-ascending directions, turned out to yield improved results compared to the non-perturbed CG. Furthermore, we showed that the theoretical termination of the resulting algorithm is ensured. In practice, we observed further improved results when we perturbed the iterates with the proximal mapping itself. Based on this observation, we developed a new Potts superiorization approach, which we call Potts S-CG. Potts S-CG uses the proximal mapping as before and additionally lets the underlying PWLS problem evolve in the course of the iterations, so that the final result becomes genuinely piecewise constant.

The energy minimization approach (Potts ADMM) and the superiorization approach (Potts S-CG) both produce solutions which are (multi-channel) Potts-regularized. We identified Potts ADMM and Potts S-CG as suitable choices within their respective class of methods by comparing them to a penalty method and a method which Potts-superiorizes the Landweber iteration, respectively. Despite the different abstract interpretations, the iterations in Potts ADMM as well as Potts S-CG involve a data step and a regularizing step. The latter decomposes into univariate Potts problems (solved efficiently by the method of Chapter 3). A significant difference between Potts ADMM and Potts S-CG is that the data step of Potts ADMM corresponds to solving a Tikhonov problem, while the data step of Potts S-CG corresponds to one CG step.

In numerical experiments, we applied Potts ADMM and Potts S-CG to simulated multi-spectral CT data and compared them to the existing TV-based methods. The benefits of these methods based on the Potts prior manifested themselves in sharper edges and mostly higher MSSIM values than the TV-type methods. These benefits may be attributed to the channel-coupling by the multi-channel Potts prior.

6. Smoothing Images with the Second Order Mumford-Shah Model

In this chapter, we consider second and higher order Mumford-Shah models for images. This chapter is based on the publication [120].

Organization of the chapter. In Section 6.1, we give an overview and discuss related work. In Section 6.2, we reformulate second and higher order Mumford-Shah models in terms of Taylor jets and derive corresponding discrete formulations. We start with the second order anisotropic case, and proceed with discussing on how to increase the isotropy and how to deal with the general higher order case. We further consider an approach based on distances between Taylor jets and discuss its relation to the previously proposed approach. In particular, we point out a particular instance which is motivated by Hermite interpolation. In Section 6.3, we develop a strategy for minimizing the energy functional. In Section 6.4, we apply the proposed approach to the second order Mumford-Shah model to numerical data and compare it with existing numerical approaches. Furthermore, we use the proposed method in connection with edge detection. Finally, Section 6.5 concludes the chapter with a summary.

6.1. Overview and related work

Mumford-Shah models are established variational tools for edge preserving smoothing of images. The models produce a piecewise smoothed image together with the corresponding edges. The classical first order Mumford-Shah model (cf. (1.1)) [148, 149] is given by the minimization problem

$$\operatorname*{argmin}_{u,\Gamma} \int_{\Omega} |u(x) - f(x)|^2 \, dx + \beta^2 \int_{\Omega \setminus \Gamma} \|\nabla u(x)\|^2 \, dx + \gamma \, \mathrm{length}(\Gamma). \tag{6.1}$$

We recall that the arguments of the functional are a (closed) set Γ which represents the edge set and a function u which is smooth on $\Omega \subset \mathbb{R}^2$ outside the edge set Γ, i.e., u is continuously differentiable on $\Omega \setminus \Gamma$. For more details we refer to (1.1) and below in Chapter 1.

Already in Chapter 3, we saw that first order (univariate) Mumford-Shah models exhibit the "gradient-limit effect" (recall Figure 3.2) [21]. It describes the phenomenon that spurious edges are introduced to account for steep slopes in the signal/image. The gradient-limit effect stems from penalizing the (squared) norm of the gradient of u in (6.1), i.e., the model penalizes deviations from constant functions as their gradients are zero. In Chapter 3, we considered higher order Mumford-Shah models to address the gradient-limit effect in the univariate situation. In this chapter, we consider higher Mumford-Shah models *for images* with special emphasis on the *second order Mumford-Shah model*.

We start by formulating the second order Mumford-Shah model. To this end, we replace the gradient in (6.1) by the Hessian matrix as proposed in [21] and obtain the second order Mumford-Shah model

$$\operatorname*{argmin}_{u,\Gamma} \int_\Omega |u(x) - f(x)|^2 \, dx + \beta^2 \int_{\Omega \setminus \Gamma} \|\nabla^2 u(x)\|_F^2 \, dx + \gamma \operatorname{length}(\Gamma), \tag{6.2}$$

where the second order penalty is now the squared Frobenius norm of the Hessian matrix $\nabla^2 u$ of u which is given by $\|\nabla^2 u(x)\|_F^2 = \partial_{11} u(x)^2 + 2\partial_{12} u(x)^2 + \partial_{22} u(x)^2$. The arguments of the second order Mumford-Shah functional are the closed (edge) set Γ as well as a function u which is twice continuously differentiable on Ω outside the edge set Γ. The remaining symbols have the same role as in the first order Mumford-Shah model. Analogously to the univariate case, the second order Mumford-Shah model (6.2) does not penalize linear trends in u. In Figure 6.1, we illustrate how the second order Mumford-Shah model can prevent the gradient-limit effect by recovering steep color gradients.

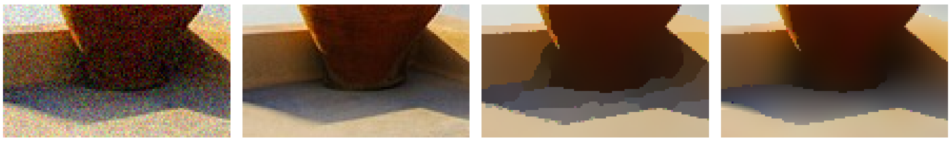
We note that the squared L^2-distance in (6.2) which promotes closeness to data f may be replaced by a general L^p-based distance or a more general pseudo-metric to account for different noise types. Further, one may replace the Frobenius norm of the Hessian by another norm. For instance, one could employ weighted ℓ_2-type norms $\|\nabla^2 u(x)\|_w^2 = w_{11}\partial_{11} u(x)^2 + 2w_{12}\partial_{12} u(x)^2 + w_{22}\partial_{22} u(x)^2$ with positive weights w_{11}, w_{12}, w_{22}. Furthermore, Blake and Zisserman [21] also considered the Laplacian of u instead of the (weighted) Frobenius norm of the Hessian of u. However, they report on undesired effects when using this modification. A natural choice corresponds to the norm of $\nabla^2 u(x)$ as a bilinear form. In view of the difficulty of computing operator norms, (weighted) Frobenius norms of the Hessian are a computable compromise. (Further, also more general Schatten norms, in particular, trace norms, have been considered in the context of regularizing ill-posed problems (using different regularizers); see for instance [133].)

Second derivatives add an additional level of complexity. Thus, the second order models are at least as demanding as the classical first order Mumford-Shah models, both on the theoretical and the algorithmic side. On the algorithmic side, there are far fewer approaches to the second order Mumford-Shah model than for the first order Mumford-Shah model. The graduated non-convexity approach by Blake and Zisserman, cf. [21], and Ambrosio-Tortorelli type approaches [2, 216, 215] are the only approaches the author is aware of. Hence, the main goal of this chapter is to derive algorithms for second order Mumford-Shah models. Work on theoretical aspects is discussed in the related work section below.

In this chapter, we develop an algorithmic framework for second order Mumford-Shah models based on reformulations of second (and higher) order Mumford-Shah models in terms of Taylor jets. After discretization, these reformulations allow to model a local Taylor expansion in each pixel, in contrast to an image value only. We propose an algorithmic approach to the corresponding minimization problems which uses an ADMM splitting such that all subproblems can be solved exactly and non-iteratively These subproblems correspond to univariate segmented jet problems for which we derive fast solvers which use the dynamic programming approach of Chapter 3 as a basis. To show the potential of the proposed method, we apply it to numerical data. In particular, we perform a qualitative and quantitative comparison of our

(a) Noisy data (b) Clean image (groundtruth)

(c) First order Mumford-Shah model (d) Second order Mumford-Shah model (with the proposed method)



(e) Detail of (a) (f) Detail of (b) (g) Detail of (c) (h) Detail of (d)

Figure 6.1.: Qualitative comparison of first and second order Mumford-Shah models. The classical first order Mumford-Shah model produces extra spurious edges to amount for steep slopes which is called the “gradient-limit effect” (see, e.g., the rooftop or the floor next to the amphora). The second order Mumford-Shah model recovers steep color gradients and yields an overall smoother image. (The respective model parameters were chosen optimally w.r.t. the MSSIM).

algorithmic approach with the methods for the second order Mumford-Shah problem existing in the literature.

Related work. The model (6.2) was proposed by Blake and Zisserman [21]. In particular, they proposed a more general model which incorporates an additional penalty for the kink set of the image and is called the weak plate model. The second order Mumford-Shah model (6.2) considered in the present work corresponds to the weak plate energy where the kink parameter is set to zero. In many image processing applications one is typically interested in detecting the jumps. Hence, setting the kink parameter to zero is reasonable. The weak plate energy has later been termed “Blake & Zisserman model” by several authors; e.g., [39]. We avoid this term as the finite difference discretization of the first order Mumford-Shah functional is frequently called Blake-Zisserman functional as well; e.g., [79].

Regarding theoretical results, the existence of minimizers has been shown in [39]. Related work on minimizers is [41, 24]. In particular, [41] studies properties of local minimizers and [24] considers the generic uniqueness of minimizers. Other aspects are Euler equations [40] and density estimates [42]. The approximation by elliptic functionals in the sense of Γ-convergence and related algorithms based on these approximations have been investigated in [2]. (As corresponding approximations for the first order Mumford-Shah model are frequently named after Ambrosio-Tortorelli, we call these approximations Ambrosio-Tortorelli type approximations.) Partial Γ-convergence and compactness results are obtained in [28] for a discrete potential formulation of the second order Mumford-Shah model.

Concerning algorithms, Blake and Zisserman proposed the graduated non-convexity (GNC) approach [21]. Another class of algorithms is based on Ambrosio-Tortorelli type approximations. For first order models such an approach was first proposed in [2]. For second order models, it has been considered in [216, 215] where they propose a numerical scheme which employs the Ambrosio-Tortorelli type approximation of the second order Mumford-Shah model.

6.2. Higher order Mumford-Shah models based on jet formulations

In this section, we reformulate higher order Mumford-Shah models and, in particular, second order Mumford-Shah models in terms of Taylor jets. Based on these reformulations, we propose a suitable discretization to obtain discrete higher order Mumford-Shah models. We start with the anisotropic second order case, proceed with the extension to near-isotropic discretizations and then discuss the case of higher order Mumford-Shah models. Finally, we briefly discuss discrete models based on distances between Taylor jets and discuss their relation to the approach considered before. Particularly, we reveal a specific instance, which is motivated by classical univariate Hermite interpolation.

6.2.1. Jet formulation of higher order Mumford-Shah models

We here derive a new formulation of higher order Mumford-Shah models in terms of Taylor jets. For basics on Taylor jets –as we need them in this section– we refer to Section 2.3. We first develop the concepts for single-channel images and generalize them to multi-channel images later on (in Section 6.3.4). We start with second order Mumford-Shah models of the form (6.2) and consider higher order Mumford-Shah models afterwards.

Jet formulation of second order Mumford-Shah models. In the following, we use the notation introduced in (2.20), i.e., the first order Taylor jet Ju of the twice continuously differentiable function $u : \Omega \to \mathbb{R}$ in the point x is represented by $J_x = (u(x), a_J(x), b_J(x))$ with $a_J(x) = \partial_1 u(x)$, $b_J(x) = \partial_2 u(x)$.

As a first step, we use this representation to rewrite the squared Frobenius norm of the

Hessian matrix employed in (6.2) as

$$\mathcal{R}(Ju_x) = \mathcal{R}(J_x) := \|D^2 u_x\|_F^2 = \lambda\, \partial_{11}u(x)^2 + (1-\lambda)\, \partial_1 a_J(x)^2 + \lambda\, \partial_{22}u(x)^2 \\ + (1-\lambda)\, \partial_2 b_J(x)^2 + \partial_2 a_J(x)^2 + \partial_1 b_J(x)^2, \tag{6.3}$$

taking into account the constraints

$$a_J(x) = \partial_1 u(x), \quad \text{and} \quad b_J(x) = \partial_2 u(x). \tag{6.4}$$

The consistency condition (6.4) ensures that $(u(x), a_J(x), b_J(x))$ represents a Taylor jet. Thus, (6.3) is indeed a reformulation of the Frobenius norm of the Hessian matrix of u for any real valued parameter λ. From now on, we let $\lambda \in [0,1]$ in (6.3). We reformulate the consistency condition for a_J in (6.4) to its integral form given by

$$C_a(J_x) : \; u(x_1, x_2) + \int_{x_1}^{x_1+\rho} a_J(\xi, x_2)\, d\xi = u(x_1 + h, x_2) - \int_{x_1+\rho}^{x_1+h} a_J(\xi, x_2)\, d\xi, \tag{6.5}$$

where $h > 0$ and ρ is such that $0 < \rho < h$. Analogously, we reformulate the consistency condition for b_J in (6.4) as

$$C_b(J_x) : \; u(x_1, x_2) + \int_{x_2}^{x_2+\rho} b_J(x_1, \xi)\, d\xi = u(x_1, x_2 + h) - \int_{x_2+\rho}^{x_2+h} b_J(x_1, \xi)\, d\xi. \tag{6.6}$$

The following basic proposition ensures that (6.5) together with (6.6) are indeed a reformulation of the consistency condition (6.4), i.e., that (6.4) is equivalent to (6.5) together with (6.6).

Proposition 6.1. *The consistency constraints given by* (6.4) *and the conditions as given by* $C_a(J_x)$ *and* $C_b(J_x)$ *are equivalent.*

Proof. We proof the equivalence of the first item of (6.4) and (6.5) and note that the proof for the second item of (6.4) and (6.6) works analogously. The identity (6.5) follows from (6.4) by simply plugging in $a(x) = \partial_1 u(x)$ and using the fundamental theorem of calculus. Conversely, if (6.5) holds for all $h > 0$, then also $\frac{1}{h}(u(x_1 + h, x_2) - u(x_1, x_2)) = \frac{1}{h}\int_{x_1}^{x_1+h} a(\xi, x_2) d\xi$. Then taking the limit $h \to 0$ yields $\partial_1 u(x) = a(x)$ which can be seen by, e.g., applying the mean value theorem to the right-hand side before taking the limit. □

As a consequence of Proposition 6.1, we will also call (6.5) and (6.6) consistency conditions. We note that we will frequently drop the index J in our notation in the following.

By using (6.3) together with the consistency conditions (6.5) and (6.6) we obtain a formulation of the second order Mumford-Shah model (6.2) in terms of Taylor jets given by

$$\underset{J,\Gamma}{\operatorname{argmin}} \int_\Omega |u(x) - f(x)|^2\, dx + \beta^2 \int_{\Omega\setminus\Gamma} \mathcal{R}(J_x)\, dx + \gamma\, \operatorname{length}(\Gamma) \tag{6.7}$$

$$\text{subject to} \quad C_a(J_x) \quad \text{and} \quad C_b(J_x), \qquad \text{for all } x \in \Omega \setminus \Gamma. \tag{6.8}$$

The minimization takes place w.r.t. first order jets J which are represented in the form (2.20) and the constraint (6.8) ensures that J is indeed the Taylor jet of a function u, i.e., $J = Ju$. As in the function setting before, the set Γ is chosen from the set of finite unions of smooth curves. We record the following observation whose proof is straight-forward.

Theorem 6.2. *The second order Mumford-Shah model* (6.2) *and its jet formulation* (6.7), (6.8) *are equivalent. In particular, the constrained problem of minimizing* (6.7) *subject to* (6.8) *has a minimizer.*

Proof. First, we consider a minimizer u^* (together with the edge set Γ^*) of (6.2) and let $Ju^* = J^1u^*$ be its first order Taylor jet defined by (2.13). We note that Ju^* is well-defined on the complement of the closed edge set Γ^*. By its definition as a Taylor jet, Ju^* meets the constraints (6.8). The functional value of Ju^* (together with Γ^* as second argument) w.r.t. (6.7) equals the functional value of u^* (together with Γ^* as second argument) w.r.t. (6.2). Furthermore, for any admissible jet J, i.e., for J fulfilling the constraints (6.8), (with edge set Γ) the functional value of J (together with Γ as second argument) w.r.t. (6.7) equals the functional value of u (together with Γ as second argument) w.r.t. (6.2). Here, u denotes the coefficient function u obtained by identifying the jet $J(x)$ with its coefficient functions $u(x)$, $a_J(x)$, and $b_J(x)$ according to (2.20). Hence, if (u^*, Γ^*) minimizes (6.2), then (Ju^*, Γ^*) minimizes (6.7) under the constraints (6.8).

For the converse direction we assume that the jet J^* (together with the edge set Γ^*) minimizes (6.7) subject to (6.8). We identify the jet J^* with its coefficient functions u^*, a_J^*, and b_J^* according to (2.20). We note that the functional value of (J^*, Γ^*) w.r.t. (6.7) equals the functional value of u^* w.r.t. (6.2). Further, for a candidate function u, together with a candidate edge set Γ, the functional value w.r.t. (6.7) of its Taylor jet Ju (which fulfills (6.8)) together with Γ equals the functional value of (u, Γ) w.r.t. (6.2). Hence, (u^*, Γ^*) is a minimizer of (6.2).

The statement on the existence of a minimizer for (6.7) subject to (6.8) is a consequence of the existence of a minimizer of the equivalent problem (6.2) which has been shown in [39]. (Please note that in this reference a minimizer of (6.2) is called a strong minimizer of the Blake-Zisserman functional.) □

Remark 6.3. We can also obtain an analogous jet formulation for any weighted ℓ_2-type norm $\|\nabla^2 u(x)\|_w^2 = w_{11}\partial_{11}u(x)^2 + 2w_{12}\partial_{12}u(x)^2 + w_{22}\partial_{22}u(x)^2$ with nonnegative weights w_{11}, w_{12}, w_{22}, by inserting the weights into (6.3) and proceeding analogously.

Jet formulation of general higher order Mumford-Shah models. In the following, we use the notation of Section 2.3 for general higher order differentials. We consider the general k-th order Mumford-Shah model

$$\operatorname*{argmin}_{u,\Gamma} \int_\Omega |u(x) - f(x)|^2 \, dx + \beta^2 \int_{\Omega\setminus\Gamma} \|D^k u(x)\|_w^2 \, dx + \gamma \operatorname{length}(\Gamma). \tag{6.9}$$

Here, $D^k u(x)$ denotes a general k-th order differential of u at x and the squared norm

$$\|D^k u(x)\|_w^2 = \sum_{\alpha : |\alpha| = k} w_\alpha \, \tfrac{k!}{\alpha!} \, |\partial_\alpha u(x)|^2 \tag{6.10}$$

measures the magnitude of $D^k u(x)$. The summation in (6.10) takes place w.r.t. all multiindices α of order k, the symbol w_α denotes a positive weight for each α and $\partial_\alpha u(x)$ denotes the k-th order derivative of the function u at x w.r.t. the directions given by α. (We note that the special case $k = 2$ together with $w_\alpha = 1$ for all α corresponds to the problem (6.7), (6.8) considered before.) The arguments of the k-th order Mumford-Shah functional are the (closed) discontinuity set Γ as well as a function u which is k times continuously differentiable on $\Omega \setminus \Gamma$. The other symbols have the same role as in the first and second order Mumford-Shah models. We emphasize that the k-th order Mumford-Shah functional (6.9) does not penalize polynomial trends of order $k-1$ within u. As discussed for the second order model, a natural norm $\|D^k u(x)\|$ for $D^k u(x)$ would be its norm as a (symmetric) k-linear form. However, since computing this expression involves the computation of eigenvalues for the unknown u at any point x. Thus, employing this norm would be computationally very demanding. For this reason we consider (6.10) which yields a computationally more accessible alternative.

We introduce new functions v_{α_1}, one function for each multiindex α_1 with $\alpha_1 < \alpha$. Analogously to (6.4), we employ the consistency condition

$$v_{\alpha_1}(x) = \partial_{\alpha_1} u(x), \tag{6.11}$$

where in particular $v_{(0,0)} = u$. By using the consistency condition (6.11) we rewrite (6.10) as

$$\mathcal{R}(J_x^{k-1}) \; := \mathcal{R}(J_x) \; = \; \|D^k u(x)\|_w^2 = \sum_{(\alpha_1,\alpha_2)\in Z} w'_{(\alpha_1,\alpha_2)} |\partial_{\alpha_2} v_{\alpha_1}(x)|^2, \tag{6.12}$$

where (α_1, α_2) are the members of the set Z of admissible pairs of multiindices and $w'_{(\alpha_1,\alpha_2)}$ are the respective nonnegative weights. (The weights $w'_{(\alpha_1,\alpha_2)}$ are chosen such that the sum of all $w'_{(\alpha_1,\alpha_2)}$ which represent the same multiindex α, i.e., $\alpha_1 + \alpha_2 = \alpha$, equal $w_\alpha \frac{k!}{\alpha!}$.) The set of admissible pairs of multiindices is given by

$$Z = \left\{(\alpha_1, \alpha_2) \in \mathbb{N}_0^2 \times \mathbb{N}_0^2 : |\alpha_1| + |\alpha_2| = k;\ \text{precisely one component of } \alpha_2 \text{ is zero}\right\}. \tag{6.13}$$

We point out that, for each of the newly introduced functions v_{α_1}, only partial derivatives with respect to one coordinate direction are involved. Next, we restate the consistency condition (6.11) for each $(\alpha_1, \alpha_2) \in Z$ (which implies that α_2 is nonzero) in integral form as

$$C_{(\alpha_1,\alpha_2)}(J_x) : v_{\alpha_1}(x) + \int_x^{x+\rho\frac{\alpha_2}{|\alpha_2|}} v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(\xi)\, d\xi = v_{\alpha_1}(x + h\tfrac{\alpha_2}{|\alpha_2|}) - \int_{x+\rho\frac{\alpha_2}{|\alpha_2|}}^{x+h\frac{\alpha_2}{|\alpha_2|}} v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(\xi)\, d\xi, \tag{6.14}$$

for $h > 0$ and ρ with $0 < \rho < h$. As for the second order case, the equivalence of the differential constraints (6.11) and the integral constraints (6.14) is a consequence of the fundamental theorem of calculus.

The reformulation of the k-th order penalty (6.12) together with the consistency conditions (6.14) allows us to formulate general higher order Mumford-Shah models of the form (6.9) in terms of Taylor jets by

$$\operatorname*{argmin}_{J,\Gamma} \int_\Omega |u(x) - f(x)|^2\, dx + \beta^2 \int_{\Omega\setminus\Gamma} \mathcal{R}(J_x)\, dx + \gamma \operatorname{length}(\Gamma) \tag{6.15}$$

$$\text{subject to } C_{(\alpha_1,\alpha_2)}(J_x), \quad \text{for all } (\alpha_1,\alpha_2) \in Z, \quad \text{and for all } x \in \Omega \setminus \Gamma. \tag{6.16}$$

The minimization takes place w.r.t. $(k-1)$-th order jets; the constraints (6.16) ensure that J_x is the Taylor jet of a function; again, the discontinuity set Γ is chosen from the set of finite unions of smooth curves.

6.2.2. Discretization

To apply the jet formulation of the Mumford-Shah models to image data (which are discrete) we need a discrete formulation. We start out with the (anisotropic) discretization of the jet formulation of the second order Mumford-Shah model (6.7). Then we consider means to derive more isotropic discretizations of the second order Mumford-Shah model. Finally, we discuss discretizations of higher order Mumford-Shah models and discuss on how to increase the isotropy in this general situation.

We fix the notation. First, we let h denote the grid size of the discrete rectangular image domain $\Omega_h = \{h, 2h, \dots, hm\} \times \{h, 2h, \dots, hn\}$ and we use the symbols $e_1 = (1,0)^T$, $e_2 = (0,1)^T$ to denote the compass directions.

In the discrete situation, the data term is given as the sum of the quadratic deviations between the discrete data $f : \Omega_h \to \mathbb{R}$ and the discrete candidate $u : \Omega_h \to \mathbb{R}$. We use the following notion of a discrete edge set which is suitable for our purposes.

Definition 6.4 (Discrete edge set). As edge set Γ of a discrete image we understand a subset of the edges $K = \Big\{\{x, x+e_1\}, \{x, x+e_2\} \,:\, x, x+e_1, x+e_2 \in \Omega_h\Big\}$ of the (undirected) four-neighborhood graph with vertex set Ω_h. We denote the edges in direction e_1 by Γ^1 and those in direction e_2 by Γ^2. We denote the number of edges in Γ^j by $|\Gamma^j|$, $j = 1, 2$.

We use the symbol Γ for both discrete and continuous edge sets. It will be clear from the context if we refer to Γ as the edge set in the continuous domain or its discrete counterpart.

Next, we need a discretization of the second order penalty (6.3). As the image domain is a pixel grid, we use finite differences. The second directional derivatives $\partial_{11}u$ and $\partial_{22}u$ of u for the coordinate axis in (6.3) are discretized in terms of second order central differences of u in direction e_1 and e_2, respectively. The discrete gradients of the jet coefficients a and b in (6.3) are given by forward differences in directions e_1 and e_2. More precisely, we denote the forward differences operator w.r.t. the direction e_1 by ∇_1, i.e., $\nabla_1 a(x) = \frac{a(x+he_1)-a(x)}{h}$. Further, we denote by ∇_1^2 the second order differences operator, i.e., $\nabla_1^2 u(x) = \frac{u(x-he_1)-2u(x)+u(x+he_1)}{h^2}$. The analogue expressions for the direction e_2 are denoted by ∇_2 and ∇_2^2, respectively. To discretize the integrals in the consistency condition (6.5) we employ the simplest quadrature rule based on piecewise constant approximation, i.e.,

$$\int_{x_1}^{x_1+\rho} a_J(\xi, x_2)\, d\xi \approx \rho a(x_1, x_2) \quad \text{and} \quad \int_{x_1+\rho}^{x_1+h} a_J(\xi, x_2)\, d\xi \approx (h-\rho) a(x_1+h, x_2),$$

respectively. We consider the quadratic penalty relaxation of the problem (6.7), (6.8). By using the approximation above we obtain the penalty $\alpha\Big|u(x) + \rho a_J(x) - u(x+he_1) + (h-\rho) a_J(x+he_1)\Big|^2$, where $\alpha > 0$ is a parameter. This term yields the *discrete soft constraints* which correspond to $C_a(J_x)$. (We note that employing hard constraints is undesired as will be explained in Remark 6.5 below.) In the same way, we obtain the discrete soft constraints corresponding to $C_b(J_x)$, that is, $\alpha\Big|u(x) + \rho b_J(x) - u(x+he_2) + (h-\rho) b_J(x+he_2)\Big|^2$. After including the discretization of $\mathcal{R}(J_x)$ in terms of first and second differences as detailed above, we obtain the term $P(J_x) = P^1(J_x) + P^2(J_x)$ which denotes the discrete second order Mumford-Shah regularizer for the smooth part in the jet formulation; it is given by

$$P^1(J_x) = P^1_{\lambda,\alpha,\rho}(J_x) = \lambda\, |\nabla_1^2 u(x)|^2 + (1-\lambda)\, |\nabla_1 a_J(x)|^2 + |\nabla_1 b_J(x)|^2 \\ + \alpha \left| u(x) + \rho a_J(x) - u(x+he_1) + (h-\rho) a_J(x+he_1) \right|^2, \tag{6.17}$$

and

$$P^2(J_x) = P^2_{\lambda,\alpha,\rho}(J_x) = \lambda\, |\nabla_2^2 u(x)|^2 + (1-\lambda)\, |\nabla_2 b_J(x)|^2 + |\nabla_2 a_J(x)|^2 \\ + \alpha \left| u(x) + \rho b_J(x) - u(x+he_2) + (h-\rho) b_J(x+he_2) \right|^2. \tag{6.18}$$

We discuss the parameters α, λ, ρ. The jet-parameter α controls the coupling between the image u and the slope variables a_J and b_J. The jet-parameter $\lambda \in [0,1]$ balances the second order differences in u and the first order differences in the slope variables a_J and b_J, respectively. Intuitively, for large values of α, the usage of different values of the parameter λ has a small influence since then the slopes are tightly coupled with the image. In view of symmetry, a reasonable choice for the evaluation parameter $\rho \in [0,h]$ corresponds to $\rho = h/2$. In connection with this, we notice that, in order to obtain symmetry, it would also be possible to use a pair of evaluation parameters ρ_1, ρ_2 with $\rho_2 = h - \rho_1$ and average the results. As we will see in Section 6.3.2, this is algorithmically more expensive than a single evaluation parameter which motivates to employ $\rho = h/2$ throughout this chapter. For a discussion on the practical influence of the parameters λ and α we refer to the experimental section.

Remark 6.5. We note that the straight-forward discretization of the hard constraint (6.5), i.e., requiring that $u(x_1,x_2) + \rho a(x_1,x_2) = u(x_1+h,x_2) - (h-\rho)a(x_1+h,x_2)$, where $h > 0$ and $0 < \rho < h$, is undesired. This constraint means to evaluate the affine-linear polynomials/the jet elements sitting in (x_1,x_2) and (x_1+h,x_2) in the point $(x_1+\rho h, x_2)$ and require that they are equal. In turn this would mean that the polynomials in neighboring points are *equal* which is undesired here because this corresponds to a discrete variant of the affine-linear Potts model instead. This interesting model is considered separately in Chapter 7.

Discrete objective of the second order Mumford-Shah model; anisotropic version. We have gathered all necessary ingredients for a discrete jet formulation of the second order Mumford-Shah model (6.7); it is given by

$$\operatorname*{argmin}_{J,\Gamma^1,\Gamma^2} \|u-f\|^2 + \sum_{x\in\Omega_h\setminus\Gamma^1} \beta^2 P^1(J_x) + \sum_{x\in\Omega_h\setminus\Gamma^2} \beta^2 P^2(J_x) + \gamma|\Gamma^1| + \gamma|\Gamma^2|. \tag{6.19}$$

In Figure 6.2, we illustrate the resulting (discrete) jet J for different choices of β. In particular, we see that the jet elements are allowed to vary within segments, that is, the corresponding affine planes may be twisted w.r.t. their neighbors. The amount of variation is controlled by the smoothing parameter β. (Recall that for $\beta \to \infty$ no variation is allowed.)

We record that the discrete model (6.19) has a minimizer.

Theorem 6.6. *The discrete second order Mumford-Shah jet model in its anisotropic version* (6.19) *has a minimizer.*

We give the proof of Theorem 6.6 together with the proof of the analogue statement for the near isotropic discretization which is formulated as Theorem 6.10 below.

The following example motivates a particular choice of the jet parameters α, λ which can be related to the notion of univariate Hermite splines.

Example 6.7. *For the choice $\lambda = 0$, $\alpha = 12/h^6$, where h denotes the grid size, and the evaluation point $\rho = h/2$ in* (6.17) *and* (6.18) *we obtain*

$$P^1(J_x) = |\nabla_1 a(x)|^2 + |\nabla_1 b(x)|^2 + \tfrac{12}{h^6}\Big|u(x) + \tfrac{h}{2}a(x) - u(x+he_1) + \tfrac{h}{2}a(x+he_1)\Big|^2, \tag{6.20}$$

and

$$P^2(J_x) = |\nabla_2 a(x)|^2 + |\nabla_2 b(x)|^2 + \tfrac{12}{h^6}\Big|u(x) + \tfrac{h}{2}b(x) - u(x+he_2) + \tfrac{h}{2}b(x+he_2)\Big|^2; \tag{6.21}$$

We call the problem (6.19) *with the implementations* (6.20) *and* (6.21) *the* discrete second order spline-related Mumford-Shah jet model.

To motivate this name, we notice that the cubic (univariate) Hermite spline p which interpolates the values $u(x), u(x+he_1)$ and *the directional derivatives $a(x), a(x+he_1)$ in direction e_1 along the segment $[x, x+he_1]$ is characterized by the minimizing property*

$$p^* = \underset{p \text{ is a cubic polynmial}}{\operatorname{argmin}} \int_x^{x+he_1} p''(z)^2 dz, \tag{6.22}$$

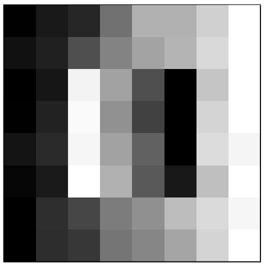
subject to the constraints that the cubic polynomial p interpolates both the values $u(x), u(x+he_1)$ and the directional derivatives $a(x), a(x+he_1)$. The minimizing energy of p^* in (6.22) *is given by*

$$d(J_x, J_{x+he_1}) := |a(x) - a(x+he_1)|^2 + \tfrac{12}{h^6}\Big|u(x) + \tfrac{h}{2}a(x) - u(x+he_1) + \tfrac{h}{2}a(x+he_1)\Big|^2. \tag{6.23}$$

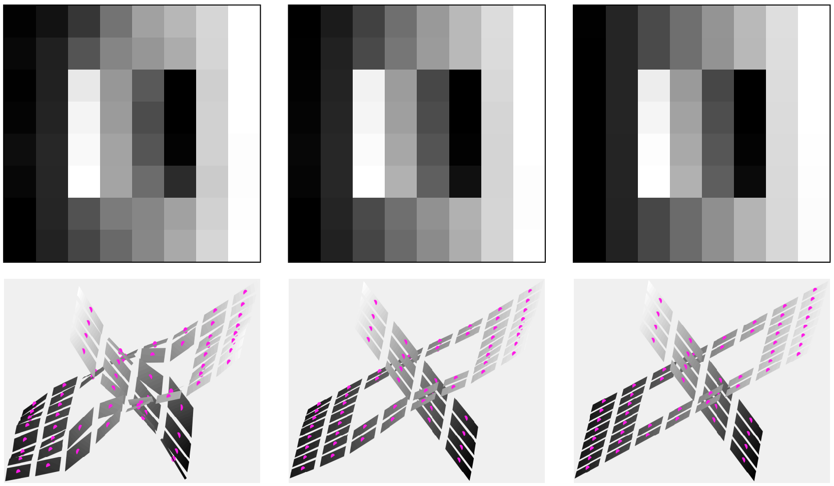
This expression can also be seen as a distance between two line elements consisting of function value and directional derivative, i.e., two univariate first order Taylor jets. (This is detailed in Section 6.2.3, where we study discrete second order models in terms of distances between jets.) After adding a corresponding term for the derivatives in the orthogonal direction, we obtain precisely (6.20). *The analogous consideration with the roles of the coordinate directions e_1 and e_2 interchanged yields* (6.21).

Increasing the isotropy, near isotropic discretizations. We extend our approach to obtain a near-isotropic discretization of the second order Mumford-Shah model; the benefit of employing a near-isotropic discretization is illustrated in Figure 6.3. To this end, we consider the following generalization of Definition 6.4 w.r.t. eight neighborhoods.

Definition 6.8. The *eight-neighborhood edge set* Γ of a (discrete) image is a subset Γ of the edges K of the (undirected) eight-neighborhood graph with vertex set Ω_h which is given by $K = \Big\{\{x, x+e_j\} : x, x+e_j \in \Omega_h \text{ for } j = 1,2,3,4\Big\}$, for the coordinate directions e_1, e_2 and the diagonal directions $e_3 = e_1 + e_2$, $e_4 = e_1 - e_2$. The edges in the coordinate directions e_1, e_2 are denoted by Γ^1 and Γ^2, respectively, and those in the diagonal directions e_3 and e_4 by Γ^3 and Γ^4, respectively. The number of edges in Γ^j are denoted by $|\Gamma^j|$, $j = 1,2,3,4$.



(a) Input data



(b) $\beta = 0.25$ (c) $\beta = 0.75$ (d) $\beta = 2$

Figure 6.2.: Illustration of the (discrete) jet coefficients for three values of the smoothing parameter β. The pink spheres correspond to the pixel values of the smoothed images u above and the small plates illustrate the first order polynomials in each pixel given by u, a, b. We observe that for larger values of β, less variation is allowed between jet elements (outside edges). In the limit case $\beta \to \infty$, no variation is allowed which corresponds to another model, the affine-linear Potts model. (The affine-linear Potts model is considered separately in Chapter 7.)

Towards including an eight-neighborhood in our discretization, we use the invariance of the Frobenius norm w.r.t. rotations and rewrite the second order Mumford-Shah model (6.2) equivalently as

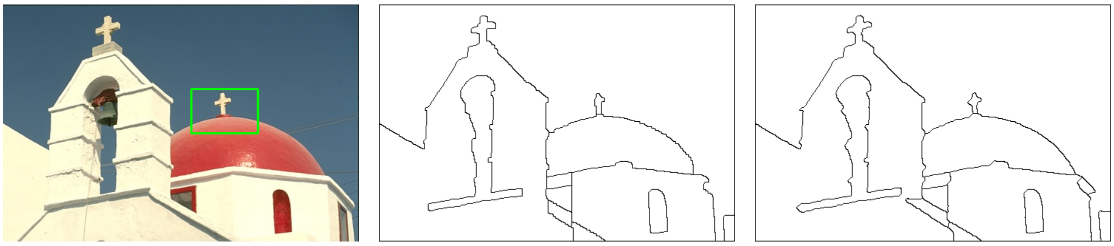
$$\operatorname*{argmin}_{u,\Gamma} \|u - f\|_2^2 + \beta^2 \int_{\Omega \setminus \Gamma} \left(\tfrac{1}{2} \|\nabla^2 u(x)\|_F^2 + \tfrac{1}{2} \|R \nabla^2 u(x) R^T\|_F^2 \, dx \right) + \gamma \operatorname{length}(\Gamma), \tag{6.24}$$

where R denotes the rotation w.r.t. the angle $\pi/4$ about the origin in $\mathbb{R}^2$. By applying the rotations R and R^T to the Hessian matrix $\nabla^2 u(x)$ of u in (6.24) we obtain the identity

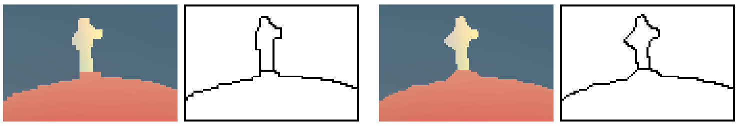
$$R \nabla^2 u(x) R^T = \frac{1}{2} \begin{pmatrix} \partial_{11} u(x) + 2\partial_{12} u(x) + \partial_{22} u(x) & \partial_{22} u(x) - \partial_{11} u(x) \\ \partial_{22} u(x) - \partial_{11} u(x) & \partial_{11} u(x) - 2\partial_{12} u(x) + \partial_{22} u(x) \end{pmatrix}. \tag{6.25}$$

The entries on the main diagonal on the right-hand side of (6.25) correspond to the second directional derivatives w.r.t. the diagonal directions $e_3/\sqrt{2} = (e_1 + e_2)/\sqrt{2}$ and $e_4/\sqrt{2} = (e_1 -$

(a) Noisy input (b) Anisotropic discretization (c) Near-isotropic discretization



(d) Original [4] (e) Edge set of (b) (f) Edge set of (c)



(g) Details of (b) and (e) (h) Details of (c) and (f)

Figure 6.3.: Comparison of the anisotropic and the near-isotropic discretization. The the near-isotropic discretization yields smoother boundaries than the anisotropic discretization which exhibits “geometric staircasing” (see, e.g., the detail view or the edge between the sky and the cupola).

$e_2)/\sqrt{2}$, respectively. The off diagonal entries correspond to the mixed second derivatives w.r.t. the directions $e_3/\sqrt{2}$ and $e_4/\sqrt{2}$. As a result, (6.25) corresponds to the Hessian matrix for the coordinate system formed by $e_3/\sqrt{2}$ and $e_4/\sqrt{2}$. In the following, we use the symbols

$$c(x) = \partial_{e_3} u(x) \quad \text{and} \quad d(x) = \partial_{e_4} u(x), \tag{6.26}$$

to denote the directional derivatives of u w.r.t. the directions e_3 and e_4, respectively. By linearity, the diagonal derivatives $c(x)$ and $d(x)$ can be written as $c(x) = a(x) + b(x)$ and $d(x) = a(x) - b(x)$, respectively. (We recall that the symbols a, b were used to denote the directional derivatives w.r.t. the coordinate axes.) We employ this notation in (6.25) to rewrite the second summand in the integral on the right-hand side of (6.24) as

$$\begin{aligned} \mathcal{R}_{\frac{\pi}{4}}(J_x) := \|R\nabla^2 u(x) R^T\|^2 &= \tfrac{1}{4}\lambda|\partial_{e_3}\partial_{e_3} u(x)|^2 + \tfrac{1}{4}\lambda|\partial_{e_4}\partial_{e_4} u(x)|^2 \\ &+ \tfrac{1}{4}(1-\lambda)|\partial_{e_3} c(x)|^2 + \tfrac{1}{4}(1-\lambda)|\partial_{e_4} d(x)|^2 + \tfrac{1}{4}|\partial_{e_4} c(x)|^2 + \tfrac{1}{4}|\partial_{e_3} d(x)|^2, \end{aligned} \tag{6.27}$$

where ∂_v denotes the directional derivative w.r.t. the direction v.

We fix the jet notation which is suitable for a more isotropic discretization of the second order Mumford-Shah model. We consider jets J represented via (2.20) by $J_x := (u(x), a_J(x), b_J(x))$ as before and we further define $c = c_J := a_J + b_J$ as well as $d = d_J := a_J - b_J$. (We note that a_J, b_J play the role of variables and c_J, d_J are defined via a_J, b_J.)

Similarly to (6.5), (6.6), we can write the consistency conditions for $C_c(J_x)$ and $C_d(J_x)$ in integral form. For $C_c(J_x)$ this corresponds to

$$C_c(J_x) \ : \ u(x) + \int_x^{x+\rho e_3} c(\xi)\, d\xi = u(x + he_3) - \int_{x+\rho e_3}^{x+he_3} c(\xi)\, d\xi, \tag{6.28}$$

where $h > 0$ and ρ is such that $0 < \rho < h$. In terms of a_J, b_J this corresponds to

$$C_c(J_x) \ : \ u(x) + \int_x^{x+\rho e_3} (a(\xi) + b(\xi)) \ d\xi = u(x + he_3) - \int_{x+\rho e_3}^{x+he_3} (a(\xi) + b(\xi)) \ d\xi. \tag{6.29}$$

The above integrals are understood as univariate integrals along the straight line segments which connect the points x, $x + \rho e_3$ and $x + \rho e_3$, $x + e_3$, respectively. Analogously, we obtain for $C_d(J_x)$,

$$C_d(J_x) \ : \ u(x) + \int_x^{x+\rho e_4} (a(\xi) - b(\xi)) \ d\xi = u(x + he_4) - \int_{x+\rho e_4}^{x+he_4} (a(\xi) - b(\xi)) \ d\xi. \tag{6.30}$$

We use the consistency conditions in the above form to rewrite the second order Mumford-Shah problem (6.2) as its jet formulation

$$\begin{aligned} &\operatorname*{argmin}_{J,\Gamma} \int_\Omega |u(x) - f(x)|^2 \, dx + \beta^2 \int_{\Omega\setminus\Gamma} \tfrac{1}{2} \left(\mathcal{R}(J_x) + \mathcal{R}_{\frac{\pi}{4}}(J_x)\right) \, dx + \gamma \operatorname{length}(\Gamma) \\ &\text{subject to} \quad C_a(J_x), C_b(J_x), C_c(J_x), C_d(J_x), \quad \text{for all } x \in \Omega \setminus \Gamma, \end{aligned} \tag{6.31}$$

where $\mathcal{R}(J_x)$ was defined in (6.3).

We now consider discretizing (6.31). Similarly to (6.17)-(6.18), we obtain the discrete second order penalties with respect to the diagonal directions e_3, e_4. The second order penalty with respect to e_3 is given by

$$\begin{aligned} P^3(J_x) =& \tfrac{1}{4}\lambda \, |\nabla_3^2 u(x)|^2 + \tfrac{1}{4}(1-\lambda) \, |\nabla_3 c(x)|^2 + \tfrac{1}{4}|\nabla_3 d(x)|^2 \\ &+ \alpha' \Big| u(x) + \rho c(x) - u(x + he_3) + (h - \rho) c(x + he_3) \Big|^2, \end{aligned} \tag{6.32}$$

where we recall that $c(x)$ is short for $a(x) + b(x)$ and $d(x)$ is short for $a(x) - b(x)$. Further, the first and second order finite differences ∇_3 of c, d and ∇_3^2 of u w.r.t. the direction $e_3 = e_1 + e_2$ are given by $\nabla_3 c(x) = \frac{1}{h} \left(c(x + he_3) - c(x)\right)$ and $\nabla_3^2 u(x) = \frac{1}{h^2} \left(u(x - he_3) - 2u(x) + u(x + he_3)\right)$. Analogously, we obtain the second order penalty for e_4,

$$\begin{aligned} P^4(J_x) =& \tfrac{1}{4}\lambda \, |\nabla_4^2 u(x)|^2 + \tfrac{1}{4}(1-\lambda) \, |\nabla_4 d(x)|^2 + \tfrac{1}{4}|\nabla_4 c(x)|^2 \\ &+ \alpha' \Big| u(x) + \rho d(x) - u(x + he_4) + (h - \rho) d(x + he_4) \Big|^2, \end{aligned} \tag{6.33}$$

where ∇_4, ∇_4^2 denote the first and second order differences w.r.t. the direction $e_4 = e_1 - e_2$. The parameters λ, ρ have the same roles as discussed below (6.18). The parameter $\alpha' > 0$ denotes the diagonal analogue of the parameter α discussed below (6.18).

Remark 6.9. The factor 1/4 which appears in (6.32) and (6.33) accounts for taking second order differences w.r.t. the vectors e_3 and e_4 which have length $\sqrt{2}$. The same factor appears in front of a_J and b_J as for them first order differences are formed w.r.t. the vectors e_3, e_4. Further, they are related to first order derivatives of u for the same vectors of length $\sqrt{2}$.

Discrete objective of the second order Mumford-Shah model, near isotropic version. We can now formulate the near-isotropic discrete second order Mumford-Shah model in jet formulation,

$$\operatorname*{argmin}_{J,\Gamma} \|u-f\|^2 + \beta^2 \sum_{s=1}^{4} \sum_{x\in\Omega_h\setminus\Gamma^s} \tfrac{1}{2} P^s(J_x) + \gamma \sum_{s=1}^{4} \omega_s \, |\Gamma^s|. \tag{6.34}$$

As for the first order Mumford-Shah and Potts model (cf. Section 4.2.1), we employ the weights ω_s given by $\omega_1 = \omega_2 = \sqrt{2}-1$ and $\omega_3 = \omega_4 = 1 - \frac{\sqrt{2}}{2}$. We record the existence of minimizers.

Theorem 6.10. *The discrete second order Mumford-Shah jet model in its near isotropic version* (6.34) *has a minimizer.*

Since the proofs are similar, we give a unified proof of Theorem 6.6 and Theorem 6.10.

Proof of Theorem 6.6 and Theorem 6.10. For the anisotropic situation of Theorem 6.6, we denote the energy in (6.19) by $F(\Gamma,J)$, and rewrite it in the following suitable way:

$$F(\Gamma,J) = \gamma|\Gamma^1| + \gamma|\Gamma^2| + E_\Gamma(J) \tag{6.35}$$

with

$$E_\Gamma(J) = \|u-f\|^2 + \sum_{x\in\Omega_h\setminus\Gamma^1} \beta^2 P^1(J_x) + \sum_{x\in\Omega_h\setminus\Gamma^2} \beta^2 P^2(J_x). \tag{6.36}$$

Analogously, for the situation of Theorem 6.10, we also use the symbol $F(\Gamma,J)$ to denote the energy in (6.34) and rewrite it as

$$F(\Gamma,J) = \gamma \sum_{s=1}^{4} \omega_s \, |\Gamma^s| + E_\Gamma(J) \tag{6.37}$$

with

$$E_\Gamma(J) = \|u-f\|^2 + \beta^2 \sum_{s=1}^{4} \sum_{x\in\Omega_h\setminus\Gamma^s} \tfrac{1}{2} P^s(J_x). \tag{6.38}$$

Now we can reformulate the assertion of both theorems in a unified way: $F(\Gamma,J)$ has a minimizer.

For a fixed edge set Γ the function which incorporates all terms depending on the discrete jet J is given by $E_\Gamma(J)$ in (6.35) and (6.37), respectively. In both cases, finding the minimum of $E_\Gamma(J)$ w.r.t. J corresponds to solving a least squares problem in the discrete coefficient functions u, a, b of the discrete jet $J = (u, a, b)$. Therefore, there is a (not necessarily unique) jet J_Γ^* that minimizes $E_\Gamma(J)$ for fixed Γ. By denoting $\tilde{E}(\Gamma) := \min_J E_\Gamma(J)$, we obtain

$$\inf_{\Gamma} \tilde{E}(\Gamma) = \inf_{\Gamma,J} F(\Gamma,J). \tag{6.39}$$

Since the set to optimize for on the left-hand side of (6.39) is finite, the infimum is attained and a minimum exists. Having found a minimizer Γ^* of $\tilde{E}(\Gamma)$, then (Γ^*, J_Γ^*) is a minimizer of $F(\Gamma,J)$. Thus, the right-hand side of (6.39) has a minimizer which completes the proof. □

We extend the spline-related discrete model introduced in Example 6.7 to the more isotropic situation.

Example 6.11. *We here discuss the more isotropic variant of the discrete second order spline-related Mumford-Shah jet model introduced in Example 6.7. In addition to the regularizing atoms* P^1 *and* P^2 *defined via* (6.20) *and* (6.21)*, respectively, we employ the atoms* P^3 *and* P^4 *in* (6.34)*. For* P^3, P^4*, we choose* $\lambda = 0$, $\alpha' = \frac{3}{2h^6}$ *for the grid size* h *in* (6.32) *and* (6.33)*. Together, we obtain*

$$P^3(J_x) = \tfrac{1}{4}|\nabla_3 c(x)|^2 + \tfrac{1}{4}|\nabla_3 d(x)|^2 + \tfrac{3}{2h^6}\Big|u(x) + \tfrac{h}{2}c(x) - u(x + he_3) + \tfrac{h}{2}d(x + he_3)\Big|^2, \tag{6.40}$$

and

$$P^4(J_x) = \tfrac{1}{4}|\nabla_4 d(x)|^2 + \tfrac{1}{4}|\nabla_4 c(x)|^2 + \tfrac{3}{2h^6}\Big|u(x) + \tfrac{h}{2}c(x) - u(x + he_4) + \tfrac{h}{2}d(x + he_4)\Big|^2. \tag{6.41}$$

We call the problem (6.34) *with the implementations of* P^s *given by* (6.20), (6.21), (6.40) *and* (6.41), *the* discrete near-isotropic second order spline-related Mumford-Shah jet model.

In our experiments (see Figure 6.3 and Section 6.4), we compare the anisotropic (four-neighborhood) and the near-isotropic (eight-neighborhood) discretization. We observe that the near-isotropic discretization significantly reduces geometric staircasing effects.

Remark 6.12. We note that the key ingredient to derive more isotropic discretizations is the invariance of the Frobenius norm under orthogonal transformations. In particular, this makes the functional invariant under rotations. This observation makes it possible to increase the isotropy further by including knight move differences (recall Section 4.2.1).

Since knight moves correspond to rotations of an angle chosen from the set $\{\arctan(\pm 1/3), \arctan(\pm 3)\}$, we may employ the techniques used to derive (6.24) (6.25) to obtain analogous formulas for the knight move case. Then it is possible to derive a corresponding jet formulation as well as a corresponding discretization by following the recipe for the four- and eight-neighborhood case.

Extension to higher order models. We point out the considerations needed to generalize the approach presented for the second order Mumford-Shah model to higher order models.

We start with the anisotropic case. We recall the jet formulation of higher order Mumford-Shah models (6.15), (6.16). We start from (6.15), (6.16) and derive a discretization by following the lines of the anisotropic second order case. The notions of discrete edge sets and discrete data terms remain the same as in the second order case. We observe that (6.15), (6.16) use higher order differentials up to order k. Each differential depends on one coordinate direction only. To discretize l-th order derivatives we employ (central) l-th order finite differences. More precisely, by (6.13) the multiindex α_2 has the form $\alpha_2 = (l, 0)$ or $\alpha_2 = (0, l)$ for a positive integer $l \leq k$. If $\alpha_2 = (l, 0)$, we use $\nabla_1^l u(x) = \sum_{r=0}^{l} \binom{r}{l} (-1)^{r+1} u(x_1 + \lceil l/2 \rceil - r, x_2)$ where $\lceil \cdot \rceil$ assigns the smallest integer which is larger or equal to the argument. If $\alpha_2 = (0, l)$, we employ the difference $\nabla_2^l u(x) = \sum_{r=0}^{l} \binom{r}{l} (-1)^{r+1} u(x_1, x_2 + \lceil l/2 \rceil - r)$.

Next, we have to discretize the consistency conditions (6.14) in integral form for each (α_1, α_2) in the admissible set Z. To this end, we use the simplest quadrature rule based on piecewise constant approximation (as in the second order case) to obtain

$$\int_x^{x+\rho\frac{\alpha_2}{|\alpha_2|}} v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(\xi)\,d\xi \approx \rho v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(x) \quad \text{and}$$

$$\int_{x+\rho\frac{\alpha_2}{|\alpha_2|}}^{x+h\frac{\alpha_2}{|\alpha_2|}} v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(\xi)\,d\xi \approx (h-\rho) v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(x+h\frac{\alpha_2}{|\alpha_2|}).$$

By using these discretizations we obtain the discrete objective function of the (anisotropic) k-th order Mumford-Shah model given by

$$\operatorname*{argmin}_{J,\Gamma} \|u-f\|^2 + \sum_{x\in\Omega_h\setminus\Gamma^1} \beta^2 P^1(J_x) + \sum_{x\in\Omega_h\setminus\Gamma^2} \beta^2 P^2(J_x) + \gamma|\Gamma^1| + \gamma|\Gamma^2|. \tag{6.42}$$

Here, J_x denotes a $(k-1)$-th order jet which is represented by the (now discrete) grid functions $v_{\alpha_1}(x)$ with α_1 such that $|\alpha_1| < k$. Further, for $s \in \{1,2\}$, we employ the discrete higher order penalties given by

$$\begin{aligned} P^s(J_x) = P^s_{\theta,\rho}(J_x) = &\sum_{(\alpha_1,\alpha_2)\in Z_s} w'_{(\alpha_1,\alpha_2)} \left|\nabla_s^{|\alpha_2|} v_{\alpha_1}\right|^2 \\ &+ \theta \sum_{(\alpha_1,\alpha_2)\in Z_s} \left| v_{\alpha_1}(x) + \rho v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(x) - v_{\alpha_1}(x+h\tfrac{\alpha_2}{|\alpha_2|}) + (h-\rho) v_{\alpha_1+\frac{\alpha_2}{|\alpha_2|}}(x+h\tfrac{\alpha_2}{|\alpha_2|})\right|^2, \end{aligned} \tag{6.43}$$

where the weights $w'_{(\alpha_1,\alpha_2)}$ are given in (6.12) and where we subdivide the set Z in (6.13) into

$$Z_1 = \{(\alpha_1,\alpha_2) : |\alpha_1| + |\alpha_2| = k;\ \text{the second component of } \alpha_2 \text{ equals } 0\}, \tag{6.44}$$

and

$$Z_2 = \{(\alpha_1,\alpha_2) : |\alpha_1| + |\alpha_2| = k;\ \text{the first component of } \alpha_2 \text{ equals } 0\}. \tag{6.45}$$

Finally, we point out how to obtain near isotropic discretizations of general higher order Mumford-Shah models. Here, we assume that the weights w_α in (6.10) all equal one. We need this assumption to guarantee the rotational invariance of the (continuous domain) higher order Mumford-Shah functional which is shown by the following well-known lemma. The essential point here is that the weights are equal, that is, equality to one is not needed but convenient from a notational point of view. We record the following folklore lemma. We give a short proof for the reader's convenience.

Lemma 6.13. *Let $f : \Omega \to \mathbb{R}$ be a sufficiently smooth function defined on an open set $\Omega \subset \mathbb{R}^d$ with d being a positive integer. Further, let R be a rotation, i.e., R denotes an orthogonal linear operator with determinant equal to one. Then*

$$\sum_{i_1,\dots,i_k=1}^{d} |D^k f_x(Re_{i_1},\dots,Re_{i_k})|^2 = \sum_{i_1,\dots,i_k=1}^{d} |D^k f_x(e_{i_1},\dots,e_{i_k})|^2. \tag{6.46}$$

Please note that for $k=2$, (6.46) states the rotational invariance of the Hessian matrix of f.

Proof. The assertion of the lemma follows by induction. The case $k = 1$ follows from the rotational invariance of the Euclidean norm of the gradient. The induction step is given by

$$\begin{aligned}\sum_{i_1,\ldots,i_k=1}^{d} |D^k f_x(Re_{i_1},\ldots,Re_{i_k})|^2 &= \sum_{i_1,\ldots,i_k=1}^{d} |D^k f_x(e_{i_1},\ldots,e_{i_{k-1}},Re_{i_k})|^2 \\ &= \sum_{i_1,\ldots,i_k=1}^{d} |D^k f_x(e_{i_1},\ldots,e_{i_k})|^2,\end{aligned}$$

where the first equality follows from the induction hypothesis and the second equality from the rotational invariance of the Euclidean norm of the gradient. □

Lemma 6.13 allows us to rewrite the general higher order Mumford-Shah model (6.9) with unit weights as

$$\begin{aligned}\operatorname*{argmin}_{u,\Gamma} \int_\Omega |u(x) - f(x)|^2\, dx + \beta^2 \int_{\Omega\setminus\Gamma} \left(\tfrac{1}{2}\|D^k u(x)\|_F^2 + \tfrac{1}{2}\|D^k u(x)(R\,\cdot,\ldots,R\,\cdot)\|_F^2\right) dx \quad (6.47)\\ + \gamma\,\mathrm{length}(\Gamma),\end{aligned}$$

We may now proceed similarly to the paragraphs after (6.31) to derive a corresponding jet formulation and then a corresponding more isotropic discretization. Finally, we note that Remark 6.12 also applies to the case of higher order Mumford-Shah models. Hence, if needed, it is possible to include knight moves in the setup by incorporating corresponding additional rotation terms into (6.47).

6.2.3. Discrete formulations based on distances between polynomials

The regularizing terms of (discrete) *first* order Mumford-Shah models may be formulated in terms of distances of neighboring items of the discrete image u. In the anisotropic case, this corresponds to the functional

$$R(J,\Gamma) = \beta^2 \sum_{x\in\Omega_h\setminus\Gamma^1} d(u_{x_1+h,x_2}, u_{x_1,x_2})^2 + \beta^2 \sum_{x\in\Omega_h\setminus\Gamma^2} d(u_{x_1,x_2+h}, u_{x_1,x_2})^2 + \gamma|\Gamma|,$$

where d is a distance. For single-channel images, d amounts to the absolute value of the differences and for multi-channel images, d is typically the Euclidean distance; see, e.g., [21, 101]. For higher order models, higher order differences of u are involved such that a direct interpretation in terms of distances is not possible at first glance. However, we shift the perspective to distances between Taylor jets instead of function values. This allows to generalize the distance based perspective to higher orders. We here develop this perspective and reveal that the discrete second order spline-related Mumford-Shah jet models introduced in Examples 6.7 and 6.11 are instances of such distances.

We start out by considering the functional which describes the anisotropic (discrete) situation,

$$R(J,\Gamma) = \beta^2 \sum_{x\in\Omega_h\setminus\Gamma^1} d(J_{x_1+h,x_2}, J_{x_1,x_2})^2 + \beta^2 \sum_{x\in\Omega_h\setminus\Gamma^2} d(J_{x_1,x_2+h}, J_{x_1,x_2})^2 + \gamma|\Gamma|. \quad (6.48)$$

Here, as previously, Γ denotes the edge set and J denotes the jet to optimize for. The symbol d is new in (6.48): d may depend on the location x as well as on the direction e_i, $i = 1, 2$,

$$d = \left(d^i_{x_1,x_2}\right)_{x=(x_1,x_2)\in\Omega,\, i\in\{1,2\}} . \tag{6.49}$$

For fixed x, i we require that $d^i_{x_1,x_2}$ is a distance between polynomials. More precisely, $d^i_{x_1,x_2}$ is non-negative, symmetric and fulfills the triangle inequality. In particular, $d^i_{x_1,x_2}$ is zero if and only if its two arguments define the same polynomial. In the second order case, they define the same affine-linear function. Summing over all indices yields a distance on the corresponding (discrete) jets. Therefore, we call d a *jet distance.* The remaining symbols in (6.48) have the same meaning as before.

We can obtain instances of $d^i_{x_1,x_2}$ from a set of linear functionals $\varphi_j = (\varphi_j)^i_{x_1,x_2} : \Pi_{k-1} \to \mathbb{R}$ on the linear space of $(k-1)$-th order bivariate polynomials Π_{k-1} which may depend on x_1, x_2, i. It follows from basic linear algebra that we have to choose linear functionals φ_j such that the dimension of their span equals the dimension of Π_{k-1}. Then we define

$$d^i_{x_1,x_2}(J, J') = \sum_j \left|\varphi_j(J - J')\right|^2 . \tag{6.50}$$

In the second order situation, we may consider three linearly independent linear functionals on the linear space of affine-linear polynomials $\varphi_j : \Pi_1 \to \mathbb{R}$. Thereby, we obtain

$$d^i_{x_1,x_2}(J, J') = \sum_{j=1}^{3} \left|\varphi_j(J - J')\right|^2 . \tag{6.51}$$

By basic linear algebra, $d^i_{x_1,x_2}(J, J') = 0$ if and only if J and J' describe the same polynomial. The triangle inequality and symmetry hold by construction. Therefore, (6.50) and (6.51) indeed define distances. We note that such distances have the additional benefit of being distances induced by norms, and even more, by scalar products.

We consider examples.

Example 6.14. *We define a first class of jet distances for first order jets which are based on slope and point evaluations: we fix a first order polynomial P and define the linear functionals*

$$\varphi_1(P) = (\varphi_1)^i_{x_1,x_2}(P) = \frac{\partial}{\partial x_1} P(z), \quad \varphi_2(P) = (\varphi_2)^i_{x_1,x_2}(P) = \frac{\partial}{\partial x_2} P(z), \quad \textit{and} \quad \varphi_3(P) = \sqrt{\alpha} P(z).$$

Here, φ_1, φ_2 yield the partial derivatives of P and φ_3 evaluates P in the point z weighted by $\alpha > 0$. If we consider a first order jet J in its representation $J_x = (u(x), a_J(x), b_J(x))$, we see that $\varphi_1(J_x) = a_J(x)$, $\varphi_2(J_x) = b_J(x)$, $\varphi_3(J_x) = \sqrt{\alpha} J_x(z)$, $\alpha > 0$, in (6.51)*. In particular, φ_1 and φ_2 do not depend on the evaluation point z, whereas φ_3 does. (Note that all functionals depend on x_1, x_2. The functional φ_3 in particular depends on the evaluation point z which itself will be chosen to depend on x_1, x_2.) Using these functionals, we obtain the jet distance d defined via*

$$d^i_{x_1,x_2}(J, J') = |a_x - a'_x|^2 + |b_x - b'_x|^2 + \alpha |J_x(z) - J'_x(z)|^2 . \tag{6.52}$$

The remaining degrees of freedom in (6.52) *are the points z in which the the jet elements are evaluated and the weight $\alpha > 0$. A reasonable choice of the evaluation point z is a point on the line segment between x and $x + e_i$, i.e. $z = x + \rho e_i$, $\rho \in [0, 1]$.*

In view of (6.17) and (6.18), we see that the jet distances of Example 6.14 yield discretizations of the second order Mumford-Shah model in the sense of Section 6.2.2. More precisely, they define the subclass of (6.17), (6.18) with the jet-parameter λ being zero.

Finally, we remark that maybe the simplest class of jet distances is given by point evaluations only. This means that each φ_j corresponds to the evaluation in a point z_j. For first order jets, this results in the jet distance

$$d(J_x, J_y) = \sum_{j=1}^{3} |J_x(z_j) - J_y(z_j)|^2, \tag{6.53}$$

where the three points z_j are non-collinear. A reasonable choice of the evaluation point z_j is given by $z_1 = x$, $z_2 = x + e_1$, and $z_3 = x + e_2$. (Alternatively, one could choose more than three points whose span equals $\mathbb{R}^2$.)

Spline-based jet distance. We discuss the discrete second order spline-related Mumford-Shah jet models introduced in Examples 6.7 and 6.11. In the following, we interpret and motivate this approach in the context of distances between polynomials. To this end, we consider two *one-dimensional* jet elements J_x and J_{x+h}, $x \in \mathbb{R}$, with function values and slopes $u(x), u(x+h), a(x), a(x+h)$. We recall that cubic Hermite splines interpolate data points *and* given derivative data by third order polynomials. We consider the cubic Hermite polynomial p which interpolates the data $u(x), u(x+h), a(x), a(x+h)$ and interpret its energy as a univariate jet distance, i.e., we consider $d(J_x, J_{x+h}) = \int_x^{x+h} p''(z)^2 \, dz$. We note that (6.22) is zero if and only if the interpolating polynomial p is affine-linear. This is in turn the case if and only if the jet elements J_x, J_{x+h} are equal (which means that J_x, J_{x+h} describe the same affine-linear function/polynomial). The univariate spline-based jet distance then reads

$$d(J_x, J_{x+h}) = |a(x) - a(x+h)|^2 + \tfrac{12}{h^6}\left|u(x) + \tfrac{h}{2}a(x) - u(x+h) + \tfrac{h}{2}a(x+h)\right|^2. \tag{6.54}$$

We note that the second summand in (6.54) corresponds to evaluating J_x and J_{x+h} in the point $x + \frac{1}{2}h$ and comparing the resulting values.

Motivated by the univariate situation, we define the spline-based jet distance d_{spline} by

$$d_{\text{spline}}(J_x, J_y) = |a(x) - a(y)|^2 + |b(x) - b(y)|^2 + \tfrac{12}{h^6}\left|u(x) + \tfrac{h}{2}a(x) - u(y) + \tfrac{h}{2}a(y)\right|^2. \tag{6.55}$$

For the other direction, we simply exchange the roles of a and b.

We observe that the spline-based jet distance (6.55) falls into the class of jet distances defined Example 6.14. This can be seen by choosing the weight $\alpha = \frac{12}{h^6}$ and the point evaluation parameter $\rho = \frac{1}{2h}$ in (6.52).

Remark 6.15. We point out that the generalization to more isotropic settings (using diagonal directions, as well as knight move directions) is straightforward. For instance, the spline based jet distance which incorporates diagonal directions is given via (6.20), (6.21), (6.40), (6.41). Further, jet distances on $(k-1)$-th order polynomials may be employed to deal with general k-th order Mumford-Shah problems. Examples which are based on evaluations of point values and values of higher derivatives can be deduced from (6.43) (by choosing $w'_{(\alpha_1,\alpha_2)} = 0$ for $|\alpha_2| > 1$).

6.3. Algorithmic approach

Here, we develop an algorithmic approach to the discrete second order Mumford-Shah model in its jet formulation (6.19). To this end, we employ an ADMM splitting strategy. (ADMM approaches have turned out to often work well for various non-convex problems, e.g., [197, 211, 54, 101].) We derive the proposed ADMM splitting in Section 6.3.1. The problem is split in a way such that all subproblems can be solved exactly. In Section 6.3.2, we derive efficient solvers for these subproblems which are based on the dynamic programming techniques developed in Chapter 3. We further provide the extensions to more isotropic discretizations and to multi-channel images in Section 6.3.3 and in Section 6.3.4, respectively.

6.3.1. Splitting approach to the anisotropic discretization with the ADMM

We consider the jet-based discrete Mumford-Shah model in its anisotropic formulation (6.19). The derivations in the more isotropic case are similar but involve more notation. We consider them in Section 6.3.3 below. As a starting point, we split the target jet J into two variables J^1, J^2 subject to the constraint that they are equal:

$$\begin{aligned} \operatorname*{argmin}_{J^1,J^2,\Gamma^1,\Gamma^2} \Bigg\{ & F(J^1,J^2,\Gamma^1,\Gamma^2) = \\ & \tfrac{1}{2}\|u^1 - f\|^2 + \tfrac{1}{2}\|u^2 - f\|^2 + \beta^2\Big(\sum_{x\in\Omega_h\setminus\Gamma^1} P^1(J^1_x) + \sum_{x\in\Omega_h\setminus\Gamma^2} P^2(J^2_x)\Big) + \gamma|\Gamma^1| + \gamma|\Gamma^2| \Bigg\} \end{aligned} \tag{6.56}$$

$$\text{subject to} \quad J^1_x = J^2_x \text{ for all } x \in \Omega_h.$$

The constraints in (6.56) ensure that (6.56) and the original problem (6.19) are equivalent. We recall that two jets are equal if and only if their offsets and slopes are equal (2.20). Thus, we understand the equality constraints in (6.56) as the equalities of the discrete offsets and slopes of J^1 and J^2. We treat them as $m \times n$-matrices $u^1, u^2, a^1, a^2, b^1, b^2$. (Recall that $\|\cdot\|$ corresponds to the Frobenius norm.) Towards an ADMM approach, we need to formulate the augmented Lagrangian of (6.56). It is given by

$$\begin{aligned} \mathcal{L}_{\mu,\nu}(J^1,J^2,\Gamma^1,\Gamma^2,\tau_u,\tau_a,\tau_b) = {} & F(J^1,J^2,\Gamma^1,\Gamma^2) + \tfrac{\mu}{2}\|u^1 - u^2\|^2 + \langle \tau_u, u^1 - u^2\rangle \\ & + \tfrac{\nu}{2}\|a^1 - a^2\|^2 + \langle \tau_a, a^1 - a^2\rangle + \tfrac{\nu}{2}\|b^1 - b^2\|^2 + \langle \tau_b, b^1 - b^2\rangle. \end{aligned} \tag{6.57}$$

The equality constraints in (6.56) are now part of the objective in the form of quadratic deviations. The variables $\tau_u, \tau_a, \tau_b \in \mathbb{R}^{m\times n}$ denote the Lagrange multipliers which correspond to the equality constraints. We note that $\mu, \nu > 0$ are parameters which determine how strong differences between the split variables are penalized. In this context, we note that we use different coupling parameters for the pixel values u and the slopes a, b since both typically live on different scales.

Each iteration of the ADMM corresponds to minimizing the Lagrangian $\mathcal{L}$ w.r.t. J^1, Γ^1 and w.r.t. J^2, Γ^2 followed by gradient ascents on the Lagrange multipliers τ_u, τ_a, τ_b. In the following, we derive this iterative scheme. First, we complete the squares in (6.57) and obtain

$$\begin{aligned} \mathcal{L}_{\mu,\nu}(J^1,J^2,\Gamma^1,\Gamma^2,\tau_u,\tau_a,\tau_b) = {} & F(J^1,J^2,\Gamma^1,\Gamma^2) + \tfrac{\mu}{2}\|u^1 - (u^2 - \tfrac{\tau_u}{\mu})\|^2 - \tfrac{1}{2\mu}\|\tau_u\|^2 \\ & + \tfrac{\nu}{2}\|a^1 - (a^2 - \tfrac{\tau_a}{\nu})\|^2 - \tfrac{1}{2\nu}\|\tau_a\|^2 + \tfrac{\nu}{2}\|b^1 - (b^2 - \tfrac{\tau_b}{\nu})\|^2 - \tfrac{1}{2\nu}\|\tau_b\|^2. \end{aligned} \tag{6.58}$$

We derive the subproblem corresponding to J^1. (The derivation for J^2 is completely analogous.) The minimization problem in J^1 is given by

$$\begin{aligned}\operatorname*{argmin}_{J^1,\Gamma^1} \tfrac{1}{2}\|u^1 - f\|^2 + \gamma\,|\Gamma^1| + \sum_{x\in\Omega_h\setminus\Gamma^1} \beta^2 P^1(J^1_x) + \tfrac{\mu}{2}\|u^1 - (u^2 - \tfrac{\tau_u}{\mu})\|^2 \\ + \tfrac{\nu}{2}\|a^1 - (a^2 - \tfrac{\tau_a}{\nu})\|^2 + \tfrac{\nu}{2}\|b^1 - (b^2 - \tfrac{\tau_b}{\nu})\|^2.\end{aligned} \tag{6.59}$$

The other terms in (6.58) do not depend on J^1, so we dropped them. To bring (6.59) into a convenient form we use the fact that

$$\sum_{i=1}^{N} x_i(p - t_i)^2 = \Big(\sum_{i=1}^{N} x_i\Big)\Big(p - \tfrac{\sum_{i=1}^N t_i x_i}{\sum_{i=1}^N x_i}\Big)^2 + C \tag{6.60}$$

holds for $p, t_1, \dots, t_N \in \mathbb{R}$ and $x_1, \dots, x_N > 0$ and a constant C that does not depend on p [181]. We apply (6.60) to the terms in (6.59) which involve u^1 and obtain

$$\begin{aligned}\operatorname*{argmin}_{J^1,\Gamma^1} \Big\{ \gamma|\Gamma^1| + \sum_{x\in\Omega_h\setminus\Gamma^1} \beta^2 P^1(J^1_x) + \tfrac{1+\mu}{2}\Big\|u^1 - \tfrac{f+\mu(u^2-\tau_u/\mu)}{1+\mu}\Big\|^2 \\ + \tfrac{\nu}{2}\|a^1 - (a^2 - \tfrac{\tau_a}{\nu})\|^2 + \tfrac{\nu}{2}\|b^1 - (b^2 - \tfrac{\tau_b}{\nu})\|^2 \Big\}.\end{aligned} \tag{6.61}$$

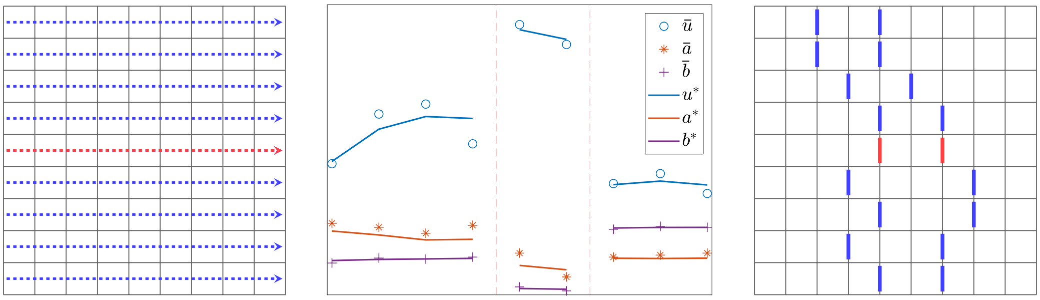
A final multiplication by $\frac{2}{\nu}$ yields the desired subproblem w.r.t. J^1. As a result, we obtain the following iterative scheme.

$$\begin{aligned}\Big((J^1)^{j+1}, (\Gamma^1)^{j+1}\Big) = \operatorname*{argmin}_{J^1,\Gamma^1} \tfrac{2\gamma}{\nu_j}|\Gamma^1| + \tfrac{1+\mu_j}{\nu_j}\Bigg\|u^1 - \frac{f+\mu_j\Big((u^2)^j - \tau_u^j/\mu_j\Big)}{1+\mu_j}\Bigg\|^2 + \sum_{x\in\Omega\setminus\Gamma^1} \tfrac{2\beta^2}{\nu_j} P^1(J^1_x) \\ + \Big\|a^1 - \Big((a^2)^j - (\tau_a)^j/\nu_j\Big)\Big\|^2 + \Big\|b^1 - \Big((b^2)^j - (\tau_b)^j/\nu_j\Big)\Big\|^2,\end{aligned}$$

$$\begin{aligned}\Big((J^2)^{j+1}, (\Gamma^2)^{j+1}\Big) = \operatorname*{argmin}_{J^2,\Gamma^2} \tfrac{2\gamma}{\nu_j}|\Gamma^2| + \tfrac{1+\mu_j}{\nu_j}\Bigg\|u^2 - \frac{f+\mu_j\Big((u^1)^{j+1} + \tau_u^j/\mu_j\Big)}{1+\mu_j}\Bigg\|^2 + \sum_{x\in\Omega\setminus\Gamma^2} \tfrac{2\beta^2}{\nu_j} P^2(J^2_x) \\ + \Big\|a^2 - \Big((a^1)^{j+1} + (\tau_a)^j/\nu_j\Big)\Big\|^2 + \Big\|b^2 - \Big((b^1)^{j+1} + (\tau_b)^j/\nu_j\Big)\Big\|^2,\end{aligned}$$

$$\begin{aligned}(\tau_u)^{j+1} &= (\tau_u)^j + \mu_j((u^1)^{j+1} - (u^2)^{j+1}),\\ (\tau_a)^{j+1} &= (\tau_a)^j + \nu_j((a^1)^{j+1} - (a^2)^{j+1}),\\ (\tau_b)^{j+1} &= (\tau_b)^j + \nu_j((b^1)^{j+1} - (b^2)^{j+1}).\end{aligned} \tag{6.62}$$

We employ increasing sequences $(\mu_j)_{j\in\mathbb{N}}$, $(\nu_j)_{j\in\mathbb{N}}$ for the penalty parameters as it is common when dealing with non-convex problems. Thereby, we allow the splitting variables to evolve rather independently in the beginning and become closer in the course of the iterations. We stop the iteration (6.62) when the jets J^1 and J^2 become (approximately) equal.



(a) Decomposition of the first ADMM suproblem

(b) Solving the segmented jet problem for the data along the red arrow

(c) The boundaries in (b) yield the horizontal edges Γ^1

Figure 6.4.: (a) The first ADMM subproblem in (6.62) decomposes into univariate segmented jet problems along the pixel rows. (b) The univariate segmented jet problem along the red arrow in (a) for data $\overline{u}, \overline{a}, \overline{b}$ and solution u^*, a^*, b^*. The red dashed lines indicate the boundaries of the optimal partition $\mathcal{I}^*$. (c) The partition boundaries of the univariate subproblems (b) constitute the directional edges Γ^1 in (6.62).

6.3.2. Efficient solution of the subproblems

In each iteration of the ADMM (6.62), we have to solve the two non-convex minimization problems in J^1 and J^2, respectively. The crucial observation is that the second order penalty P^1 operates exclusively along the rows of an image and P^2 along the columns (recall (6.17) and (6.18)). As a consequence, the subproblems in (6.62) decompose into *univariate* segmented least squares problems along the rows and columns of the image domain Ω. In Figure 6.4, we give an illustration of the decomposition of the first subproblem.

These subproblems can be solved in parallel. We develop a solver based on the ideas of the solver for univariate higher order Mumford-Shah problems of Chapter 3. In particular, we adapt it to deal with the present problems.

We consider exemplarily the first subproblem of (6.62) as the second subproblem of (6.62) works analogously. To this end, we reformulate the univariate problem in terms of partitions instead of the edges Γ^1 : for offset data $\overline{u} \in \mathbb{R}^n$ and slope data $\overline{a}, \overline{b} \in \mathbb{R}^n$ these problems have the generic form given by

$$(J^*, \mathcal{I}^*) = \tag{6.63}$$
$$\underset{\substack{J=(u,a,b)^T \in \mathbb{R}^n,\\ \mathcal{I} \text{ partition of } \{1,\dots,n\}}}{\operatorname{argmin}} \sum_{I \in \mathcal{I}} \left\{ \quad \eta^2 \|u_I - \overline{u}_I\|_2^2 + \|a_I - \overline{a}_I\|_2^2 + \|b_I - \overline{b}_I\|_2^2 + \beta'^2 \sum_{j=1}^{|I|} P^1(J_j) \right\} + \gamma'(|\mathcal{I}| - 1),$$

where P^1 corresponds to the second order penalty (6.17) and γ', η, β' are nonnegative parameters. A solution of (6.63) depends on the jet J as well as the partition $\mathcal{I}$ which is given as a set of discrete intervals of the form $I = \{l, l+1, \dots, r\}$ (cf. Section 3.2). As in Chapter 3, we denote a discrete interval with bounds l, r by $I = l : r$. The penalty term P^1 involves first and second order differences. In this context, we note that the first order difference ∇ is only well-defined for vectors with length greater than 1 and the second order difference ∇^2 for length greater than 2, respectively. Consequently, we let $\nabla a_I = \nabla b_I = 0$ for $|I| = 1$ and $\nabla^2 u_I = 0$ for $|I| \leq 2$.

We explain the relation between the partition $\mathcal{I}$ and the directional edges Γ^1 in (6.62). There is an edge between neighboring pixels x and $x + e_1$, that is $\{x, x + e_1\} \in \Gamma^1$, if and only if x and $x + e_1$ belong to two distinct segments $I_1, I_2 \in \mathcal{I}$. Consequently, $|\mathcal{I}| - 1$ counts the number of edges in Γ^1 along the image row; see also Figure 6.4.

We can formulate problem (6.63) in terms of the partition only by

$$\mathcal{I}^* = \underset{\mathcal{I} \text{ partition of } 1:n}{\operatorname{argmin}} \sum_{I \in \mathcal{I}} \left(\mathcal{E}^I + \gamma' \right), \tag{6.64}$$

where $\mathcal{E}^I$ denotes the (optimal) jet approximation error on the segment I, that is,

$$\mathcal{E}^I = \min_{J=(v,w,z)} \eta^2 \|v - \bar{u}_I\|_2^2 + \|w - \bar{a}_I\|_2^2 + \|z - \bar{b}_I\|_2^2 + \beta'^2 \sum_{j=1}^{|I|} P(J_j). \tag{6.65}$$

After an optimal partition $\mathcal{I}^*$ in (6.64) was found, the optimal jet J^* is recovered by solving the least squares objective (6.65) w.r.t. u, a, b for each segment $I \in \mathcal{I}$ separately.

The partitioning problem (6.64) can be solved efficiently by the algorithm developed in Chapter 3. To see this, we denote the functional in (6.64) by B, i.e.,

$$B(\mathcal{I}) = \sum_{I \in \mathcal{I}} \left(\mathcal{E}^I + \gamma' \right). \tag{6.66}$$

Accordingly, we denote the minimal value on the reduced domain $1 : r$ by

$$B_r^* = \min_{\mathcal{I} \text{ partition on } 1:r} B(\mathcal{I}). \tag{6.67}$$

The minimal value B_r^* on the domain $1 : r$ satisfies the Bellman equation

$$B_r^* = \min_{l=1,\dots,r} \left\{ \mathcal{E}^{l:r} + \gamma' + B_{l-1}^* \right\}, \tag{6.68}$$

where we set $B_0^* = -\gamma'$. Thus, by the dynamic programming principle we can compute $B_1^*, B_2^*, \dots$ until we reach B_n^*. We keep track of an optimal partition $\mathcal{I}^*$ by storing at step r the minimizing argument l' of (6.68) as the value L_r, so that L encodes the boundaries of an optimal partition.

In contrast to the problems considered in Chapter 3, the problems here are given in terms of discrete jets (6.63) instead of discrete functional values only. Nevertheless, the error update scheme of Algorithm 3.1 can be applied. To this end, we reformulate (6.65) as the least squares problem

$$\mathcal{E}^{l:r} = \min_{v,w,z} \|A^q \left(v_1, w_1, z_1, \dots, v_q, w_q, z_q\right)^T - g^{lr}\|^2, \tag{6.69}$$

where $q = r - l + 1$ denotes the length of the interval $I = l : r$ and the system matrix $A^q \in \mathbb{R}^{(7q-5)\times(3q)}$ and the vector $g^{l:r} \in \mathbb{R}^{7q-5}$ are given by

$$A^q = \begin{pmatrix} E^q \\ \beta' D^q \end{pmatrix}, \quad g^{l:r} = \left(\eta\bar{u}_l, \bar{a}_l, \bar{b}_l,\ \eta\bar{u}_{l+1}, \bar{a}_{l+1}, \bar{b}_{l+1},\ \dots,\ \eta\bar{u}_r, \bar{a}_r, \bar{b}_r, 0, \dots, 0 \right)^T. \tag{6.70}$$

Here, the diagonal matrix E^q and the band matrix D^q in A^q are given by

$$E^q = \begin{pmatrix} \eta & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & & & \eta & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \in \mathbb{R}^{3q \times 3q} \quad \text{and} \quad D^q = \begin{pmatrix} d_1^T & & \\ d_2^T & & \\ d_3^T & & \\ d_4^T & & \\ & \ddots & \\ & & d_1^T \\ & & d_2^T \\ & & d_3^T \end{pmatrix} \in \mathbb{R}^{4q-5 \times 3q}$$

for the band vectors $d_1, \ldots, d_4$. The band vector d_1 represents the softened jet-consistency constraint via

$$d_1 = \sqrt{\alpha}\begin{pmatrix}1 & \rho & 0 & -1 & (1-\rho) & 0\end{pmatrix}^T, \tag{6.71}$$

the band vector d_2 the weighted forward difference in the slope variable a by

$$d_2 = \sqrt{1-\lambda}\begin{pmatrix}0 & -1 & 0 & 0 & 1 & 0\end{pmatrix}^T, \tag{6.72}$$

the band vector d_3 the forward difference in the slope variable b,

$$d_3 = \begin{pmatrix}0 & 0 & -1 & 0 & 0 & 1\end{pmatrix}^T, \tag{6.73}$$

and the band vector d_4 the weighted second-order central difference in the pixel values u by

$$d_4 = \sqrt{\lambda}\begin{pmatrix}1 & 0 & 0 & -2 & 0 & 0 & 1\end{pmatrix}^T. \tag{6.74}$$

Thus, the system matrix A^q has a (sparse) band structure such that we can employ the error update strategy developed in Chapter 3 and in turn Algorithm 3.1 (cf. Remark 3.11). We recall that Algorithm 3.1 provides a theoretical worst case quadratic runtime guarantee for solving (6.64) which scales linearly in practice when using the pruning strategies discussed in Chapter 3.

We recall that we discussed the possibility of using a pair of evaluation parameters ρ_1, ρ_2 in Section 6.2.2. There, we pointed out that such a choice is algorithmically more expensive. Indeed, in this situation the band matrix D^q has two band vectors of the form of d_1 instead of one. This would increase the computational costs as more matrix entries need to be eliminated for each error update.

6.3.3. Extension to more isotropic discretizations

We extend the algorithmic approach of Section 6.3.1 to the more isotropic discretization (6.34) of the second order Mumford-Shah model. To this end, we split the target jet J of the discretized problem (6.75) into four variables $J^1, \ldots, J^4$. Analogously to Section 6.3.1, we obtain the constrained problem

$$\begin{aligned} &\operatorname*{argmin}_{J^1,\ldots,J^4,\Gamma} \left\{ F\left(\{J^s\}_s, \Gamma\right) = \sum_{s=1}^{4} \left\{ \tfrac{1}{4}\|u^s - f\|^2 + \beta^2 \sum_{x \in \Omega_h \setminus \Gamma^s} \tfrac{1}{2} P^s(J^s_x) + \omega_s \gamma \, |\Gamma^s| \right\} \right\} \\ &\text{subject to} \quad J^s_x = J^t_x \quad \text{for all } x \in \Omega_h, \text{ and } 1 \le s < t \le 4. \end{aligned} \tag{6.75}$$

As in Section 6.3.1, we apply the ADMM to (6.75). To this end, we formulate the augmented Lagrangian of (6.75) which is given by

$$\begin{aligned}\mathcal{L}_{\mu,\nu}(\{J^s\}_s, \{\Gamma^s\}_s, \{\tau_u^{s,t}\}, \{\tau_a^{s,t}\}, \{\tau_b^{s,t}\}) = F\Big(\{J^s\}_s, \{\Gamma^s\}_s\Big) + \sum_{s=1}^{4} \sum_{t=s+1}^{4} \Big(\tfrac{\mu}{2}\|u^s - (u^t - \tfrac{\tau_u^{s,t}}{\mu})\|^2 \\ - \tfrac{1}{2\mu}\|\tau_u^{s,t}\|^2 + \tfrac{\nu}{2}\|a^s - (a^t - \tfrac{\tau_a^{s,t}}{\nu})\|^2 - \tfrac{1}{2\nu}\|\tau_a^{s,t}\|^2 + \tfrac{\nu}{2}\|b^s - (b^t - \tfrac{\tau_b^{s,t}}{\nu})\|^2 - \tfrac{1}{2\nu}\|\tau_b^{s,t}\|^2\Big).\end{aligned} \tag{6.76}$$

The symbols $\tau_u^{s,t}, \tau_a^{s,t}, \tau_b^{s,t}$ denote the Lagrange multipliers which correspond to the equality constraints $J^s = J^t$. After some algebraic manipulations (which are analogous to the anisotropic case above), we obtain the ADMM iterations given by

$$\begin{aligned}\Big((J^s)^{j+1}, (\Gamma^s)^{j+1}\Big) &= \operatorname*{argmin}_{J^s,\Gamma^s} \tfrac{2\omega_s\gamma}{(S-1)\nu_j}|\Gamma^s| + \sum_{x\in\Omega\setminus\Gamma^s} \tfrac{\beta^2}{(S-1)\nu_j} P^s(J_x^s) + \tfrac{2+\mu_j S(S-1)}{\nu_j S(S-1)}\Big\|u^s - (\overline{u}^s)^j\Big\|^2 \\ &\qquad + \Big\|a^s - (\overline{a}^s)^j\Big\|^2 + \Big\|b^s - (\overline{b}^s)^j\Big\|^2, \quad \text{for all } s = 1,\dots,S = 4, \\ (\tau_u^{s,t})^{j+1} &= (\tau_u^{s,t})^j + \mu_j((u^s)^{j+1} - (u^t)^{j+1}), \\ (\tau_a^{s,t})^{j+1} &= (\tau_a^{s,t})^j + \nu_j((a^s)^{j+1} - (a^t)^{j+1}), \\ (\tau_b^{s,t})^{j+1} &= (\tau_b^{s,t})^j + \nu_j((b^s)^{j+1} - (b^t)^{j+1}), \quad \text{for all } 1 \le s < t \le S = 4,\end{aligned} \tag{6.77}$$

where we use the following abbreviations

$$(\overline{u}^s)^j = \frac{2f + \mu_j S\Big(\sum_{t=s+1}^{S}\Big((u^t)^j - \frac{(\tau_u^{s,t})^j}{\mu_j}\Big) + \sum_{r=1}^{s-1}\Big((u^r)^{j+1} + \frac{(\tau_u^{r,s})^{j+1}}{\mu_j}\Big)\Big)}{2 + \mu_j S(S-1)}, \tag{6.78}$$

$$(\overline{a}^s)^j = \tfrac{1}{S-1}\sum_{t=s+1}^{S}\Big((a^t)^j - \tfrac{(\tau_a^{s,t})^j}{\nu_j}\Big) + \tfrac{1}{S-1}\sum_{r=1}^{s-1}\Big((a^r)^{j+1} + \tfrac{(\tau_a^{r,s})^j}{\nu_j}\Big), \tag{6.79}$$

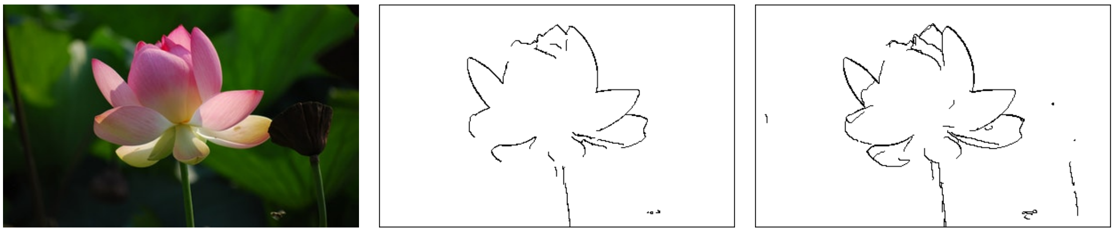
$$(\overline{b}^s)^j = \tfrac{1}{S-1}\sum_{t=s+1}^{S}\Big((b^t)^j - \tfrac{(\tau_b^{s,t})^j}{\nu_j}\Big) + \tfrac{1}{S-1}\sum_{r=1}^{s-1}\Big((b^r)^{j+1} + \tfrac{(\tau_b^{r,s})^j}{\nu_j}\Big). \tag{6.80}$$

The important observation is that each penalty P^s operates exclusively on the paths induced by the direction e_s (cf. Figure 4.2 for an illustration of these paths). Hence, similarly to the anisotropic four-neighborhood discretization (6.62) the subproblems w.r.t. (J^s, Γ^s) decompose into univariate segmented least squares problems along the lines induced by direction e_s. We solve these subproblems as described in Section 6.3.2.

6.3.4. Multi-channel images

We extend the proposed approach to multi-channel images. First, we see that applying the second order Mumford-Shah model (6.2) to multi-channel images $f : \Omega \to \mathbb{R}^L$ is not equivalent to applying the single-channel model for each channel separately. Typically, such a channel-wise approach leads to undesired artifacts as the edges are not enforced to be aligned across

(a) Noisy data (b) Channel-wise (c) Multi-channel



(d) Clean image (e) Edge set of (b) (f) Edge set of (c)

Figure 6.5.: Comparison of channel-wise and multi-channel second order Mumford-Shah models. The channel-wise approach does not enforce the edges to be aligned across the different channels. Hence, the edge set is not always consistent with the produced image (see, e.g., the bottom right petal in (b,c)). The multi-channel approach aligns the edges across the color channels and can preserve more details (see, e.g., the upper part of the blossom in (e,f)).

the channels. We refer to our experiments for a visual impression (see Figure 6.5, Section 6.4), where we compare the channel-wise approach with the one proposed next.

Jet formulation for multi-channel images. As a first step, we generalize (2.17) and define the Taylor jet of a vector-valued function $u : \Omega \to \mathbb{R}^L$ by

$$Ju(x) = \left(Ju_1(x), \ldots, Ju_L(x)\right)^T = \left(J_x^1, \ldots, J_x^L\right)^T, \tag{6.81}$$

i.e., it consists of the Taylor expansions of the component functions $u_l : \Omega \to \mathbb{R}$ of u. The multi-channel version of the jet formulation (6.31) is now given by

$$\begin{aligned} &\underset{J,\Gamma}{\operatorname{argmin}} \sum_{l=1}^{L} \left(\int_\Omega |u_l(x) - f_l(x)|^2 \, dx + \beta^2 \int_{\Omega\setminus\Gamma} \mathcal{R}(J_x^l) \, dx \right) + \gamma \operatorname{length}(\Gamma) \\ &\text{subject to} \quad C_a(J_x^l) = 0 \ \text{ and } \ C_b(J_x^l) = 0 \quad \text{for all } x \in \Omega \setminus \Gamma, \ l = 1, \ldots, L, \end{aligned} \tag{6.82}$$

where the minimum is taken over multi-channel first order jets.

Discrete multi-channel problem. The discretization of the edge penalty (in the sense of Section 6.2.2) is the same as in the single-channel case as the domain of the image is unchanged. We apply the discrete second order penalty and the relaxed jet constraints for each channel separately and take the sum. Thus, the multi-channel counterpart of (6.17) is given by

$$P^1_{\lambda,\alpha,\rho}(J_x) = \sum_{j=1}^{L} P^1_{\lambda,\alpha,\rho}(J^j_x) \tag{6.83}$$

and accordingly for P^2, P^3, P^4. In analogy to the single-channel case (6.75), we introduce the jets $J^1, \dots, J^4$ as splitting variables. and obtain the following discrete constrained problem

$$\begin{gathered}\underset{J^1,\dots,J^4,\Gamma}{\operatorname{argmin}} \left\{ F\Big(\{J^s\}_s, \Gamma\Big) = \sum_{s=1}^{4} \left\{ \sum_{l=1}^{L} \tfrac{1}{4}\|u^s_l - f_l\|^2 + \beta^2 \sum_{x \in \Omega_h \setminus \Gamma^s} \tfrac{1}{2} P^s(J^s_x) + \omega_s \gamma \, |\Gamma^s| \right\} \right\} \\ \text{subject to} \quad (J^l_x)^s = (J^l_x)^t \quad \text{for all } x \in \Omega_h, \text{ and } 1 \le s < t \le 4,\ l = 1, \dots, L.\end{gathered} \tag{6.84}$$

The Lagrangian of (6.84) is now understood w.r.t. the Lagrange multipliers $\tau^{s,t}_u, \tau^{s,t}_a, \tau^{s,t}_b \in \mathbb{R}^{m \times n \times L}$ and the squared Frobenius norm $\|u\|^2 = \sum_{i,j,l} u^2_{i,j,l}$. The derivation of the ADMM scheme is analogous to the single-channel case.

Univariate subproblems for multi-channel data. We can apply our solver for the univariate subproblems also to multi-channel data. To this end, we recall the multi-channel jet-based second order penalty (6.83) and observe that the jet approximation errors $\mathcal{E}^I$ in Section 6.3.2 are now given by the channel-wise sum

$$\mathcal{E}^I = \sum_{l=1}^{L} \mathcal{E}^I_l. \tag{6.85}$$

Therefore, we can extend the error update strategy to the multi-channel case by channel-wise updating the approximation errors. We note all channels share the same system matrix A^q which was defined in (6.70). Hence, the recurrence coefficients for the error update scheme have to be computed only once instead of L times. As a direct consequence of (6.85) and Theorem 3.10 we obtain:

Corollary 6.16. *The multi-channel version of the proposed scheme produces an exact solution of the subproblems in* (6.63) *with* $\mathcal{O}(n^2 L)$ *time.*

In view of the linear complexity in the number of channels, the proposed algorithm can be effectively applied to vector-valued images with a high-dimensional codomain.

6.4. Experimental results

In this section, we apply the proposed approach to the second order Mumford-Shah model to numerical data. First, we provide the necessary implementation details. Then, we discuss the influence of the parameters α, λ which arise in the proposed discretization with Taylor jets. In particular, we single out a particular set of parameters which we use in the following. Afterwards, we illustrate the benefits of using the eight-neighborhood discretization and of using the multi-channel model. Then, we compare the proposed jet-based method to a basic splitting approach which uses second order directional differences only. The central topic of this section is a qualitative and quantitative comparison of our algorithmic approach with the methods for the second order Mumford-Shah problem existing in the literature. These approaches are the graduated non-convexity approach of Blake and Zisserman [21] and the Ambrosio-Tortorelli type approach of Zanetti et al. [216, 215]. Finally, we show the potential of the proposed method in connection with edge detection.

Implementation details. We employ the algorithmic scheme (6.77) for the near-isotropic discretization (6.75) if not stated otherwise. The splitting jets are initialized by $u^s = f$, $a^s = 0$, $b^s = 0$ and the multipliers $\tau_u^{s,t}, \tau_a^{s,t}, \tau_b^{s,t}$ are all initialized by zeros. The initial coupling parameters are chosen as $\mu_0 = 10^{-3}$, $\nu_0 = 10^{-2}$. (We recall that the pixel values and the slopes typically live on different scales.) After each iteration, we increase the coupling parameters μ, ν by the factor 1.3 until the splitting jets become (approximately) equal. More precisely, we stopped the iteration when the relative difference between consecutive splitting jets was smaller than 10^{-5}, i.e., when $\|(q^s)^j - (q^{s+1})^j\| / (\|(q^s)^j\| + \|(q^{s+1})^j\|)$, for all $q \in \{u, a, b\}$ became smaller than 10^{-5} for each $1 \leq s < 4$. We point out that we always observed the numerical convergence of the proposed method.

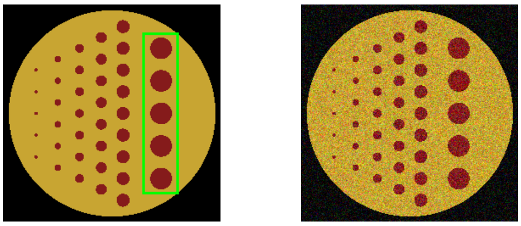
Investigation of discretization parameters. We experimentally study the influence of the parameters α, α' and λ which were introduced in connection with our jet-based discretization of the second order Mumford-Shah functional in Section 6.2.2; see (6.17), (6.18), (6.32), and (6.33). We briefly recall the meaning of these parameters. The parameter $\lambda \in [0, 1]$ balances the second order central differences of the pixel values u and the forward differences of the slope variables a and b, respectively; the positive parameters α, α' couple the slope variables a and b (which represent the derivative part in the Taylor jet of u) and the directional differences of the image u. Motivated by the discrete near-isotropic second order spline-related Mumford-Shah jet model in Section 6.2.2 (cf. (6.40), (6.41)) we let

$$\alpha' = \tfrac{1}{8}\alpha. \tag{6.86}$$

Figure 6.6 shows the results of the proposed near-isotropic jet formulation (6.19) for different values of α and λ. We observe that, for large values of α, the results become rather insensitive to the particular choice of α. Further, for sufficiently large α the results are also quite insensitive to the choice of λ. (This is as expected and confirms the theoretical considerations of Section 6.2.2.) The experiments in Figure 6.6 suggest that choosing α in the order of 10 is sufficiently large in practice and, then, the results are very similar. This motivates to stick to a particular choice of these parameters in the following experiments of this chapter. Concretely, we chose

$$\alpha = 12, \qquad \lambda = 0. \tag{6.87}$$

We observe that this choice is also appealing from a theoretical viewpoint since this choice realizes the spline-based jet distance d_{spline} of Example 6.11 and Section 6.2.3.



(a) Clean image (b) Data

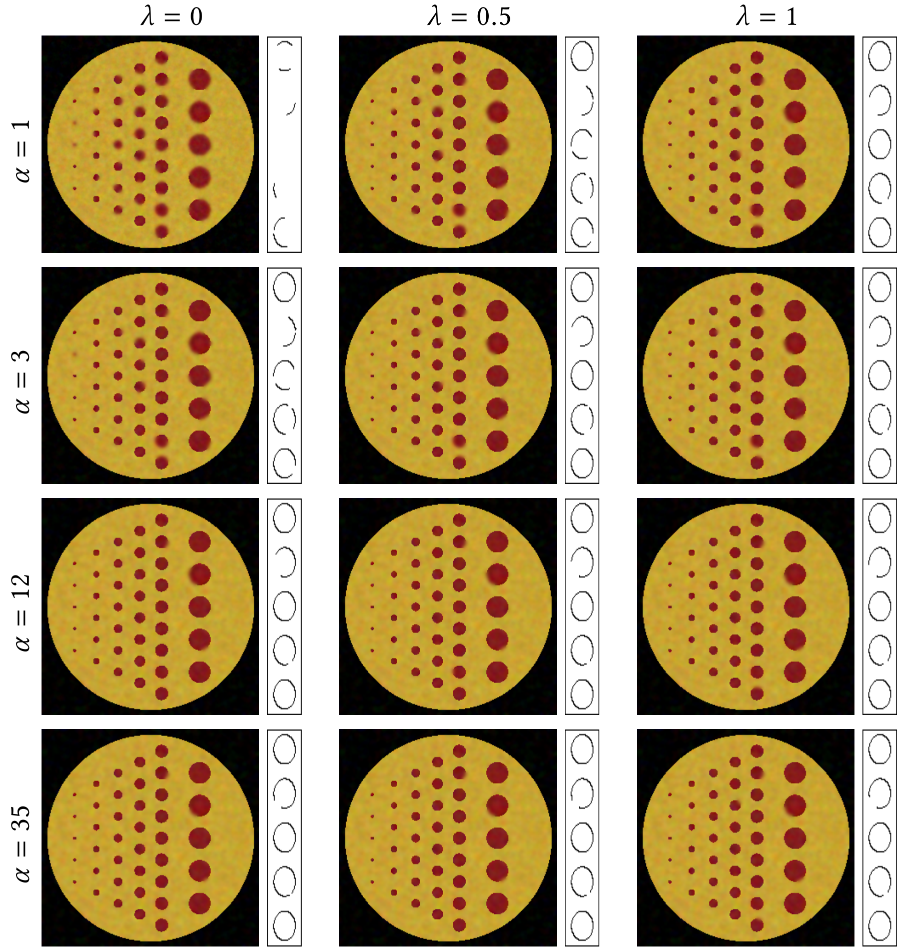


Figure 6.6.: Different choices of jet-related parameters (6.17). We compare the results for different choices of the jet-related parameters λ, α. Generally, we observe that larger choices of the jet-related parameter α lead to more detected edges (see, e.g., the rightmost column of spheres and its corresponding edges in the detail view). Choosing a higher weight λ of the second order differences yield more edges as well. For sufficiently large values of α, say α in the order of 10, these trends saturate. (Note that the point evaluation parameter has been fixed to $\rho = \frac{1}{2}$ for all experiments in this figure.)

Benefits of increasing the isotropy and of coupling the channels. In Section 6.2.2, we have extended the proposed approach to the (near-isotropic) eight-neighborhood discretization to increase the isotropy. In Figure 6.3, we show the benefit of this extension by comparing the results of the proposed approach using the eight-neighborhood discretization with using the four-neighborhood discretization. We observe, that the eight-neighborhood discretization provides smoother boundaries than the four-neighborhood discretization which exhibits "geometric staircasing effects".

In Section 6.3.4, we have extended the proposed approach to deal with multi-channel images, i.e., with vector-valued data. We show the benefit of this extension by comparing it with the channel-wise application of the single-channel approach in Figure 6.5. We observe that in the channel-wise approach the edge set is not always consistent with the produced image whereas, in the multi-channel approach, the produced edges are aligned across the color channels and more details are preserved.

Benefits of the jet-based approach compared with a basic splitting approach. We motivate the proposed jet-based approach by comparing it with a basic splitting approach which uses no jet formulation but second order directional differences only. To this end, we combined the pure use of second order directional differences with an ADMM splitting approach which was proposed for first order Mumford-Shah functionals in [101]. The basic approach uses the following straightforward discretization of the second order Mumford-Shah model

$$\operatorname*{argmin}_{u,\Gamma} \|u - f\|^2 + \sum_{s=1}^{4} \Big\{ \beta^2 \sum_{x \in \Omega_h \setminus \Gamma^s} \tfrac{1}{\|e_s\|^2} |\nabla_s^2 u(x)|^2 + \omega_s \, \gamma |\Gamma^s| \Big\}. \tag{6.88}$$

We employ a discrete eight-neighborhood to measure the edge set (as defined in Section 6.2.2), and, "for the smoothing part", second order differences in coordinate and diagonal directions. We may then proceed similar as in [101] to algorithmically approach (6.88). In particular, we may follow the derivation of the ADMM splitting scheme there to arrive at [101, Eq. (16)]. In contrast to the univariate first order Mumford-Shah problems appearing in [101, Eq. (16)], we here have to solve the corresponding second order univariate Mumford-Shah problems. We use Algorithm 3.1 to solve them. We call the just presented approach the *basic approach.*

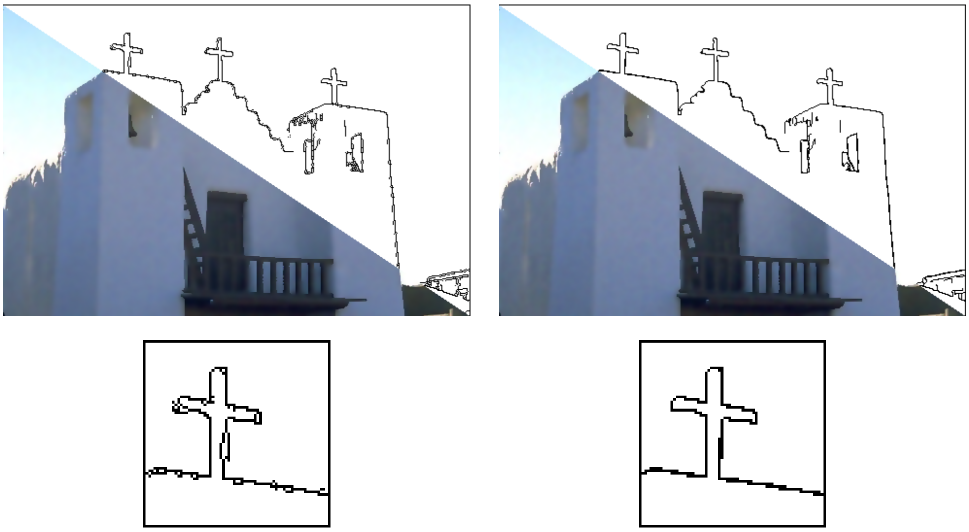
In Figure 6.7, we see the practical benefit of the proposed jet-based approach compared with the basic approach: the edges produced by the basic approach are prone to clutter, while the edges produced by the proposed jet-based approach are smoother and show less clutter.

Remark 6.17. It is necessary to employ the diagonal directions in (6.88): the discrete model which uses horizontal and vertical second order differences only, i.e.,

$$\operatorname*{argmin}_{u,\Gamma} \|u - f\|^2 + \sum_{x \in \Omega_h \setminus \Gamma^1} \beta^2 |\nabla_1^2 u(x)|^2 + \sum_{x \in \Omega_h \setminus \Gamma^2} \beta^2 |\nabla_2^2 u(x)|^2 + \gamma |\Gamma^1| + \gamma |\Gamma^2|, \tag{6.89}$$

would not yield a reasonable anisotropic discrete variant of the second order Mumford-Shah model (6.2). Indeed, the discrete smoothing term in (6.89) $|\nabla_1^2 u(x)|^2 + |\nabla_2^2 u(x)|^2$ does not reflect the mixed derivatives part $\partial_1 \partial_2 u(x)$. However, they are present in (6.2) via the Frobenius norm

(a) Clean image [4] (b) Noisy data, $\sigma = 0.03$



(c) Second order differences discretization (d) Proposed jet-based discretization

Figure 6.7.: Comparison of the basic second order differences approach and the proposed jet-based approach. The edge set produced by the basic approach based on second order differences exhibits clutter (see, e.g., the left cross in the detail view and the roof). The proposed jet-based approach produces a smoother and more coherent edge set.

of the Hessian of u. This lack is rather severe as the respective kernels differ significantly. (We note that (6.88) uses second order directional differences only as well. However, the mixed derivatives part in (6.2) is reflected here which can be seen by the bilinearity of the second derivative.)

6.4.1. Comparison with the graduated non-convexity approach and the Ambrosio-Tortorelli approach

We compare the proposed approach to the existing approaches to second order Mumford-Shah models. These approaches are the graduated non-convexity (GNC) approach proposed by Blake and Zisserman [21], as well as the Ambrosio-Tortorelli type approach by Zanetti et al. [2, 216, 215]. We perform a qualitative and a quantitative evaluation.

The graduated non-convexity approach. In [21], Blake and Zisserman proposed the so-called weak plate model. It models both jumps denoted by Γ_s (called steps) and kinks denoted by Γ_c in the image (called creases). For the squared Frobenius norm of the Hessian the model energy is given by

$$\int_\Omega |u(x) - f(x)|^2 \, dx + \beta^2 \int_{\Omega \setminus (\Gamma_c \cup \Gamma_s)} \|\nabla^2 u(x)\|^2 \, dx + \gamma_s |\Gamma_s| + \gamma_c |\Gamma_c|, \tag{6.90}$$

where $\gamma_s \geq 0$ and $\gamma_c \geq 0$ denote the jump and kink penalty, respectively, while β denotes the smoothing penalty. The discrete counterpart is formulated in terms of a potential and is given by

$$F(u) = \|u - f\|^2 + \sum_{x \in \Omega'} g_{\gamma,\beta}\left(\sqrt{V_x}\right). \tag{6.91}$$

Here, the truncated quadratic potential $g_{\gamma,\beta}$ is given by $g_{\gamma,\beta}(t) = \min\left\{\beta^2 t^2, \gamma\right\}$. and for $x \in \Omega'$, the quadratic variation of u in x is given by

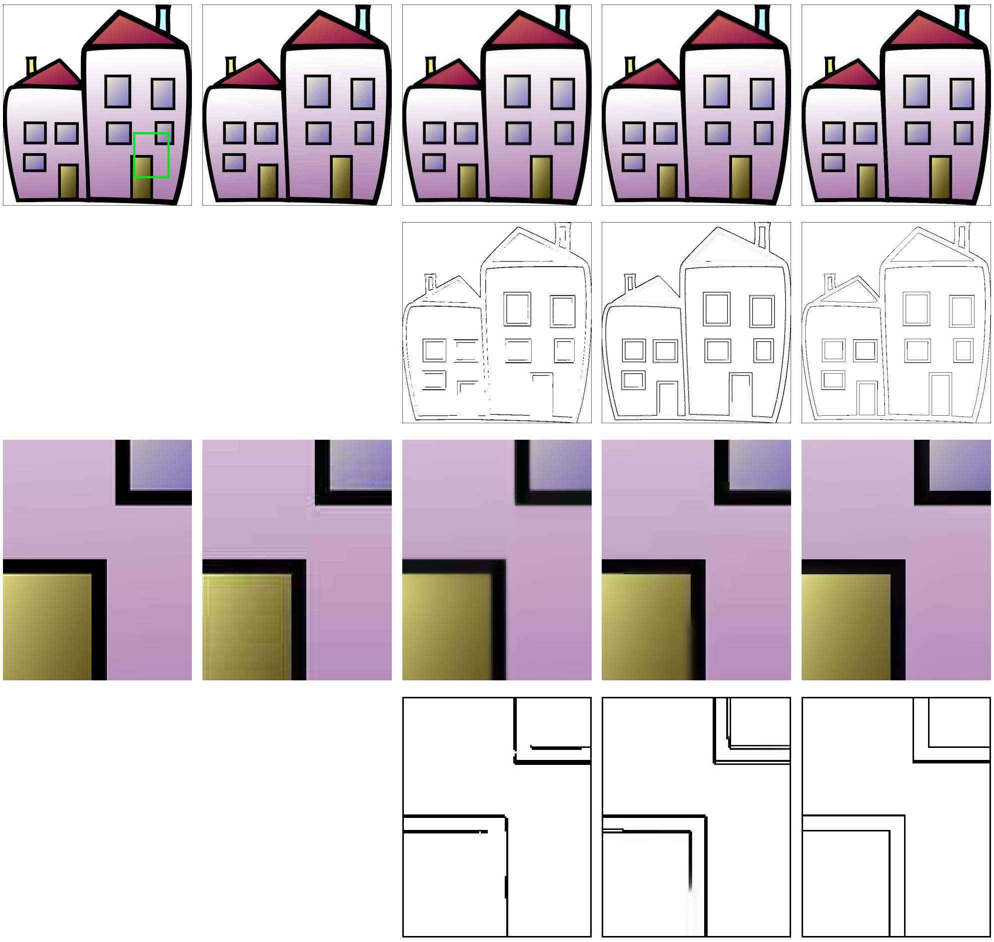
$$V_x = |\nabla_1^2 u(x)|^2 + |\nabla_2^2 u(x)|^2 + 2|\nabla_1 \nabla_2 u(x)|^2, \tag{6.92}$$

where the last summand $\nabla_1 \nabla_2 u(x) = u(x + e_2) + u(x - e_1) - u(x) - u(x + e_1 + e_2)$ denotes a discretization of the mixed partial derivative. (Recall that ∇_1^2 and ∇_2^2 denote the second order finite differences in horizontal and vertical direction, respectively.) Blake and Zisserman interpret the points with $g_{\gamma,\beta}(\sqrt{V_x}) = \gamma$ as edge set and distinguish between creases and jumps where the latter appear as contours which are two elements wide. (In other words, a jump amounts to two neighboring kinks perpendicular to the edge.)

Blake and Zisserman proposed a graduated non-convexity approach (GNC) to (6.91). The central idea is to solve a sequence of minimization problems which are easier to handle than (6.91). Initially, the non-convex energy F is approximated by a convex function F^*. Then a whole sequence of energies $F^{(p)}$ is constructed which depends on the parameter $1 \geq p \geq 0$. In particular, $F^{(1)} = F^*$, $F^{(0)} = F$ and $F^{(p)}$ changes in a continuous way from $F^{(1)}$ to $F^{(0)}$. In other words, a kind of homotopy is constructed. One solves the corresponding minimization problems sequentially for decreasing values of p, $1 \geq p > 0$ and the previous result provides the starting point for the next p. For our implementation, we follow [21] and use the approximating energies $F^{(p)} = \|u - f\|^2 + \sum_{x \in \Omega'} g_{\gamma,\beta}^{(p)}(\sqrt{V_x})$, where the approximation of $g_{\gamma,\beta}$ is given by

$$g_{\gamma,\beta}^{(p)}(t) = \begin{cases} \beta^2 t^2 & \text{if } |t| < \gamma/(\beta^2 r), \\ \gamma - c(|t| - r)^2/2 & \text{if } \gamma/(\beta^2 r) \leq |t| < r, \\ \gamma & \text{if } |t| > r. \end{cases}$$

Here, $r^2 = \gamma(2/c + 1/\beta^2)$ and $c = 1/(32p)$. For each parameter p in the parameter sequence $p \in \{\frac{1}{2^j} : j \in \mathbb{N}_0, \frac{1}{2j} > \frac{1}{\beta}\} \cup \{\frac{1}{\beta}\}$ we use gradient descend to minimize $F^{(p)}$. The result for $F^{(p)}$ is used as input for the next approximating functional $F^{(p/2)}$. For the gradient descent, we employ an Armijo-type line search and stop when the relative change of energy becomes smaller than 10^{-3}. For $p = 1$ we initialize u by the input image f.



(a) Clean image (b) JPEG data (c) GNC (d) Ambrosio-Tortorelli (e) Proposed algorithm

Figure 6.8.: Qualitative comparison of approaches to the second order Mumford-Shah model for JPEG compressed data. After suppressing the JPEG block artifacts, the result of the graduated non convexity approach (GNC) and the result of the Ambrosio-Tortorelli approach do not preserve all of the sharp edges of the original image and they produce relatively thick edges or “doubled” edges (see, e.g., the detail view). The proposed method preserves more sharp edges of the original image and the edges are better localized (see, e.g., the detail view).

The Ambrosio-Tortorelli approach. In [216, 215], Zanetti et al. use a sequence of elliptic functionals to approximate the second order Mumford-Shah model. As the approximation resembles the approximation of the first order Mumford-Shah model proposed by Ambrosio and Tortorelli [3], we refer to it as the Ambrosio-Tortorelli (type) approach. The Ambrosio-Tortorelli type approach corresponds to minimizing one of these approximating functionals which are given by

$$F(u,z) = \mu \int_\Omega |u(x) - f(x)|^2 dx + \int_\Omega z^2 \|\nabla^2 u\|^2 dx + \alpha \mathcal{AT}_\varepsilon(z). \tag{6.93}$$

Here, the edge indicator function z replaces the edge set Γ in the second order Mumford-Shah model (6.2) and the length of the edge set in (6.2) is approximated by the Ambrosio-Tortorelli component

$$\mathcal{AT}_{\varepsilon}(z) = \int_{\Omega} \varepsilon \|\nabla z\|^2 + \frac{1}{4\varepsilon}(z-1)^2 dx, \tag{6.94}$$

where $\varepsilon > 0$ denotes the edge smoothing parameter (which is chosen empirically in practice). An edge at point x is indicated by $z(x) \approx 0$.

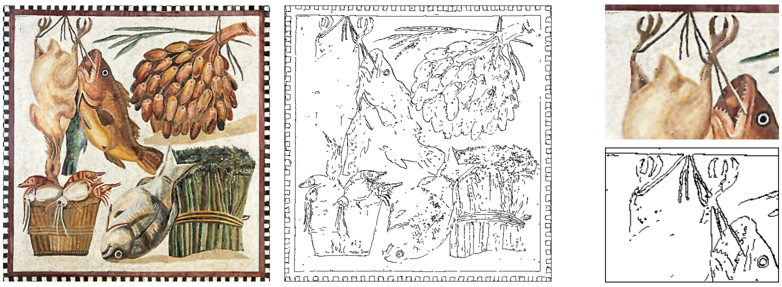
We use the algorithmic approach of [215]. There, the minimization of the discrete objective (by means of forward and central differences) is approached with a vector-valued block coordinate descent as the discrete objective is quadratic in each variable when fixing the other. In each iteration, descent directions for u and z are found by using iterative preconditioned conjugate gradient (pcg). These descent directions are applied to u and z with a constant step-size which exploits that the underlying quadratic problems are symmetric and positive definite. The iterations are stopped when the relative change of the energy becomes lower than 10^{-3} and the maximum number of iterations is 65. The variable u is initialized by the data image f, the edge indicator z by ones and the edge smoothing parameter is set to $\varepsilon = 0.01$ as in [215].

Qualitative comparison. We investigate the qualitative effects of the different approaches to the second order Mumford-Shah model. We first consider a clip art image which was corrupted by a lossy JPEG compression/decompression. A typical effect of JPEG compression are the creation of block artifacts and Gibbs effects near edges; see Figure 6.8. The original image has only constant regions and regions with a color gradient. Thus, the second order Mumford-Shah model is a reasonable model for restoring the image. (Note that there is a series of other variational models for removing JPEG artifacts, for example [208, 29, 56, 154, 30, 31, 158].) The purpose of the present experiment is to visualize the effects of the different approaches to the second order Mumford-Shah model (rather than proposing a new model for JPEG artifact removal). We chose the respective model parameters such that the compression artifacts are removed, while as many edges as possible are preserved. We observe that the proposed method preserves more edges of the original image than the GNC approach and the Ambrosio-Tortorelli approach (see, e.g., the detail view or the house roofs in Figure 6.8).

Next we consider a mosaic image which has –literally by construction– a block pattern in terms of the fine-grained structure given by the single mosaic stones; see Figure 6.9. Similarly to the previous experiment, we want to smooth out these block patterns, while preserving the edges of the depicted scene. In this experiment –for better comparability– we use a fixed smoothing parameter for all methods and then adjust the edge penalty. The smoothing penalty μ of the Ambrosio-Tortorelli type approach and the smoothing parameter β of the GNC approach and of the proposed approach are all set to one. The remaining respective model parameters are chosen to smooth out the mosaic structure in the background. In Figure 6.9, we observe that the edges produced by the GNC approach frequently have breaks and they are often thicker than the proposed edge set. We recall that the Ambrosio-Tortorelli approach produces an edge indicator function z rather than a binary edge map. There are points where the edge indicator function satisfies $z(x) < 1$ that have no counterpart in the form of an abrupt change in the produced image (e.g., the left side of the chicken and the front of the basket).

(a) Input (b) GNC (c) Edges of (b) (d) Details of (b) and (c)

(e) Ambrosio-Tortorelli (f) Edges of (e) (g) Details of (e) and (f)



(h) Proposed algorithm (i) Edges of (h) (j) Details of (h) and (i)

Figure 6.9.: Qualitative comparison of approaches to the second order Mumford-Shah model for the Roman food mosaic image. The edge set of the GNC approach frequently shows “dashed” outlines (see, e.g., the fishes and the basket) and is often thicker than the proposed edge set (see, e.g., the horizontal edge above the chicken or the eye of the brown fish). The edge indicator function of the Ambrosio-Tortorelli type approach yields grayish contours that have no counterpart in the sense of an abrupt change in the produced image (e.g., the left side of the chicken and the front of the basket). Further, values that indicate an edge are often spread across several pixels (e.g., the cords in the upper left corner). The edge set produced by the proposed method is better localized and corresponds to the locations of abrupt changes in the produced image.

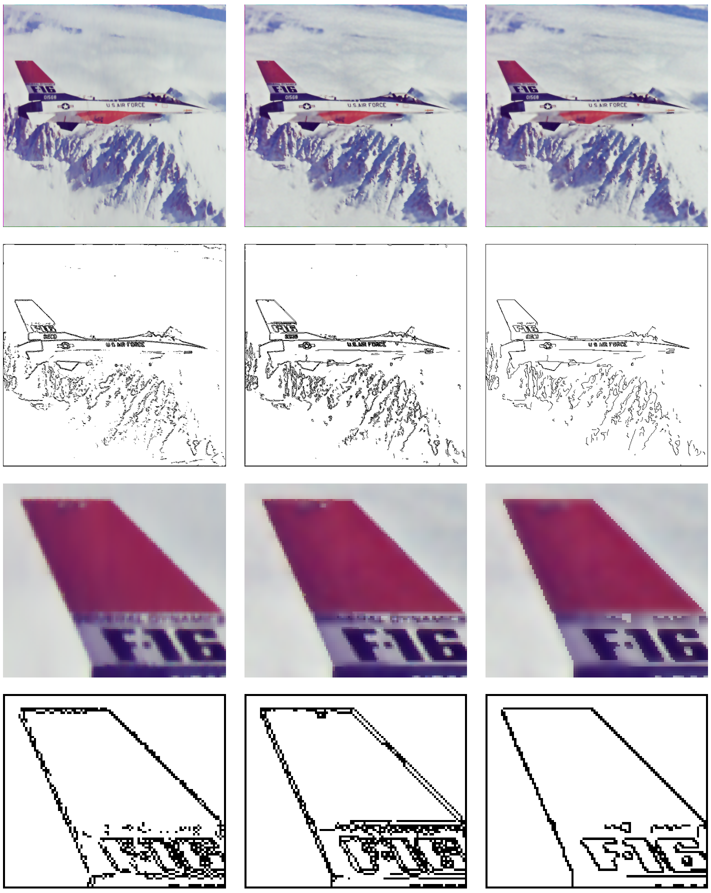
γ	β	GNC	AT	Prop.
0.025	0.5	1460.1	1228.6	**1064.2**
	1	2732.4	1944.2	**1474.0**
	2	5409.6	2837.3	**1783.0**
	3	8729.8	3519.0	**1918.2**
	5	17148.0	4787.3	**2049.3**
	8	32323.7	6874.9	**2141.1**
0.035	0.5	1493.6	1312.9	**1138.2**
	1	2842.9	2136.9	**1624.0**
	2	5815.9	3176.4	**2014.3**
	3	9536.4	3955.2	**2193.2**
	5	17863.2	5288.9	**2371.2**
	8	33581.0	7280.7	**2498.8**
0.05	0.5	1500.0	1376.5	**1200.4**
	1	2878.4	2301.4	**1760.9**
	2	5877.3	3518.4	**2252.1**
	3	9630.9	4433.9	**2486.1**
	5	19677.9	5959.5	**2728.3**
	8	34300.2	7950.1	**2904.8**
0.1	0.5	1480.5	1390.9	**1262.2**
	1	2767.1	2377.8	**1928.3**
	2	5790.6	3858.3	**2595.0**
	3	10317.9	5111.4	**2953.7**
	5	21490.8	7229.5	**3354.2**
	8	38044.1	9857.4	**3672.6**

Table 6.1.: Mean values of the proxy energy (6.95) on the IVC dataset [132]. We report the results for the edge penalties γ and the smoothing penalties β given in the first and second column, respectively. The proposed method attains lower mean energies than the GNC method and the AT method.

Furthermore, the edge indicator function is often spread across several pixels (e.g. the cords in the upper left corner). The edge set of the proposed method is typically sharper localized than those of the other methods (that is, the width of the edges is smaller, typically only one or two pixels). At the same time, the edge set is binary by construction. Consequently, no binarization step is necessary as in the Ambrosio-Tortorelli approach, and the edge set suffers less from broken edges than the GNC approach.

Quantitative comparison. We start the quantitative comparison with a comparison in terms of model energy. Regarding the energy used for the comparison, we observe the following: we can not use the discrete energy (6.19) since its evaluation requires a discrete jet field and the methods of comparison (the GNC approach and the Ambrosio-Tortorelli approach) do not provide jet fields. Employing the weak plate energy (6.91) is not appropriate since it uses different penalties for the kinks and jumps which is not reflected in the considered continuous model (6.2). The Ambrosio-Tortorelli approximation yields a whole family of elliptic energies depend-

(a) Input



(b) GNC, $E = 3989.8$ (c) AT, $E = 2058.3$ (d) Prop., $E = 1497.8$

Figure 6.10.: Results and energies for the “avion” image from the IVC dataset [132] with $\gamma = 0.05$ and $\beta = 2$. The detail images reveal that GNC and AT tend to produce duplicate edges near object boundaries (see, e.g., the right boundary of the rear wing). The proposed approach (Prop.) yields sharp edges and achieves the lowest energy value. The mean energies on the IVC dataset for additional values of γ, β are reported in Table 6.1.

ing on the parameter ε in the Ambrosio-Tortorelli component of the energy (6.94) and there is no reasonable means to distinguish one of them with respect to which one should compare. A reasonable discrete (proxy) energy for which all algorithms provide the necessary input is given by

$$E(u,\Gamma) = \|u-f\|^2 + \beta^2 C(Hu,\Gamma) + \gamma\,\ell(\Gamma). \tag{6.95}$$

Here, $\ell(\Gamma) = |\Gamma^1| + |\Gamma^2|$ measures the length of the horizontal and vertical jump sets. We measure the edge length in this anisotropic way here due to simplicity and because we can provide the necessary information for all methods of comparison. (In connection with this we point out that the Ambrosio-Tortorelli approach actually yields an edge indicator function. We derive an edge set from it as explained below. To keep this derivation transparent and simple is a particular reason to consider the edge penalty $\ell(\Gamma)$ above.) For the smoothing term $C(Hu,\Gamma)$, we use the sum of the squared Frobenius norms of the discrete Hessian Hu and exclude those entries of Hu for which the stencils of the underlying finite differences in u meet the edge set Γ. (Here, $Hu(x)$ is defined in terms of the second order differences in coordinate directions and the mixed second order differences given by $\nabla_2\nabla_1 u(x) = u(x+e_1+e_2) - u(x+e_2) - u(x+e_1) + u(x)$.)

We note that (6.95) yields a discretization of the second order Mumford-Shah model in its formulation (6.2) (with the squared Frobenius norm of the Hessian discretized in terms of finite differences of u only). In view of this, the energy (6.95) and the proposed jet-based energy (6.19) are related as two discretizations of the two equivalent formulations of the second order Mumford-Shah model (6.2) and (6.15). However, the energy (6.95) does not require a jet to be evaluated. Its evaluation only requires a piecewise smoothed image u and a corresponding discrete edge set $\Gamma = \Gamma^1 \cup \Gamma^2$ as arguments.

For the comparison we identify the parameters α, μ of the Ambrosio-Tortorelli type approach (6.93) with the parameters of (6.95). The smoothing parameter μ in the Ambrosio-Tortorelli formulation directly corresponds to $\frac{1}{\beta^2}$ in (6.95). Therefore, we set $\mu = \frac{1}{\beta^2}$. For the edge penalty, we let $\alpha = \gamma/(2\varepsilon\beta^2 + \frac{1}{2\varepsilon}\beta^2)$ which is motivated as follows: we consider a binary image with a single vertical edge. For a given edge parameter γ, the corresponding edge penalty should be γ times the height of the image. We require that the Ambrosio-Tortorelli component (which accounts for the edge part) attains the same value. To evaluate the Ambrosio-Tortorelli functional we need an edge indicator function z. We make the idealized assumption that the edge indicator function is binary. Then, for sufficiently large β, the edge indicator function equals zero in a two pixel wide tube along the edge and is one elsewhere. Further, for sufficiently large β, the only nonzero contribution to the Ambrosio-Tortorelli functional value is given by the Ambrosio-Tortorelli component. Thus, the total function value is given by $2\alpha\beta^2(\varepsilon + \frac{1}{4\varepsilon})$ times the height of the image. Setting this term equal to γ times the height of the image yields $\alpha = \gamma/(2\varepsilon\beta^2 + \frac{1}{2\varepsilon}\beta^2)$.

In order to evaluate the proxy energy (6.95) for the output of the Ambrosio-Tortorelli approach, we need to derive an edge set Γ from the edge indicator function z. To this end, we employ the following procedure. We first threshold the edge indicator function by $z \le \frac{1}{2}$. Among the remaining candidates, we then choose the ones that realize the smallest row-wise and column-wise univariate energy given by

$$\beta^2|\nabla_1^2 u(x_{j,:})|^2 + \beta^2|\nabla_2\nabla_1 u(x_{j,:})|^2 + \gamma|\Gamma^1_{j,:}| \quad \text{and} \quad \beta^2|\nabla_2^2 u(x_{:,i})|^2 + \beta^2|\nabla_2\nabla_1 u(x_{:,i})|^2 + \gamma|\Gamma^2_{:,i}|,$$

respectively. (Please note that the finite differences are not taken across Γ^1 and Γ^2, respectively.) We point out that, pure thresholding often produces "thicker" edges than this procedure. Typically, this would result in higher functional values. For the GNC method, the candidates for the edges are those pixels x for which the corresponding quadratic potential in (6.91) satisfies $g_{\gamma,\beta}(V_x) = \gamma$. The edge thinning is done as for the Ambrosio-Tortorelli approach.

To compare the energy values of the three methods, we use the IVC dataset [132] which consists of 10 natural color images of size 512×512. Each image is slightly corrupted by Gaussian noise with standard deviation $\sigma = 0.02$. First, we discuss the choice of the parameters γ and β in (6.95). If $\gamma \to 0$, the results will (approximately) reproduce the data f by introducing edges for nearly all pixels which is typically not desired. We consider a fixed (and not too small) γ. If the parameter β is too small relative to γ the produced results only have a few edges and the results correspond (approximately) to Tikhonov regularized solutions. (For fixed γ, and $\beta \to 0$ we indeed obtain the Tikhonov solution which can be confirmed analytically.) On the other hand, for parameters β which are too large relative to γ, the solutions become approximately piecewise affine-linear. (In the limit $\beta \to \infty$, the solutions are indeed piecewise affine-linear. We develop an algorithm for that problem, i.e., the affine-linear Potts model, in Chapter 7.) Further, for larger values of β there are some issues with the methods of comparison. The computation of descent directions in the Ambrosio-Tortorelli scheme for u,z becomes difficult since the underlying linear systems become ill-conditioned for large β and moderate γ. The practical consequences are too many edges in the produced results. For the GNC approach, the initial convex relaxation is less tight for larger values of β; see [21]. We observe that the results of the GNC method are not sufficiently smoothed and also exhibit too many edges for larger values of β. Consequently, we choose the range of γ and β such that we obtain meaningful results for the IVC dataset. In particular, γ is not too small to reproduce the data, β is in an intermediate range relative to γ, and β is small enough such that the methods of comparison still produce reasonable results.

In Table 6.1, we report the mean energies of each method for the parameters

$$\gamma \in \{0.025, 0.035, 0.05, 0.1\} \quad \text{and} \quad \beta \in \{0.5, 1, 2, 3, 5, 8\}.$$

(See Figure 6.10 for example results.) The proposed method yields lower mean energies than the other methods which illustrates the effectiveness of the proposed algorithmic approach (6.62). In particular, for larger smoothing parameters, the energies of the proposed method are notably lower. Although a sufficiently large choice of β for approximating the piecewise affine-linear case ($\beta \approx 500$) is about a magnitude larger than $\beta = 8$, the GNC and the AT approach become already problematic: the approaches produce too many edges (“clutter”). Further, the GNC results are not sufficiently smooth within segments. This is reflected by the large energy values attained by AT and GNC for $\beta \geq 3$ reported in Table 6.1. The proposed method remains competitive also for large choices of β.

We next give a quantitative comparison in terms of the denoising performance. To this end, we consider the CSIQ image dataset [130] which consists of 30 natural color images of size 512×512. We corrupt each image by additive Gaussian noise with standard deviation $\sigma = 0.1$. The model parameters were chosen to provide the best MSSIM values (cf. Section 2.4 for details concerning the MSSIM). In particular, for the GNC approach, we chose the best parameters from $\gamma \in \{0.01, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25\}$ and $\beta \in \{0.5, 1, 2, 5, 10\}$. For the Ambrosio-Tortorelli type approach (6.93), we let $\mu \in \{10^k, 5 \cdot 10^k : k = -3, \ldots, 0\}$

	MSSIM			
Image	Noisy image	GNC result	AT result	Prop. result
1600	0.695	*0.876*	0.865	**0.877**
aerial-city	0.493	*0.805*	0.800	**0.812**
boston	0.465	*0.751*	0.745	**0.756**
bridge	0.523	*0.830*	0.821	**0.835**
butter-flower	0.372	0.654	*0.658*	**0.666**
cactus	0.753	0.891	*0.894*	**0.897**
child-swimming	0.648	*0.839*	0.826	**0.841**
couple	0.700	*0.913*	*0.913*	**0.915**
elk	0.612	**0.869**	0.866	**0.869**
family	0.482	*0.732*	*0.732*	**0.740**
fisher	0.627	0.935	*0.936*	**0.939**
foxy	0.728	*0.887*	0.880	**0.891**
geckos	0.682	0.846	**0.861**	*0.852*
lady-liberty	0.552	*0.953*	0.947	**0.957**
lake	0.656	0.768	**0.779**	*0.769*
log-seaside	0.702	*0.864*	0.852	**0.865**
monument	0.478	**0.777**	0.764	*0.775*
native-american	0.570	**0.872**	0.866	*0.871*
redwood	0.448	**0.733**	0.720	*0.731*
roping	0.463	*0.644*	0.642	**0.645**
rushmore	0.674	**0.825**	0.819	**0.825**
shroom	0.625	0.870	*0.885*	**0.886**
snow-leaves	0.460	*0.667*	0.663	**0.672**
sunset-sparrow	0.790	**0.941**	0.938	**0.941**
sunsetcolor	0.417	0.852	*0.859*	**0.864**
swarm	0.462	**0.917**	0.901	*0.913*
trolley	0.646	*0.825*	0.818	**0.844**
turtle	0.608	0.910	*0.911*	**0.914**
veggies	0.524	*0.672*	0.666	**0.673**
woman	0.535	**0.827**	0.823	**0.827**

Table 6.2.: Quantitative comparison of the denoising performance on the CSIQ dataset. We report the MSSIM values for each test image. The respective model parameters were chosen optimally w.r.t. the MSSIM. The clean images were corrupted by additive Gaussian noise with $\sigma = 0.1$. We compare the proposed approach (abbreviated by Prop. above) with the graduated non-convexity approach (GNC) and the Ambrosio-Tortorelli approach to the second order Mumford-Shah model (AT).

and $\alpha \in \{10^k, 5 \cdot 10^k : k = -5, \dots, 0\}$ similarly to [215]. For the proposed method, we let $\gamma \in \{0.04, 0.05, \dots, 0.1\}$ and $\beta \in \{1, 2, 3, 5, 15\}$. We report the results in Table 6.2 and note that for most of the images (24 out of 30), the proposed method achieves the best MSSIM. In Figure 6.11, we show example results.

(a) Noisy "sunsetcolor" image, MSSIM=0.417

(b) GNC, MSSIM=0.852

(c) Ambrosio-Tortorelli, MSSIM=0.859

(d) Proposed, MSSIM=0.864

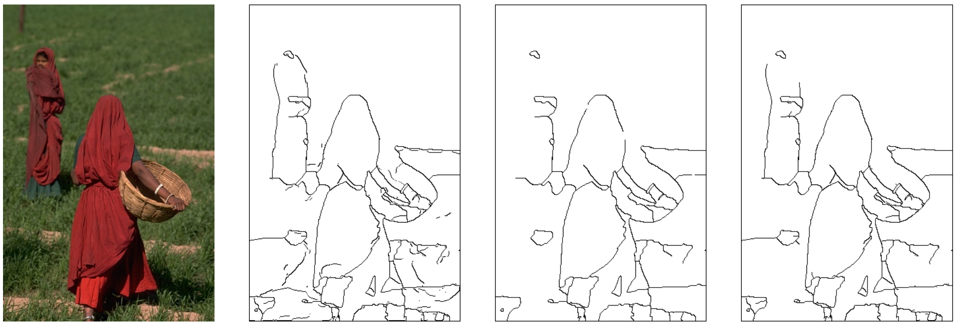
Figure 6.11.: An example from the CSIQ dataset. All three approaches to the second order Mumford-Shah problem give a reasonable denoising result, yet, the result of the proposed method has a slightly higher MSSIM than the results of the other approaches (which is representative for the CSIQ dataset, see Table 6.2). Qualitatively, we observe that the proposed approach produces more sharp object boundaries and preserves more small scale details (see, e.g., the detail view).

6.4.2. Edge detection

We here investigate the potential of the proposed method in connection with edge detection. More precisely, we employ the output of the proposed approach, and in particular the derived edge set, to obtain *strong edge sets* and *weak edge sets*. Then, a hysteresis step is performed to obtain the final result. The idea of strong and weak edge sets in connection with hysteresis thresholding was already used by the classical Canny edge detector [38]. In the Canny edge detector, a pixel belongs to the final edge set if it either belongs to the strong edge set or to a connection component of the weak edge set which contains at least one strong edge pixel. Two thresholds on the gradient magnitude are employed to determine the weak and strong edges: after a non-maximum suppression, pixels exceeding the upper threshold belong to the strong edges, whereas the ones with values between the lower threshold and the upper threshold belong to the weak edges.

We here employ a different notion of strong and weak edge sets based on the result of the proposed approach to the second order Mumford-Shah model. In particular, the weak and strong edges are determined based on the regularized edge set Γ and the regularized image u. Basically, if there are indications for an edge from both outputs Γ and u of the Mumford-Shah regularization, then the corresponding pixel belongs to the strong edge set. If there are only indications by one of the outputs Γ and u, it is part of the weak edge set. More precisely, the procedure is as follows.

Procedure 6.1. *1.* Obtaining indications for an edge based on u. *(i) For each pixel $x \in \Omega'$, determine the discrete gradient $Du(x) = (\nabla_1 u(x), \nabla_2 u(x))$ of u and determine its magnitude*



(a) Input (b) Weak edge set (c) Strong edge set (d) Final result after the hysteresis step

Figure 6.12.: Illustration of the steps of the proposed edge detection procedure. The weak edge set contains the significant edges (see, e.g., the contours of the girl on the left), but also spurious edges (see, e.g., the foreground). The strong edge set shows less spurious edges, but misses some important edges (see, e.g., the girl on the left). After the hysteresis step, the final result complements the strong edge set by including missing edges, while having less spurious edges than the weak edge set.

$G(x) = \|Du(x)\|_2$. *(ii) Thresholding: if* $G(x) > \eta$ *for the threshold* η*, keep those* x *in the candidate set. (iii) Edge thinning via non-maximum suppression: compare the gradient magnitude at the candidate* x *with the two neighbors w.r.t. the orientation* t *of the gradient at* x *and let* $x \in C_u$ *if it yields the largest gradient magnitude. More formally,*

$$C_u = \left\{x : \|G(x)\|_2 > \eta \text{ and } \|G(x)\|_2 \geq \max\left(\|G(x+t)\|, \|G(x-t)\|\right)\right\}.$$

2. Computing the weak edge set. *We define the weak edge set as the union of the output* $\Gamma' = \Gamma^1 \cup \Gamma^2$ *of horizontal and vertical edges of the Mumford-Shah regularization model with the candidate set* C_u *from the previous step, i.e.,*

$$E_{weak} = \Gamma' \cup C_u.$$

3. Computing the strong edge set. *Get the strong edge set by thresholding* Γ'*, i.e.,*

$$E_{strong} = \Gamma' \cap \{x : \|G(x)\|_2 > \eta\}.$$

(We notice that Γ' *is already thin and no thinning is necessary here.)*

4. Hysteresis step. *The final result E is the union of the strong edge set and those weak edges which are connected to the strong edge set (w.r.t. the eight-neighborhood).*

In Figure 6.12, we give an illustration of the weak edge set, the strong edge set and the hysteresis step.

We compare our approach with Canny edge detection [38]. There, the image is first filtered with the (sampled) directional derivatives of a Gaussian. This amounts to first smoothing and then passing to the gradient image. After performing non-maximum suppression, a hysteresis step is performed on the resulting gradient image. This step employs a strong and a weak edge

	F-score (ODS)	F-score (OIS)
Canny edge detector	0.583	0.633
Proposed method	0.624	0.664

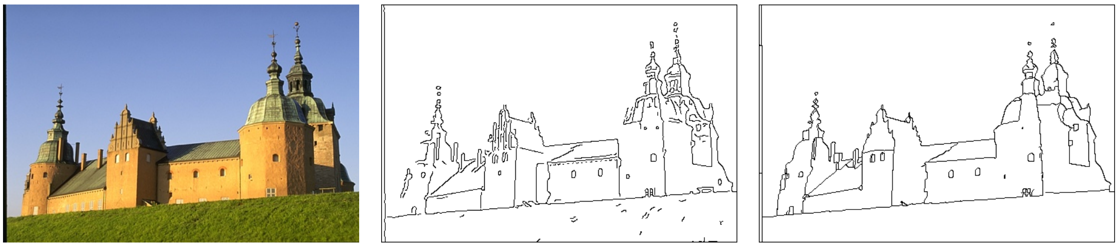
Table 6.3.: Quantitative comparison on the BSD500 test set. We report the global F-scores of the detected edges for the Berkley test set for parameter values optimal on dataset scale (ODS) and optimally chosen on image scale (OIS). The proposed method yields higher global F-scores on both dataset scale and image scale compared with Canny edge detection.

set which are derived by two thresholds as already explained above. (We note that this is in contrast to the proposed scheme which only employs a single threshold η.)

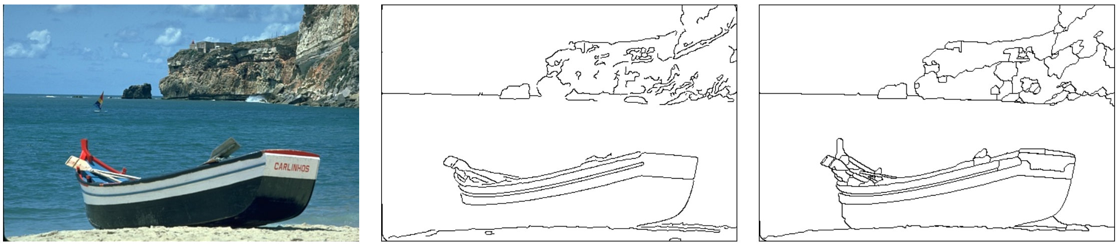
We perform a quantitative comparison of the proposed method and Canny edge detection for the 200 test images of the Berkley segmentation dataset (BSD500) [4]. The quality of the results of the proposed scheme and of Canny edge detection is measured by their maximum F-score. The F-score is defined as the harmonic mean of precision and recall, i.e., $\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$ w.r.t. the provided human annotated groundtruth. *Precision* rates the amount of correctly detected edges among all detected edges and *Recall* rates the amount of correctly recovered edges among all ground truth edges. The F-score is bounded by 0 and 1, where high values are preferable. We use the Matlab implementation of the Canny edge detector. We tune the thresholds by choosing them optimally w.r.t. the F-score among $\eta_1 \in \{0.01, 0.02, \ldots, 1\}$ for the high threshold η_1 and $\eta_2 = \kappa\eta_1$ with $\kappa \in \{0, 0.1, \ldots, 0.9\}$ for the low threshold η_2. The standard deviation of the Gaussian in the smoothing step is chosen as the standard Matlab parameter which is given by $\sqrt{2}$. For the proposed method, we let $\beta = 150$ be fixed, the edge penalty be chosen from $\gamma \in \{0.01, 0.02 \ldots 0.1, 0.125, 0.15, 0.2, 0.25, 0.3\}$ and the threshold from $\eta \in \{0, 0.001, \ldots, 0.01\} \cup \{0.015, 0.02, \ldots, 0.2\} \cup \{0.25, 0.3, \ldots, 1\}$. In Table 6.3, we report the F-scores of Canny edge detection and the proposed method. In particular, we report the F-scores for the optimal parameters on a dataset scale (ODS) and for the optimally chosen parameters for each image separately (OIS). The proposed scheme achieves higher F-scores on both dataset scale and image scale. As illustration, we show example results in Figure 6.13. In our experiments, we observe that the strong edge set of the proposed method contains the prominent edges of the input image which are then complemented by the weak edge set via the hysteresis step such that spurious edges are suppressed (see also Figure 6.12).

6.5. Summary of the chapter

In this chapter, we developed an algorithmic framework for second order Mumford-Shah models. As a first step, we reformulated second (and higher) order Mumford-Shah models equivalently in terms of Taylor jets. After discretizing this jet-based formulation, we obtained a family of discrete second order models. We proved that these discrete models have a minimizer. Towards a numerical approach, we then derived an appropriate ADMM splitting. This splitting was designed in a way which allowed us to solve all subproblems exactly and non-iteratively. These subproblems do not correspond to solving standard univariate (second order) Mumford-Shah problems, but to *solving univariate segmented least square jet problems.* We derived fast



(a) Input (b) Canny, $F = 0.786$ (c) Proposed method, $F = 0.829$



(d) Input (e) Canny, $F = 0.705$ (f) Proposed method, $F = 0.783$

Figure 6.13.: Examples of edge detection results from the Berkley data set. *From left to right:* input image from the BSD500 data set, results for Canny edge detection, and the results of the proposed scheme. We observe that the edges detected by the proposed method are often more complete than the edges detected by the Canny method (see, e.g., the boundary on the left side of the castle in the first image or the left side of the boat in the second image). This is reflected by the F-values reported below the results which are higher for the proposed method.

solvers for these segmented jet estimation problems using the dynamic programming techniques of Chapter 3. The scheme further takes into account the consistency of the produced jet with the image and realizes the appropriate coupling of the subproblems with regard to the jet formulation. We illustrated the potential of the proposed approach to the second order Mumford-Shah problem in numerical experiments. In particular, we showed its benefit compared to a basic splitting approach which uses second order directional differences only. We performed a qualitative and quantitative comparison with existing methods for this problem. We observed that in most cases the proposed scheme produces slightly better results than these methods. In particular, the edge sets are sharply located and show fewer breaks. Finally, we used the proposed method as a basis for an edge detector.

7. Image Partitioning with the Affine-Linear Potts Model

In this chapter, we consider the affine-linear Potts model for image partitioning. This chapter is based on the publications [117, 119].

Organization of the chapter. In Section 7.1, we give an overview and discuss related work. In Section 7.2, we derive a reformulation of the affine-linear Potts model in terms of Taylor jets and derive a corresponding discrete formulation. In Section 7.3, we develop the proposed algorithm and extend the approach to multi-channel images. In Section 7.4, we apply the proposed algorithm to natural color images. In particular, we compare our approach to the state-of-the-art method based on the iterative application of the graph cut algorithm. Finally, Section 7.5 concludes the chapter with a summary.

7.1. Overview and related work

A central task in computer vision is image partitioning. It describes the task of dividing the domain of an image into regions of homogeneous image characteristic [80, 92]. In a supervised setup, learning based approaches, most notably convolutional neural networks, have become standard. If training data is not sufficiently available, model-based methods will be employed. Classical approaches use k-means clustering, region growing (see e.g. [16]), and the watershed algorithm (see e.g. [17]). A popular variational approach to image partitioning is the Potts model (1.2) which is frequently applied for unsupervised segmentation of color images [53, 27], depth images [27], texture images [113], and medical images [165, 183], to mention only some examples. We recall that the Potts model partitions the image domain into segments of constant color intensity such that the total boundary length is small and the corresponding piecewise constant image is close to the input image.

As already seen in Chapter 3 for univariate signals, the piecewise constant model assumption of the Potts model is often restrictive as many types of images possess linear trends within their segments. Consider for example the sky in a landscape image, or an object with an illumination gradient in a conventional image. In such situations, applying the Potts model can produce extra (spurious) segments at steeper slopes and the results become oversegmented; see Figures 7.1 and 7.2. A natural way to allow for linear trends in an image is to consider a model which allows for affine-linear segments. Taking this into account leads to the *affine-linear Potts model* which for an input image $f : \Omega \to \mathbb{R}$ defined on an open domain $\Omega \subset \mathbb{R}^2$ corresponds

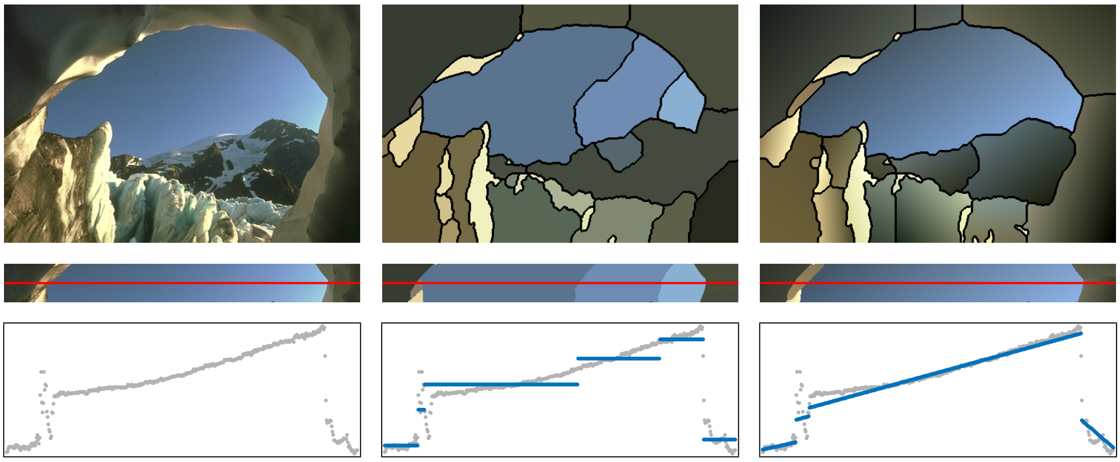

(a) Input image (b) Classical Potts model (c) Affine-linear Potts model

Figure 7.1.: Comparison of the classical (constant) Potts model and the affine-linear Potts model. The classical Potts model oversegments the sky, while the right boundary of the cave is merged with the mountain. The affine-linear Potts model yields an improved partitioning. The scanline plots of the blue channel reveal that the piecewise constant model produces spurious segments to account for the color gradient in the sky; the affine-linear Potts model adjusts to the color gradient.

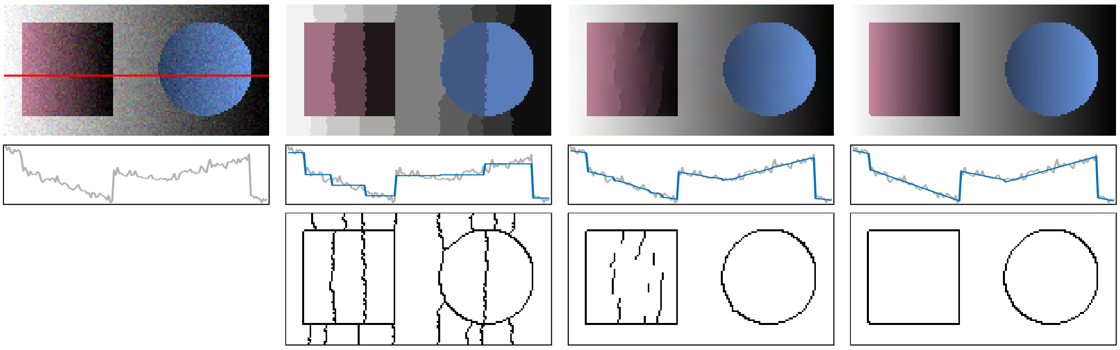
to the minimization problem given by

$$\begin{aligned} &\underset{u,\mathcal{P}}{\operatorname{argmin}} \sum_{P\in\mathcal{P}} \left\{ \int_P |u(x) - f(x)|^2 \, dx + \frac{\gamma}{2} \operatorname{length}(\partial P) \right\}, \\ &\text{subject to } u|_P \text{ is affine-linear for all } P \in \mathcal{P}. \end{aligned} \tag{7.1}$$

The minimum is taken w.r.t. the partitions $\mathcal{P}$ of the image domain Ω and the corresponding piecewise affine-linear functions u on Ω. A partition $\mathcal{P}$ of Ω is understood as a set of pairwise disjoint connected subsets P (the segments) whose union equals Ω. The parameter $\gamma > 0$ in (7.1) controls the tradeoff between the length penalty and the data fidelity. Figure 7.1 illustrates the Potts model and the affine-linear Potts model for a natural image with color gradients. While the Potts model introduces spurious segments to approximate steep color gradients, the affine-linear Potts model recovers the color gradients and provides an improved result.

On the one hand, one might be principally interested in solving (7.1) to find an optimal partition $\mathcal{P}^*$, which may be used for segmentation or superpixelations as for instance in [196, 55]. On the other hand, one might be more interested in finding an optimal piecewise affine-linear function u^*, which may be used for the regularization of flow fields as in [77] or for the guidance of image filters as in [139]. We focus in this chapter on the efficient computation of the partition.

Another possibility to account for color gradients in images is the piecewise smooth Mumford-Shah model (1.1) which penalizes deviations from piecewise constant functions. However, the piecewise smooth model comes with side-effects: the estimated discontinuity set is in general not the boundary set of a partition as it can have open ends (crack tips), and it suffers from



(a) Input (b) Potts model, $\gamma = 0.3$ (c) Mumford-Shah model, $\gamma = 0.03$, $\beta = 10$ (d) Affine-linear Potts model, $\gamma = 0.4$

Figure 7.2.: Comparison of the Potts model, the affine-linear Potts model and the Mumford-Shah model for color gradients. *Top:* solution u^*; *center:* scanlines of the blue channel along the highlighted red line in the input image, *bottom:* discontinuity sets corresponding to u^*. The Potts model oversegments the input image. The Mumford-Shah model produces extra (spurious) discontinuities at steeper slopes ("gradient limit effect") and the discontinuity set is further not the boundary set of a partition (so-called "crack tips"). The affine-linear Potts model recovers the steep gradients and the produced discontinuity set forms a partition.

the gradient limit effect; see Figure 7.2 for an illustration and see Chapter 3 for the univariate situation.

In this chapter, we develop a novel algorithm to efficiently compute approximate solutions of the affine-linear Potts model (7.1). To this end, we reformulate the underlying minimization problem in terms of Taylor jets which eliminates the dependence on the partition $\mathcal{P}$ in (7.1). We propose a novel algorithmic approach based on the ADMM, where all subproblems can be solved exactly and non-iteratively The subproblems decompose into linewise segmented jet estimation problems for which we derive fast solvers using the dynamic programming techniques of Chapter 3.

Related work. For related work regarding the (classical) Potts model (also sometimes called ℓ_0 gradient smoothing), we refer to Section 4.1. Besides the challenges of computing the Potts model (cf. Chapter 4), the affine-linear case comes with an additional complication: a commonly used reformulation, which eliminates the dependence on the partition and which is used in many algorithmic approaches for the Potts model (cf. (4.4)), has no direct analogy for the affine-linear case. In fact, the variety of algorithms is much smaller for the affine-linear Potts model. Existing approaches are based on graduated non-convexity [21], active contours [196], graph cuts [27, 19, 62, 212] and ADMM splitting [77]. We discuss these algorithms in more detail in the next paragraph.

Further related work is the total generalized variation model (TGV) [32] which can be seen as an ℓ_1-based convex relative of the affine-linear Potts model. The results of TGV are approximately piecewise affine-linear. In contrast to the affine-linear Potts model, the solution of TGV does not yield a partition of the image. In [85], the authors apply a piecewise affine-linear image model in the context of the analysis sparsity model in the sense of compressed sensing. We notice that the solutions do not yield a partition directly. We finally mention the related recent work [209] on piecewise affine-linear estimation in connection with deep edge aware filters.

Approaches to the affine-linear Potts model. One can sort the approaches to the affine-linear Potts problem into two classes: those that are mainly interested in finding an approximately optimal partition $\mathcal{P}^*$ and those that are mainly interested in computing an approximately piecewise affine estimate u^*. The latter ones, in general, produce an approximation which does not yield a partition as we will see below.

(i) *Partition-based methods.* The common starting point is the formulation of the affine-linear Potts problem in terms of partitions:

$$\operatorname*{argmin}_{\mathcal{P}} \sum_{P\in\mathcal{P}} \left\{ \min_{a,b,c} \int_P |ax_1 + bx_2 + c - f(x)|^2 \, dx + \frac{\gamma}{2} \operatorname{length}(\partial P) \right\}. \tag{7.2}$$

This formulation gives rise to the following iterative repartitioning scheme

1. Choose an initial partition $\mathcal{P}^0$.
2. Iterate until convergence:
   a. for all $P \in \mathcal{P}^j$ compute the optimal affine coefficients,
   $$(a^*, b^*, c^*) = \operatorname*{argmin}_{a,b,c} \sum_{x\in P} |ax_1 + bx_2 + c - f(x)|^2;$$
   b. compute a partition $\mathcal{P}^{j+1}$ which is (approximately) optimal for the determined set of affine coefficients.

While step 2a is a simple linear regression, step 2b is an NP-hard optimization problem. One possibility to tackle this problem is to represent the partition by level set functions and to evolve them guided by a PDE (active contours approach), see e.g. [196]. Another possibility is to compute the partition by a min-cut/max-flow algorithm (graph cuts approach) which is used for instance in [105, 63, 212]. The graph cut approach is often employed because a highly efficient library exists [27, 26, 127]. We use this method (which uses graph cuts partitioning) as the benchmark method.

(ii) *Function-based methods without explicit partitioning.* In their early work, Blake and Zisserman [21] proposed to use a graduated non-convexity approach (GNC). The first step of GNC is to approximate the non-convex energy function F of the closely related weak plate model by a convex function F^*. Then a whole sequence of energy functions $F^{(p)}$, $1 \geq p \geq 0$ depending on the parameter $1 \geq p \geq 0$ is constructed such that $F^{(1)} = F^*$ and $F^{(0)} = F$ and $F^{(p)}$ changes in a continuous way from $F^{(1)}$ to $F^{(0)}$. One sequentially solves the corresponding minimization problems (e.g. by gradient descent) for $1 \geq p > 0$ and decreasing p using the results of a higher p as starting point for the next lower p. Blake and Zisserman point out that GNC shows good performance for small variation penalties of the weak plate model. Unfortunately, the affine-linear Potts model is the limiting case of the weak plate model for infinitely large variation penalty, and the GNC results are less favorable in this case. Recently, Fortun et al. [77] proposed a method using a splitting into coupled subproblems. The splitting is such that the subproblems boil down to univariate affine-linear Potts problems which can be solved efficiently. We will elaborate on the relations to the proposed method later on (Section 7.3.4). For now, the important point is that the final results do not provide partitions.

Advantages and tradeoffs. The central advantage of the partition-based approaches is that they explicitly yield a partition which can be directly employed in a segmentation pipeline or as a basis for a superpixel generator. Furthermore, a corresponding piecewise affine-linear function can be easily computed by linear regression on the segments. On the flipside, the existing partition-based approaches need an initialization procedure to obtain an initial partitioning. Further, their runtime grows with the number of labels and parallelization of the algorithms is more involved. The advantages of function-based methods are that they do not need an initial partitioning, and that they can be easily parallelized. The main drawback of function-based methods is that they only provide a (not necessarily) piecewise affine-linear approximation of the piecewise affine-linear function u whose discontinuity set is in general not the boundary set of a partition. As an undesirable side-effect, this means that the functional value in (7.2) cannot be evaluated directly which makes it difficult to quantify the quality of the algorithms.

The proposed algorithm unifies the advantages of partition-based methods and those of function-based methods: (i) it directly yields a partition $\mathcal{P}$; (ii) it does not need an initialization procedure; (iii) its average computation times are almost independent of the choice of the edge penalty; (iv) it is highly parallelizable. The experimental results show that the proposed method improves upon the state-of-the-art approach to the affine-linear Potts model in the sense that the algorithm has faster computation times and achieves lower functional values.

7.2. The affine-linear Potts model based on a jet formulation

To keep the presentation focused on the novel concepts, we carry out the derivations for single-channel images from Section 7.2.1 to Section 7.3.2. The extension to multi-channel images is discussed in Section 7.3.3.

7.2.1. Jet formulation of the affine-linear Potts model

A key point in our derivation is the formulation of problem (7.1) in terms of Taylor jets. We recall that the Taylor jet of a function u is a field of Taylor polynomials (recall Section 2.3). In the following, we will need the (first order) jet Ju of a function u. The Taylor jet Ju at the point $x \in \Omega$ is defined as the first order polynomial Ju_x given by

$$Ju_x(z) = u(x) + \partial_1 u(x)(z_1 - x_1) + \partial_2 u(x)(z_2 - x_2), \tag{7.3}$$

where $z \in \mathbb{R}^2$ is the argument of the polynomial (recall Section 2.3). Thus, $J_x u$ is the (first order) Taylor polynomial of u centered at x. Summing up, the jet $Ju : \Omega \to \Pi_1$ is a function on Ω with values in the space Π_1 of bivariate polynomials of order one; in particular, Ju_x is a polynomial, and $Ju_x(z)$ denotes the point evaluation of the polynomial Ju_x at the point $z \in \mathbb{R}^2$. It is a basic but important observation that u is piecewise affine-linear if and only if the jet Ju is piecewise constant.

Therefore we may reformulate (7.1) as

$$\begin{aligned} &\operatorname*{argmin}_{u,\mathcal{P}} \sum_{P\in\mathcal{P}} \left\{ \int_P |u(x) - f(x)|^2 \, dx + \frac{\gamma}{2} \operatorname{length}(\partial P) \right\}, \\ &\text{subject to } Ju|_P \text{ is constant for all } P \in \mathcal{P}. \end{aligned} \tag{7.4}$$

If we denote the length of the jump set of the jet Ju by $\|\nabla Ju\|_0$ (cf. Section 4.1), we can cast (7.4) into the form

$$u^* = \operatorname*{argmin}_{u} \int_\Omega |u(x) - f(x)|^2 \, dx + \gamma \, \|\nabla Ju\|_0. \tag{7.5}$$

(We note that the factor $1/2$ in the first term of (7.4), which compensated for the double counting of the boundaries, vanishes in this formulation.) Solutions of (7.5) are piecewise affine-linear because $\|\nabla Ju\|_0 < \infty$ if and only if the jet Ju of u is piecewise constant.

Instead of minimizing over u directly, we lift the problem to the jet space. This has the advantage that we may access the regularizing term more explicitly, while the data term still has a concrete representation in terms of jets. A basic but important observation is that there is a one-to-one correspondence between piecewise constant (first order) polynomial fields and the Taylor jets of piecewise affine-linear functions. First, let us denote by $\mathrm{PC}(\Omega, \Pi_1)$ the space of piecewise constant fields of first order polynomials. For each $J \in \mathrm{PC}(\Omega, \Pi_1)$ there is the corresponding piecewise affine-linear function $u(x) = J_x(x)$ such that $Ju = J$. (Recall that J_x is the evaluation of the field J at the point x, that is, a first order polynomial, and $J_x(x)$ denotes its evaluation at the point x.) This allows us –instead of (7.5)– to consider the following equivalent minimization problems over polynomial fields

$$J^* = \operatorname*{argmin}_{J\in\Pi(\Omega;\Pi_1)} \int_\Omega |J_x(x) - f(x)|^2 \, dx + \gamma \, \|\nabla J\|_0. \tag{7.6}$$

The key advantage of the jet-based formulation (7.6) is that the piecewise affine-linear problem in u has transformed into a piecewise constant problem in J. In particular, the connected components of the level sets of J^* are exactly the segments of the corresponding partition $\mathcal{P}^*$, and we have $u^*(x) = J^*_x(x)$ for all $x \in \Omega$.

7.2.2. Discretization

Let $\Omega' = \{1, \ldots, m\} \times \{1, \ldots, n\}$ be the discrete image domain. In the discrete setting, the arguments to optimize for are fields $J : \Omega' \to \Pi_1$ of affine-linear polynomials on the discrete domain Ω'. We may adopt a common way of discretizing the length penalty term in (7.6) similarly to the discretization of the length penalty in the Potts model in Section 4.2.1,

$$\|\nabla J\|_0 \approx \sum\nolimits_{s=1}^{S} \omega_s \|\nabla_{d_s} J\|_0, \tag{7.7}$$

see [48, 49, 181]. The terms on the right-hand side count the number of changes of the field J w.r.t. to a direction $d_s \in \mathbb{Z}^2$. That is,

$$\|\nabla_{d_s} J\|_0 = \left|\left\{x \in \Omega' : J_x \neq J_{x+d_s},\ x + d_s \in \Omega'\right\}\right|. \tag{7.8}$$

The directions d_s form a neighborhood system $\{d_1, \dots d_S\}$, with $S \geq 2$. We here focus on the more isotropic discretization $\{e_1, e_2, e_1 + e_2, e_1 - e_2\}$ with weights $\omega_{1,2} = \sqrt{2} - 1$ and $\omega_{3,4} = 1 - \frac{\sqrt{2}}{2}$ (cf. Section 4.2.1).

By employing the discretization (7.7) for the proposed jet formulation (7.6), we obtain the discrete problem

$$J^* = \operatorname*{argmin}_{J:\Omega' \to \Pi_1} \sum_{x \in \Omega'} |J_x(x) - f(x)|^2 + \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{d_s} J\|_0. \tag{7.9}$$

Minimizing (7.9) is a challenging problem: the second sum in (7.9) makes the problem non-smooth, non-convex and even NP-hard [190, 27]. We record that the discrete problem (7.9) has a minimizer.

Theorem 7.1. *The discrete affine-linear Potts model in its jet formulation* (7.9) *has a minimizer.*

The proof is analogous to the proof of Theorems 6.6 and 6.10 given below Theorem 6.10 in Chapter 6.

7.3. Algorithmic approach

We present a new algorithmic approach to the affine-linear Potts model (7.9). We employ a splitting approach based on the ADMM. To this end, we reformulate the original unconstrained problem (7.9) as an equivalent constrained problem with additional splitting variables. It turns out that the subproblems arising in the ADMM scheme for this constrained problem can be solved exactly and efficiently by adapting the solver developed in Chapter 3. Afterwards, we give details on the extension to multi-channel images. Finally, we discuss the relation of the proposed approach to existing approaches.

7.3.1. Splitting approach with the ADMM

In order to access the discrete length term (7.7) more directly, we reformulate (7.9) as a constrained problem. To this end, we split up the target J into S polynomial fields, $J^1, \dots, J^S$, subject to the constraints that they are all equal:

$$\begin{aligned} &\operatorname*{argmin}_{J^1,\dots,J^S} \sum_{s=1}^{S} \left\{ \frac{1}{S} \sum_{x \in \Omega'} |J^s_x(x) - f(x)|^2 + \gamma \omega_s \|\nabla_{d_s} J^s\|_0 \right\} \\ &\text{subject to } J^s = J^t \text{ for all } 1 \leq s < t \leq S. \end{aligned} \tag{7.10}$$

Note that due to the equality constraints, (7.10) is equivalent to the original problem (7.9). Since two bivariate first-order polynomials are equal if and only if their evaluation in three non-collinear points in $\mathbb{R}^2$ is equal, we can rewrite (7.10) as

$$\begin{aligned} &\operatorname*{argmin}_{J^1,\ldots,J^S} \sum_{s=1}^{S} \left\{ \frac{1}{S} \sum_{x \in \Omega'} |J_x^s(x) - f(x)|^2 + \gamma \omega_s \|\nabla_{d_s} J^s\|_0 \right\} \\ &\text{subject to } J_x^s(x) = J_x^t(x), \\ &\qquad J_x^s(x+e_1) - J_x^s(x) = J_x^t(x+e_1) - J_x^t(x), \\ &\qquad J_x^s(x+e_2) - J_x^s(x) = J_x^t(x+e_2) - J_x^t(x), \\ &\qquad \text{for all } x \in \Omega', \text{and all } s,t \text{ with } 1 \le s < t \le S. \end{aligned} \tag{7.11}$$

Recall that $e_1 = (1,0)^T, e_2 = (0,1)^T$ denote the coordinate directions. It is convenient to introduce the following notation

$$u^s(x) := J_x^s(x), \qquad \text{the function value of } J_x^s \text{ at } x, \tag{7.12}$$

$$a^s(x) := J_x^s(x+e_1) - J_x(x), \qquad \text{the horizontal slope of } J_x^s, \tag{7.13}$$

$$b^s(x) := J_x^s(x+e_2) - J_x(x), \qquad \text{the vertical slope of } J_x^s. \tag{7.14}$$

for the function value at the base point x and the slopes (i.e., the partial derivatives) of the polynomial J_x. We treat u^s, a^s, b^s as $(m \times n)$-matrices whose entries are given by $u^s(x), a^s(x), b^s(x)$, $x \in \Omega'$. Then, the constraint in (7.11) can be written as

$$u^s = u^t, \ a^s = a^t, \ b^s = b^t, \tag{7.15}$$

for all s,t with $1 \le s < t \le S$. In this context, we recall that the offsets in the origin $c(x)$ of the polynomials J_x are simply obtained by $c(x) = u(x) - x_1 a(x) - x_2 b(x)$. We note that the presented jet formulation results in a rather strong coupling which incorporates also the slopes/derivatives of the considered polynomials.

We now decompose the constrained problem into coupled subproblems using an ADMM approach. That is, we form the augmented Lagrangian form of (7.11) and iteratively perform a block-coordinate-wise minimization and gradient ascent steps on the Lagrange multipliers. We note that ADMM schemes often work well for non-convex problems; see, e.g., [197, 211, 54, 101]. The augmented Lagrangian form of (7.11) is given by

$$\begin{aligned} \mathcal{L}_{\mu,\nu}(\{J^s\}, \{\tau_u^{s,t}\}, \{\tau_a^{s,t}\}, \{\tau_b^{s,t}\}) = \textstyle\sum_{s=1}^{S} \Big\{ &\omega_s \gamma \|\nabla_{d_s} J^s\|_0 + \tfrac{1}{S}\|u^s - f\|^2 \\ + \textstyle\sum_{t=s+1}^{S} \Big(&\tfrac{\mu}{2}\Big\|u^s - (u^t - \tfrac{\tau_u^{s,t}}{\mu})\Big\|^2 - \tfrac{1}{2\mu}\|\tau_u^{s,t}\|^2 + \tfrac{\nu}{2}\Big\|a^s - (a^t - \tfrac{\tau_a^{s,t}}{\nu})\Big\|^2 - \tfrac{1}{2\nu}\|\tau_a^{s,t}\|^2 \\ &+ \tfrac{\nu}{2}\Big\|b^s - (b^t - \tfrac{\tau_b^{s,t}}{\nu})\Big\|^2 - \tfrac{1}{2\nu}\|\tau_b^{s,t}\|^2 \Big) \Big\}, \end{aligned} \tag{7.16}$$

where the $\tau_u^{s,t}, \tau_a^{s,t}, \tau_b^{s,t} \in \mathbb{R}^{m \times n}$ denote the Lagrange multipliers and $\|\cdot\|$ is the Frobenius norm, i.e. $\|u\|^2 = (\sum_{i,j} u_{ij}^2)$ for $u \in \mathbb{R}^{m \times n}$. The hard constraints in (7.11) are now part of the functional in the form of soft constraints, i.e., the squared Frobenius norms of the differences of the splitting variables. The parameters $\mu, \nu > 0$ determine how strong differences between the splitting variables are penalized. Note that we utilize two different coupling parameters: one for the slope variables a, b and one for the base point variable u. This is because slopes and base points typically live on different scales.

In each iteration, we minimize $\mathcal{L}_{\mu,\nu}$ w.r.t. each J^s and perform a gradient ascent on the Lagrange multipliers. For a fixed s, the according minimization problem reads

$$\begin{aligned} \operatorname*{argmin}_J \mathcal{L}_{\mu,\nu} = \operatorname*{argmin}_J \Big\{ & \omega_s \gamma \|\nabla_{d_s} J\|_0 + \tfrac{1}{S} \|u - f\|^2 \\ + \sum_{t=s+1}^{S} & \Big(\tfrac{\mu}{2} \|u - (u^t - \tfrac{\tau_u^{s,t}}{\mu})\|^2 + \tfrac{\nu}{2} \|a - (a^t - \tfrac{\tau_a^{s,t}}{\nu})\|^2 + \tfrac{\nu}{2} \|b - (b^t - \tfrac{\tau_b^{s,t}}{\nu})\|^2 \Big) \\ + \sum_{r=1}^{s-1} & \Big(\tfrac{\mu}{2} \|u - (u^r + \tfrac{\tau_u^{r,s}}{\mu})\|^2 + \tfrac{\nu}{2} \|a - (a^r + \tfrac{\tau_a^{r,s}}{\nu})\|^2 + \tfrac{\nu}{2} \|b - (b^r + \tfrac{\tau_b^{r,s}}{\nu})\|^2 \Big) \Big\}. \end{aligned} \tag{7.17}$$

Note that all other terms in (7.16) do not depend on J^s, so we dropped them. In the following, we bring (7.17) into a convenient form (similarly to the derivation in Section 6.3.1). To this end, we will use repeatedly the fact that

$$\sum\nolimits_{i=1}^{N} x_i (p - t_i)^2 = \Big(\sum\nolimits_{i=1}^{N} x_i \Big) \Big(p - \tfrac{\sum_{i=1}^{N} t_i x_i}{\sum_{i=1}^{N} x_i} \Big)^2 + C \tag{7.18}$$

holds for $p, t_1, \ldots, t_N \in \mathbb{R}$ and $x_1, \ldots, x_N > 0$ and a constant C that does not depend on p. Initially, this allows us to rewrite the summands in (7.17) and we get

$$\begin{aligned} \operatorname*{argmin}_J \Big\{ & \omega_s \gamma \|\nabla_{d_s} J\|_0 + \tfrac{1}{S} \|u - f\|^2 \\ & + \tfrac{(S-s)\mu}{2} \Big\| u - \frac{\sum_{t=s+1}^{S} (u^t - \frac{\tau_u^{s,t}}{\mu})}{(S-s)} \Big\|^2 + \tfrac{(s-1)\mu}{2} \Big\| u - \frac{\sum_{r=1}^{s-1} (u^r + \frac{\tau_u^{r,s}}{\mu})}{(s-1)} \Big\|^2 \\ & + \tfrac{(S-s)\nu}{2} \Big\| a - \frac{\sum_{t=s+1}^{S} (a^t - \frac{\tau_a^{s,t}}{\nu})}{(S-s)} \Big\|^2 + \tfrac{(s-1)\nu}{2} \Big\| a - \frac{\sum_{r=1}^{s-1} (a^r + \frac{\tau_a^{r,s}}{\nu})}{(s-1)} \Big\|^2 \\ & + \tfrac{(S-s)\nu}{2} \Big\| b - \frac{\sum_{t=s+1}^{S} (b^t - \frac{\tau_b^{s,t}}{\nu})}{(S-s)} \Big\|^2 + \tfrac{(s-1)\nu}{2} \Big\| b - \frac{\sum_{r=1}^{s-1} (b^r + \frac{\tau_b^{r,s}}{\nu})}{(s-1)} \Big\|^2 \Big\}. \end{aligned} \tag{7.19}$$

Again, we dropped terms that do not depend on J. For readability we introduce the following abbreviations for the sums in (7.19):

$$\begin{aligned} \Lambda &= \sum\nolimits_{t=s+1}^{S} (u^t - \tau_u^{s,t}/\mu), & \Psi &= \sum\nolimits_{t=s+1}^{S} (a^t - \tau_a^{s,t}/\nu), \\ \Delta &= \sum\nolimits_{t=s+1}^{S} (b^t - \tau_b^{s,t}/\nu), & \Gamma &= \sum\nolimits_{r=1}^{s-1} (u^r + \tau_u^{r,s}/\mu), \\ \Phi &= \sum\nolimits_{r=1}^{s-1} (a^r + \tau_a^{r,s}/\nu), & \Theta &= \sum\nolimits_{r=1}^{s-1} (b^r + \tau_b^{r,s}/\nu). \end{aligned}$$

We insert these abbreviations and apply (7.18) to all but the first line of (7.19) and obtain

$$\operatorname*{argmin}_J \omega_s\gamma\|\nabla_{d_s}J\|_0 + \tfrac{1}{S}\|u-f\|^2 + \tfrac{(S-1)\mu}{2}\Big\|u - \tfrac{\Lambda+\Gamma}{S-1}\Big\|^2 + \tfrac{(S-1)\nu}{2}\Big\|a - \tfrac{\Psi+\Phi}{S-1}\Big\|^2 + \tfrac{(S-1)\nu}{2}\Big\|b - \tfrac{\Delta+\Theta}{S-1}\Big\|^2. \tag{7.20}$$

A final application of (7.18) to both remaining terms that depend on u in (7.20) leads to

$$\operatorname*{argmin}_J \omega_s\gamma\|\nabla_{d_s}J\|_0 + \tfrac{2+\mu S(S-1)}{2S}\Big\|u - \tfrac{2f+\mu S(\Lambda+\Gamma)}{2+\mu S(S-1)}\Big\|^2 + \tfrac{(S-1)\nu}{2}\Big\|a - \tfrac{\Psi+\Phi}{S-1}\Big\|^2 + \tfrac{(S-1)\nu}{2}\Big\|b - \tfrac{\Delta+\Theta}{S-1}\Big\|^2 \tag{7.21}$$

and after multiplying (7.21) by $\frac{2}{(S-1)\nu}$, we obtain

$$\operatorname*{argmin}_J \tfrac{2\omega_s\gamma}{(S-1)\nu}\|\nabla_{d_s}J\|_0 + \tfrac{2+\mu S(S-1)}{\nu S(S-1)}\Big\|u - \tfrac{2f+\mu S(\Lambda+\Gamma)}{2+\mu S(S-1)}\Big\|^2 + \Big\|a - \tfrac{\Psi+\Phi}{S-1}\Big\|^2 + \Big\|b - \tfrac{\Delta+\Theta}{S-1}\Big\|^2. \tag{7.22}$$

Together with the gradient ascents on the Lagrange multipliers we derive the following iterative procedure

$$\begin{aligned}
(J^s)^{j+1} &= \operatorname*{argmin}_J \tfrac{2\omega_s\gamma}{(S-1)\nu_j}\|\nabla_{d_s}J\|_0 + \tfrac{2+\mu_j S(S-1)}{\nu_j S(S-1)}\|u-(\overline{u}^s)^j\|^2 + \|a-(\overline{a}^s)^j\|^2 + \|b-(\overline{b}^s)^j\|^2 \\
&\qquad \text{for all } s = 1,\dots,S, \\
(\tau_u^{s,t})^{j+1} &= (\tau_u^{s,t})^j + \mu_j\Big((u^s)^{j+1} - (u^t)^{j+1}\Big), \\
(\tau_a^{s,t})^{j+1} &= (\tau_a^{s,t})^j + \nu_j\Big((a^s)^{j+1} - (a^t)^{j+1}\Big), \\
(\tau_b^{s,t})^{j+1} &= (\tau_b^{s,t})^j + \nu_j\Big((b^s)^{j+1} - (b^t)^{j+1}\Big) \qquad \text{for all } 1 \le s < t \le S,
\end{aligned} \tag{7.23}$$

where the superscript j denotes the iteration number and where we used the following abbreviations

$$(\overline{u}^s)^j = \frac{2f + \mu_j S\left(\sum\limits_{t=s+1}^{S}\Big((u^t)^j - \frac{(\tau_u^{s,t})^j}{\mu_j}\Big) + \sum\limits_{r=1}^{s-1}\Big((u^r)^{j+1} + \frac{(\tau_u^{r,s})^{j+1}}{\mu_j}\Big)\right)}{2+\mu_j S(S-1)},$$

$$\begin{aligned}
(\overline{a}^s)^j &= \tfrac{1}{S-1}\sum_{t=s+1}^{S}\Big((a^t)^j - \tfrac{(\tau_a^{s,t})^j}{\nu_j}\Big) + \tfrac{1}{S-1}\sum_{r=1}^{s-1}\Big((a^r)^{j+1} + \tfrac{(\tau_a^{r,s})^{j+1}}{\nu_j}\Big), \\
(\overline{b}^s)^j &= \tfrac{1}{S-1}\sum_{t=s+1}^{S}\Big((b^t)^j - \tfrac{(\tau_b^{s,t})^j}{\nu_j}\Big) + \tfrac{1}{S-1}\sum_{r=1}^{s-1}\Big((b^r)^{j+1} + \tfrac{(\tau_b^{r,s})^{j+1}}{\nu_j}\Big).
\end{aligned}$$

As it is common when dealing with non-convex problems, we employ increasing coupling sequences $(\mu_j)_{j\in\mathbb{N}}, (\nu_j)_{j\in\mathbb{N}}$ as coupling parameters. Note that the computation of $\overline{u}^s, \overline{a}^s, \overline{b}^s$ and the Lagrange multiplier updates only involve pointwise basic arithmetic operations. We stop the iterations when the splitting variables become (approximately) equal. Please note that we return the mean values to avoid accentuating a particular set of splitting variables. We provide a pseudocode for the the algorithmic scheme in Algorithm 7.1.

Algorithm 7.1: ADMM strategy for the affine-linear Potts problem

```
Input: Image $f$; edge penalty $\gamma > 0$; directions $\{d_s\}_{s=1}^S$
Output: Piecewise affine-linear approximation $u$;
corresponding piecewise constant field of first order polynomials $(a, b, c)$
/* Initialization */
$J^s = (u^s, a^s, b^s) = (f, 0, 0)$ for all $s = 1, \dots, S$
$\tau_u^{s,t}, \tau_a^{s,t}, \tau_b^{s,t} = 0$ for all $1 \le s < t \le S$; $j \leftarrow 1$
repeat
    for $s = 1, \dots, S$ do
        /* Compute data for jet subproblems */
        $\overline{u}^s \leftarrow \frac{2f + \mu_j S \left( \sum_{t=s+1}^{S} \left( u^t - \frac{\lambda^{s,t}}{\mu_j} \right) + \sum_{r=1}^{s-1} \left( u^r + \frac{\lambda^{r,s}}{\mu_j} \right) \right)}{2 + \mu_j S(S-1)}$
        $\overline{a}^s \leftarrow \frac{1}{S-1} \sum_{t=s+1}^{S} \left( a^t - \frac{\tau^{s,t}}{\nu_j} \right) + \frac{1}{S-1} \sum_{r=1}^{s-1} \left( a^r + \frac{\tau^{r,s}}{\nu_j} \right)$
        $\overline{b}^s \leftarrow \frac{1}{S-1} \sum_{t=s+1}^{S} \left( b^t - \frac{\rho^{s,t}}{\nu_j} \right) + \frac{1}{S-1} \sum_{r=1}^{s-1} \left( b^r + \frac{\rho^{r,s}}{\nu_j} \right)$
        /* Solve the linewise segmented jet estimation subproblems w.r.t. direction $d_s$ */
        $J^s \leftarrow \operatorname*{argmin}_J \frac{2\omega_s \gamma}{(S-1)\nu_j} \|\nabla_{d_s} J\|_0 + \frac{2 + \mu_j S(S-1)}{\nu_j S(S-1)} \|u - \overline{u}^s\|^2 + \|a - \overline{a}^s\|^2 + \|b - \overline{b}^s\|^2$
    end
    /* Update the Lagrange multipliers */
    for $s = 1, \dots, S$ do
        for $t = s + 1, \dots, S$ do
            $\tau_u^{s,t} \leftarrow \tau_u^{s,t} + \mu_j (u^s - u^t)$
            $\tau_a^{s,t} \leftarrow \tau_a^{s,t} + \nu_j (a^s - a^t)$
            $\tau_b^{s,t} \leftarrow \tau_b^{s,t} + \nu_j (b^s - b^t)$
        end
    end
    /* Update coupling parameters */
    $\mu_j \leftarrow \mu_{j+1}$, $\nu_j \leftarrow \nu_{j+1}$
    $j \leftarrow j + 1$
until $\max_{i,j} \left\{ \frac{|q^s(i,j) - q^{s+1}(i,j)|}{|q^s(i,j)| + |q^{s+1}(i,j)|} \right\} \le \eta_{\mathrm{stop}}$ for all $s = 1, \dots, S - 1$, $q \in \{u, a, b\}$;
/* Take averages of splitting variables */
$u \leftarrow \frac{1}{S} \sum_{s=1}^{S} u^s$;  $a \leftarrow \frac{1}{S} \sum_{s=1}^{S} a^s$;  $b \leftarrow \frac{1}{S} \sum_{s=1}^{S} b^s$;
/* Compute offsets in origin $c$ from $u, a, b$ */
for $x \in \Omega'$ do
    $c(x) \leftarrow u(x) - x_1 a(x) - x_2 b(x)$;
end
return $u, a, b, c$;
```

7.3.2. Efficient solution of the subproblems

The first line of (7.23) corresponds to solving S non-convex minimization problems in $J^1, ..., J^S$. The crucial observation is that by (7.8) these subproblems decompose into mutually independent one-dimensional partitioning problems along the paths in Ω' induced by the direction d_s. In this context, we observe that they can further be solved in parallel. (An illustration of these paths was given in Figure 4.2 in Chapter 4.) In the following, we describe an exact and efficient solver for the one-dimensional partitioning problems. It follows the ideas for the solver developed in Chapter 3 which we adapt to deal with the problems here.

We consider exemplarily the first subproblem in (7.23) and note that solving the others can be done analogously. Let $\overline{u}, \overline{a}, \overline{b} \in \mathbb{R}^n$ denote the data for the functional values and the data for the slopes, respectively. Then by using the notation (7.12)-(7.14), the univariate problems we have to solve have the generic form

$$J^* = \operatorname*{argmin}_{u,a,b \in \mathbb{R}^n} \eta^2 \|u - \overline{u}\|^2 + \|a - \overline{a}\|^2 + \|b - \overline{b}\|^2 + \gamma' \|\nabla J\|_0. \tag{7.24}$$

We recall that for the one-dimensional domain $\{1, \dots, n\}$, the symbol $\|\nabla J\|_0$ counts the number of changes of J, i.e. $\|\nabla J\|_0 = |\{i \in 1, \dots, n-1\} : J_i \neq J_{i+1}\}|$. Consequently, (7.24) is a one-dimensional segmented least squares problem on $\{1, \dots, n\}$.

As a first step, we reformulate (7.24) in terms of partitions only. Recall that a partition $\mathcal{I}$ of $\{1, \dots, n\}$ is a set of pairwise disjoint discrete intervals of the form $I = \{l, l+1, \dots, r\}$ such that $\{1, \dots, n\} = \bigcup_{I \in \mathcal{I}} I$ (cf. Section 3.2). We use the notation $I = l : r$ as in Chapter 3. The formulation in terms of the partition only is now given by

$$\mathcal{I}^* = \operatorname*{argmin}_{\mathcal{I} \text{ partition of } 1:n} \sum_{I \in \mathcal{I}} \left(\mathcal{E}^I + \gamma' \right), \tag{7.25}$$

where $\mathcal{E}^I$ denotes the (optimal) jet approximation error on the segment I in the sense of

$$\mathcal{E}^I = \min_{\alpha, \beta, \delta \in \mathbb{R}} \sum_{i \in I} \eta^2 |\delta + i\alpha - \overline{u}_i|^2 + |\alpha - \overline{a}_i|^2 + |\beta - \overline{b}_i|^2. \tag{7.26}$$

As a second step, the corresponding optimal jet J^* is recovered from the optimal partition $\mathcal{I}^*$ by solving the least squares objective (7.26) w.r.t. u, a, b for each segment $I \in \mathcal{I}$ separately.

The partitioning problem (7.25) can be solved with the algorithm developed in Chapter 3. To see this, we denote the objective in (7.25) by B, i.e.,

$$B(\mathcal{I}) = \sum_{I \in \mathcal{I}} \left(\mathcal{E}^I + \gamma' \right). \tag{7.27}$$

On the reduced domain $1 : r$, $r < n$ we denote its minimal value by

$$B_r^* = \min_{\mathcal{I} \text{ partition on } 1:r} B(\mathcal{I}). \tag{7.28}$$

We observe that the minimal value B_r^* on the domain $1 : r$ satisfies the Bellman equation

$$B_r^* = \min_{l=1,\dots,r} \left\{ \mathcal{E}^{l:r} + \gamma' + B_{l-1}^* \right\}, \tag{7.29}$$

where we set $B_0^* = -\gamma'$. Thus, the dynamic programming principle [12] allows us to compute $B_1^*, B_2^*, \dots$ until we reach B_n^*. We keep track of an optimal partition $\mathcal{I}^*$ by storing at step r the minimizing argument l' of (7.29) as the value L_r, so that L encodes the boundaries of an optimal partition.

Similar to Section 6.3.2, the problems here are given in terms of discrete jets and not in terms of discrete function values only. However, the error update scheme of Chapter 3 can nevertheless be employed to solve the partitioning problem (7.25). To this end, we reformulate (7.26) to

$$\mathcal{E}^{l:r} = \min_{\alpha,\beta,\delta} \|A^q \big(\alpha, \beta, \delta\big)^T - g^{l:r}\|^2, \tag{7.30}$$

where $q = r - l + 1$ denotes the length of the interval $I = l : r$ and the system matrix $A^q \in \mathbb{R}^{3q\times 3}$ and the data vector $g^{l:r} \in \mathbb{R}^{3q}$ are given by

$$A^q = \begin{pmatrix} \eta & 0 & \eta \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ q\eta & 0 & \eta \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \text{and} \quad g^{l:r} = \begin{pmatrix} \eta\bar{u}_l \\ \bar{a}_l \\ \bar{b}_l \\ \vdots \\ \eta\bar{u}_r \\ \bar{a}_r \\ \bar{b}_r \end{pmatrix}. \tag{7.31}$$

Since the system matrices A^q are tall, we can efficiently employ the error update strategy developed in Chapter 3 and consequently Algorithm 3.1 to solve (7.24) (see also Remark 3.11).

7.3.3. Multi-channel images

We first note that solving the affine-linear Potts problem (7.1) for multi-channel images $f : \Omega \to \mathbb{R}^K$ is not equivalent to solving the single-channel variant channel by channel. The latter approach typically leads to undesired artifacts as the partition boundaries are not enforced to be aligned over the channels; see Figure 7.3.

Jet formulation for multi-channel images. To extend the proposed approach to the multi-channel affine-linear Potts problem, we consider the first order Taylor jet of $u : \Omega \to \mathbb{R}^K$ which is given by the Taylor jets of its component functions, i.e.,

$$J_x u = (J_x u_1, \dots, J_x u_K)^T. \tag{7.32}$$

Then, the multi-channel version of the jet formulation (7.6) is given by

$$J^* = \underset{J \in \Pi(\Omega;(\Pi_1)^K)}{\operatorname{argmin}} \quad \gamma \, \|\nabla J\|_0 + \sum_{k=1}^{K} \int_\Omega |J_{x,k}(x) - f_k(x)|^2 \, dx. \tag{7.33}$$

where the minimum is taken over piecewise constant multi-channel jets. (We here use the notation $J_{x,k}$ to denote the polynomial at x corresponding to the k-th channel.) As the domain

(a) Input image (b) Channel-wise approach (68.0 sec) (c) Multi-channel approach (26.9 sec)

Figure 7.3.: Comparison of channel-wise application of the affine-linear Potts model and the multi-channel affine-linear Potts model for the input image (a). The channel-wise approach (b) suffers from color artifacts (see, e.g., the face of the right boy or the ears of the left boy). (c) The multi-channel approach yields a more reasonable image approximation and an improved partition. Furthermore, it needs less computation time.

of the image does not change in the multi-channel case, the discretization of the jump penalty in (7.7) remains unchanged. We note that the partition boundaries are enforced to be aligned across the channels. This can be seen by looking at the counting of the directional differences in (7.8): in the multi-channel case, we have that $J_x \neq J_{x+d^s}$ if and only if J_x, J_{x+d_s} are different in *at least one component*, i.e., if and only if $J_{x,k} \neq J_{x+d^s,k}$ for at least one $k \in \{1, \ldots, K\}$. That means, opening a "jump" between x and $x + d^s$ in all channels has no extra costs compared to opening a jump in a single channel.

Discrete multi-channel problem. Starting from (7.33) –in analogy to the derivation starting from (7.6) in the single-channel case– we obtain the discretized problem

$$J^* = \underset{J:\Omega' \to (\Pi_1)^K}{\operatorname{argmin}} \sum_{x\in\Omega'} \sum_{k=1}^{K} |J_{x,k}(x) - f_k(x)|^2 + \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{d_s} J\|_0. \tag{7.34}$$

Then the analogue of the splitting (7.10) for the single-channel case becomes

$$\begin{aligned} &\underset{J^1,\ldots,J^S}{\operatorname{argmin}} \frac{1}{S} \sum_{x\in\Omega'} \sum_{k=1}^{K} |J^s_{x,k}(x) - f_k(x)|^2 + \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{d_s} J^s\|_0 \\ &\text{subject to } J^s = J^t \quad \text{for all } 1 \le s < t \le S. \end{aligned} \tag{7.35}$$

We rewrite (7.35) in the way of (7.11) which gives

$$\begin{aligned} &\underset{J^1,\ldots,J^S}{\operatorname{argmin}} \sum_{s=1}^{S} \left\{ \frac{1}{S} \sum_{x\in\Omega'} \sum_{k=1}^{K} |J^s_{x,k}(x) - f_k(x)|^2 + \gamma\omega_s \|\nabla_{d_s} J^s\|_0 \right\} \\ &\text{subject to } u^s(x) = u^t(x),\ a^s_x(x) = a^t(x),\ b^s_x(x) = b^t(x), \\ &\text{for all } x \in \Omega', \text{and all } s, t \text{ with } 1 \le s < t \le S, \end{aligned} \tag{7.36}$$

where we used notation (7.12)-(7.14) and note that now $u^s, a^s, b^s \in \mathbb{R}^{m\times n\times K}$. The Lagrangian of (7.36) which corresponds to (7.16) in the single-channel case is now understood w.r.t. the La-

(a) Input image

(b) Fast progression, $\varphi = 1.7$, 37 iterations, $E = 2899.9$

(c) Medium progression, $\varphi = 1.3$, 65 iterations $E = 2821.7$

(d) Slow progression, $\varphi = 1.05$, 285 iterations $E = 2796.0$

Figure 7.4.: Effect of the progression parameter. Small progression parameters φ yield improved partitions and lower functional values E, but more iterations are needed. As a good compromise of runtime and quality, we find the medium choice $\varphi = 1.3$.

grange multipliers $\tau_u^{s,t}, \tau_a^{s,t}, \tau_b^{s,t} \in \mathbb{R}^{m\times n\times K}$ and the squared Frobenius norm $\|u\|^2 = \sum_{i,j,k} u_{ijk}^2$. The further derivation of the algorithm is analogous to the single-channel case.

Univariate subproblems for multi-channel data. Concerning the univariate subproblems, the jet approximation errors $\mathcal{E}^I$ in the dynamic programming scheme are given by the channel-wise sum

$$\mathcal{E}^I = \sum_{k=1}^{K} \mathcal{E}_k^I. \tag{7.37}$$

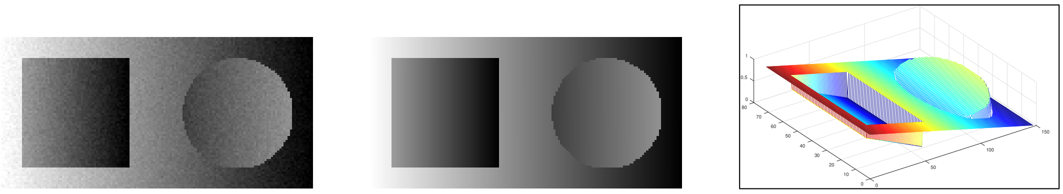
Consequently, the error update strategy based on Givens rotations from Section 7.3.2 can be used by simply updating the approximation errors channel-wise. For this we can use the same system matrix A_q from (7.31) for all channels and the recurrence coefficients in Algorithm 3.1 have to be computed only once for all channels. It is straightforward to see that Corollary 6.16 applies here as well, i.e., the multi-channel version of the proposed scheme produces an exact solution of the subproblems (7.24) in $O(n^2K)$ time.

7.3.4. Relation to other approaches

In contrast to the proposed approach, the splitting approach in [77] couples the function values u only. In view of the jet formulation of the affine-linear Potts model, a “complete” coupling requires also to couple the slopes. Hence, the method in [77] is the straightforward extension of the method in [181] and can be seen as a relaxation of the proposed method. The approach in [77] leads to standard univariate affine-linear Potts subproblems instead of the jet estimation subproblems in the proposed approach. In contrast, the proposed method requires a tailored subproblem solver, see Section 7.3.2. Compared with [77], a main advantage of the proposed method is that it yields a partition directly.

Another related problem is given (in a discretized version) by

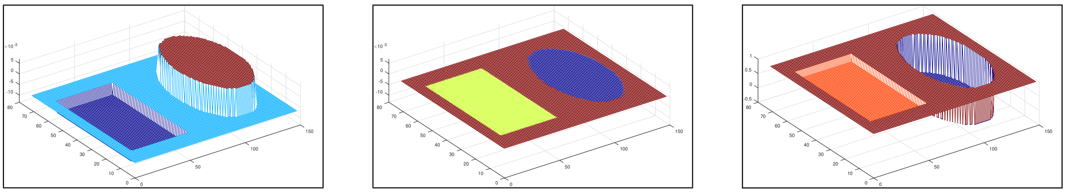
$$\operatorname*{argmin}_{u:\Omega'\to\mathbb{R}} \gamma \sum_{s=1}^{S} \omega_s \|\nabla_{d_s}^2 u\|_0 + \sum_{x\in\Omega'} |u(x) - f(x)|^2. \tag{7.38}$$



(a) Input image

(b) Piecewise affine-linear approximation *u*

(c) 3D-plot of (b)



(d) Horizontal slopes *a*

(e) Vertical slopes *b*

(f) Offsets in origin *c*

Figure 7.5.: Illustration of the jet result. While the reconstruction u is piecewise affine-linear (b),(c), the corresponding field of first order polynomials (i.e., the corresponding jet) is piecewise constant (d)-(f).

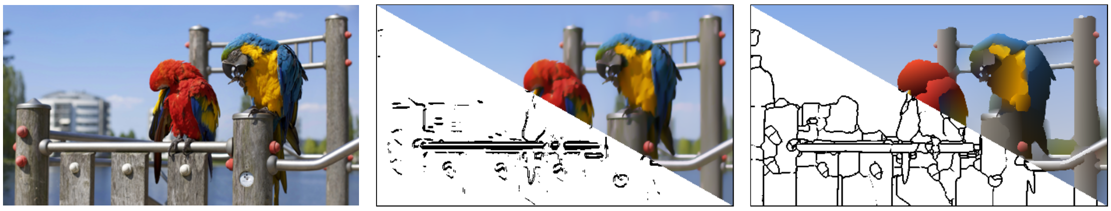
We stress that this is not equivalent to the affine-linear Potts model. The relevant difference is that $\|\nabla^2_{d^s} u\|_0$ counts the number of kinks of u instead of the number of affine coefficient changes. The practical consequence is that the functional favors solutions which tend to have kinks rather than jumps (cf. Section 3.2.4 and Example 3.9 for the univariate case).

7.4. Experimental results

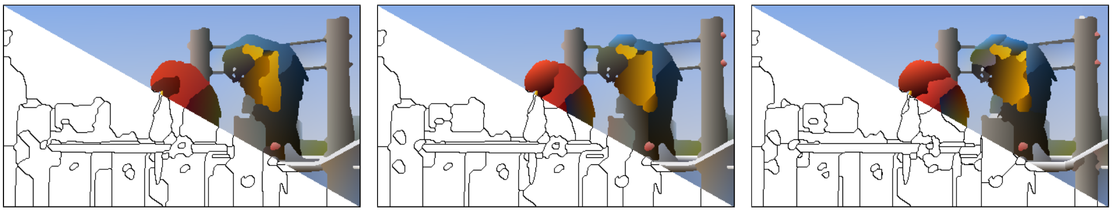
In this section, we first give the necessary implementation details. Then we apply the proposed method to a synthetic image to give an illustration of the computed field of first order polynomials. After recalling the methods for comparison, we compare them with the proposed method qualitatively. Finally, we give a quantitative comparison of the proposed method with the state-of-the-art method based on iterative repartitioning with graph cuts.

Implementation details. We initialize the splitting variables by $u_0 = f$, $a_0 = b_0 = 0$. As initial coupling parameters we set $\mu_0 = 10^{-3}$, $\nu_0 = \min\{450\gamma\mu_0, 1\}$. After each iteration (7.23), we increase μ, ν by the factor φ. As a good compromise of runtime and quality, we found the choice $\varphi = 1.3$; see Figure 7.4. We stopped the iteration when the largest pointwise relative difference between consecutive splitting variables became smaller than 10^{-2}, i.e., when $\max_{x\in\Omega'} \frac{|q^s(x)-q^{s+1}(x)|}{|q^s(x)|+|q^{s+1}(x)|} \leq 10^{-2}$ for $s = 1, \dots, S-1$ which corresponds to $\eta_{\mathrm{stop}} = 10^{-2}$ in Algorithm 7.1. All experiments were conducted on a workstation (Intel XeonE5-2620v4, 2.10GHz, 16 cores, 256 GB RAM).

Illustration of the produced jet. In addition to the piecewise affine-linear approximation u of the input image f, the proposed method yields the corresponding piecewise constant



(a) Input (b) GNC (c) FPAME



(d) GC (64 labels) (e) GC (256 labels) (f) Proposed

Figure 7.6.: Qualitative comparison of approaches to the affine-linear Potts model. *Upper half*: results u^*; *lower half*: the edge sets derived from u^* for the partition-free methods (GNC and FPAME), and the boundaries of the partition $\mathcal{P}^*$ for the partition-based methods (GC and the proposed method), respectively. The edge sets of the partition-free methods do not always correspond to closed curves, i.e., the edge sets do not form the boundaries of a partition.

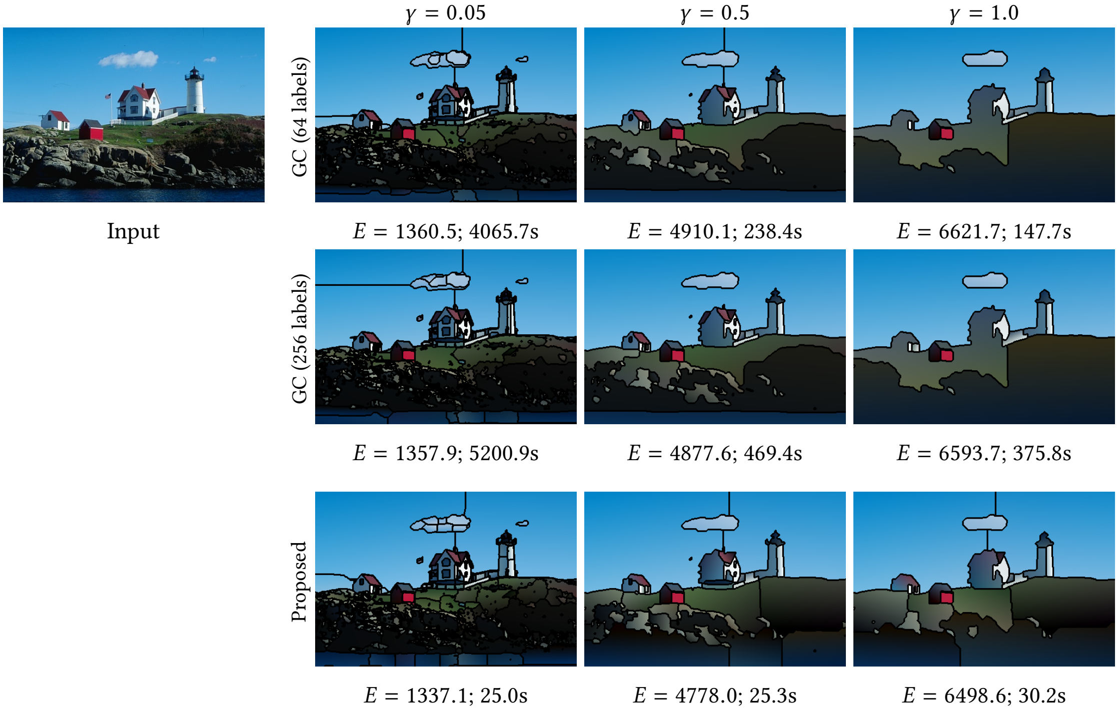
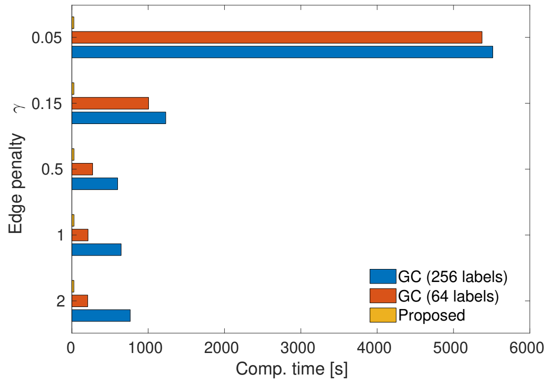


Figure 7.7.: Comparison to iterative repartitioning with graph cuts. The proposed approach and the graph cut approach both provide reasonable partitions. Yet, there are some slight differences. For, e.g., $\gamma = 0.5$, graph cuts still opens a segment for the water, but not for the fence. In the proposed approach, this is vice versa. As a general tendency, we observe that the proposed approach produces more segments throughout all γ-levels. We report the model energies E of the results and runtimes below the images.



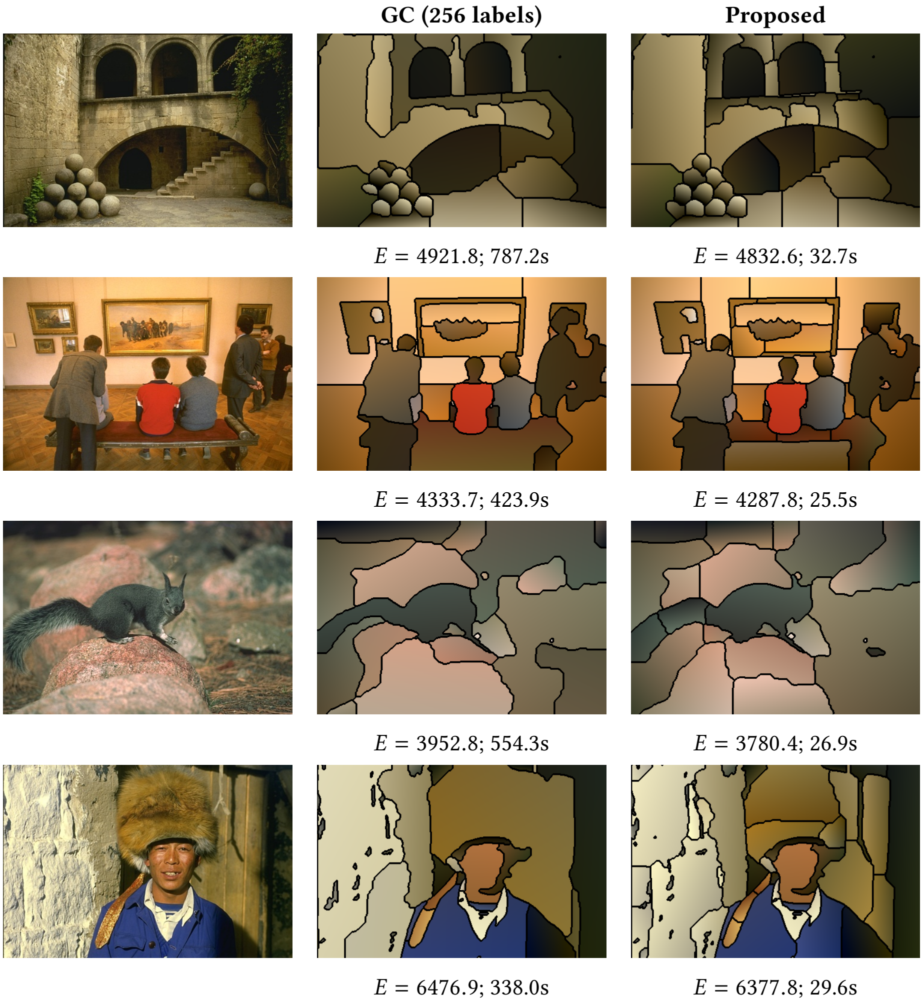
γ	GC (64 labels)	GC (256 labels)	Proposed
0.05	1802.2	1802.1	**1772.3**
0.15	3180.3	3178.2	**3146.0**
0.50	5484.8	5480.7	**5418.2**
1.00	7324.7	7317.5	**7202.7**
2.00	9551.5	9526.4	**9356.7**

Figure 7.8.: Quantitative comparison with iterative graph cuts on the Berkley test set. *Left:* average runtimes for the Berkley test set. The proposed method has significantly lower mean computation times than the iterative graph cuts method (the proposed approach needs roughly 28 sec. for any γ). *Right:* mean energies over the Berkeley test set and the parameters γ in the left column. The proposed method achieves the lowest mean energies.

field of first order polynomials in terms of their coefficients a, b, c. In Figure 7.5, we apply the proposed algorithm to a synthetic image and illustrate the corresponding jet by 3D-plots. Indeed, while u is piecewise affine-linear, the corresponding field of first order polynomials is piecewise constant and encodes the partition associated with u.

Methods for comparison. The methods for comparison are the graduated non-convexity approach (GNC), the method of [77] (FPAME), and the iterative repartitioning scheme based on graph cuts (GC). We implemented GNC by following the lines of [21]. We recall that GNC solves a relaxed problem that has a smoothing parameter (recall Section 6.4). We tested several such parameters and obtained the best visual results for the choice 5^4. Regarding FPAME, we adapted the method [77] to the ℓ_2-data term here. As GNC and FPAME do not directly yield a partition we extracted their edge sets by taking second order finite differences and thresholding as in [21, 77]. The iterative repartitioning scheme uses the α-expansion algorithm of the toolbox GCO v3.0 [27, 26, 127]. In order to obtain a reasonable initial partition $\mathcal{P}^0$ for the iterative repartitioning scheme, we extracted L (equidistant) color values from the image and performed a piecewise constant partitioning for these initial labels. Then we proceeded with the iterative repartitioning approach until the relative improvement of the energy became smaller than 10^{-3} or after a maximum number of 10 iterations was reached.

Qualitative comparison. A qualitative comparison is given in Figure 7.6. We observe that GNC does not give a genuinely piecewise affine-linear estimate. The edge sets of GNC and FPAME have open ends. Hence, they are not the boundaries of a partition. As a consequence, GNC and FPAME do not allow for an evaluation of the functional value which hinders a quantitative evaluation in terms of the energy. (Note that [21, 77] also do not provide a post-processing which we could use to obtain a partition.) Thus, in the following quantitative evaluation, we focus on a detailed comparison to GC.



GC (256 labels) **Proposed**

$E = 4921.8$; 787.2s $E = 4832.6$; 32.7s

$E = 4333.7$; 423.9s $E = 4287.8$; 25.5s

$E = 3952.8$; 554.3s $E = 3780.4$; 26.9s

$E = 6476.9$; 338.0s $E = 6377.8$; 29.6s

Figure 7.9.: Example results for the Berkeley data set. *From left to right:* four images from the BSD500, results of the graph cut approach (256 initial labels), and results of the proposed approach. The model parameter is $\gamma = 0.5$. Energies and runtimes are reported below the images.

Quantitative comparison to iterative repartitioning with graph cuts. We compare our method to GC with respect to energy value and runtime. First, we observe that the results of GC depend on the number of chosen initial labels: in most cases, choosing more labels leads to a lower energy value, while it also increases the runtime; see Figure 7.7. Hence, we compare our approach to GC with a smaller and a larger number of initial labels (64 and 256). To get meaningful results we used the test images of the Berkeley segmentation data set [4] which consist of 200 natural 481×321 color images. Figure 7.8 reveals that the proposed method achieves lower mean energies and faster computation times than GC which illustrates the effectiveness of the algorithm. In particular for small model parameter γ, the proposed scheme is significantly faster. We provide further examples in Figure 7.9.

We observed the convergence of the proposed scheme in all our experiments. For this kind of problems theoretical convergence results are rather weak in general. In particular, also for the state-of-the-art method based on iterative application of graph cuts (we compare with here) the convergence is observed only empirically as well.

7.5. Summary of the chapter

In this chapter, we developed an efficient algorithm for computing approximate solutions of the affine-linear Potts model which is a method for image partitioning based on affine-linear segments. Hence, in contrast to the classical (constant) Potts model, the affine-linear Potts model allows for estimating linear trends on the segments. As a result, the affine-linear Potts model frequently yields improved partitioning results than the Potts model as natural images often possess linear trends.

A novel formulation of the underlying minimization problem in terms of Taylor jets allowed us to invoke a splitting into few tightly coupled jet-subproblems for which we developed an efficient and exact solver. We saw that, in contrast to GNC and FPAME, the proposed method provides a genuine partition of the image. Our experiments showed that the proposed approach yields results with lower average model energies than those of the benchmark approach based on iterative repartitioning using graph cuts (with α-expansion moves). Hence, the model energy is efficiently minimized. At the same time, it needs less computation time than the benchmark method.

8. Discussion

We presented and studied novel algorithmic approaches to Mumford-Shah and Potts models which are variational methods for edge-preserving smoothing and partitioning of signals and images. The application of Mumford-Shah and Potts models corresponds to dealing with hard non-convex minimization problems. Already the simplest case of piecewise constant solutions, the classical Potts model, corresponds to solving an NP-hard problem when applied to images and indirect data, respectively. Finding new algorithmic approaches to Mumford-Shah and Potts models is an active field of research.

In Chapter 3, we studied the univariate situation and noticed that the classical first order Mumford-Shah and Potts models have limitations when applied to signals with (steep) slopes. In this situation, the classical (first order) models have the tendency to produce spurious segments which are not present in the clean signal (oversegmentation). This well-known shortcoming is called the *gradient limit effect* in literature. We addressed the gradient limit effect by switching to higher orders: *higher order Mumford-Shah models* do not penalize deviations from constant signals, but deviations from polynomial signals within segments. Analogously, solutions of *higher order Potts models* are not piecewise constant but piecewise polynomial. We showed that the solutions of the models are unique for almost all input data. Towards a solver, we noted that the higher order models can be solved via dynamic programming just like the first order models. However, the existing approaches are either too slow or become numerically unstable when applied to higher order models. Hence, we developed a fast solver for higher order Mumford-Shah and Potts models, which uses recurrence relations based on Givens rotations. We further obtained stability results and proved that the proposed algorithm has a quadratic worst case complexity. In numerical experiments, we demonstrated the benefits and limitations of higher order Mumford-Shah and Potts models. We further observed that practical runtimes can be improved to linear ones, when including suitable pruning techniques.

In Chapter 4, we considered the *inverse Potts model* for joint reconstruction and partitioning of images which are given indirectly in terms of measured data only. In contrast to the previous situation of univariate data given directly, the underlying minimization problem is NP-hard here. We developed two iterative minimization strategies for the problem. The first approach addressed a (still NP-hard) quadratic penalty relaxation of the problem. It was based on solving a sequence of easier to handle surrogate minimization problems. These surrogate problems decomposed into univariate Potts problems which could be solved by the algorithm presented in Chapter 3. We proved that the algorithm converges to a local minimum of the relaxed problem which is in the range of what can be expected best in view of the NP-hardness of the problem. We proposed a projection procedure which constructs a feasible solution of the original problem from the solutions of the relaxed problem. We further proved that the result of this procedure is close to a minimum of the original problem given the imaging operator is lower bounded. The second approach addressed the (not relaxed) Potts problem by iterating between running the first algorithm for the relaxed problem and updating the relaxation parameters according to a certain rule. We proved that the thereby obtained algorithm converges to a local minimum of the Potts problem (on subsequences). This result is comparable to the existing results for sparsity problems for which the subproblems decompose pointwise. In contrast, the subproblems we had to solve decomposed only linewise. We illustrated the potential of the proposed algorithms in numerical experiments which included direct partitioning of blurred images, reconstruction from undersampled Radon data and classical image partitioning.

In Chapter 5, we considered the reconstruction problem in multi-spectral CT. We noted that the edges in the images of the different energy channels are typically spatially correlated. To this end, we proposed to employ the multi-channel Potts prior. We showed that the multi-channel Potts prior provides a strong channel coupling. Most notably, the edges are enforced to be at the same spatial positions across the channels. We applied the multi-channel Potts prior by minimizing the multi-spectral Potts model with the ADMM strategy proposed in [183] (Potts ADMM). Furthermore, we proposed new reconstruction approaches based on the *superiorization* of the conjugate gradient method (CG). More precisely, the CG iterates were perturbed with non-ascending directions w.r.t. the (block-wise) Potts prior which corresponded to taking steps towards the proximal mapping of the block-wise Potts prior. Superiorizing the CG method in this way yielded improved results compared to the unperturbed CG method. Further, as the used variant of the CG method is strongly perturbation resilient, the resulting algorithm had a theoretical termination guarantee. However, we observed that perturbing the iterates by taking the proximal mapping of the block-wise Potts prior itself produced improved results. Based on this observation, we developed a new Potts superiorization approach – Potts S-CG. Potts S-CG perturbs the CG iterates with proximal mappings and further lets the underlying linear system evolve, so that the final result becomes genuinely piecewise constant. On the algorithmic side, a significant difference between Potts ADMM and Potts S-CG was that Potts ADMM solved a Tikhonov problem in each iteration, while Potts S-CG had to perform a single CG step only. We applied both methods to simulated multi-spectral CT data and compared them to existing TV-based methods. The benefits of Potts ADMM and of Potts S-CG manifested in sharper edges and mostly higher MSSIM values than the TV-type methods.

In Chapter 6, we considered second and higher order Mumford-Shah models for edge preserving image smoothing. Analogously to the univariate case, the classical Mumford-Shah model can exhibit the gradient limit effect when applied to images with color gradients. To overcome the gradient limit effect, we considered the *second order Mumford-Shah model*. Instead of penalizing the squared Euclidean norm of the gradient of the image in each point, the second order Mumford-Shah model penalizes the squared Frobenius norm of the Hessian. As a result, the functional penalizes deviations of the image from affine-linear functions outside its edges (instead of deviations from constant functions). As a first step, we reformulated the second order Mumford-Shah model in terms of Taylor jets which allowed for an explicit modeling of the partial derivatives. After discretization, we obtained a family of discrete jet formulations of the second order Mumford-Shah model. Towards generalizations, we also reformulated general *higher order Mumford-Shah models* in terms of Taylor jets and pointed out the considerations needed to discretize them. We developed an algorithm based on an ADMM splitting approach such that the emerging subproblems decoupled into univariate segmented jet estimation problems. For these subproblems, we derived an efficient solver based on our solver for univariate higher order Mumford-Shah models. In numerical experiments, we illustrated the advantages of the proposed jet-based discrete formulation by comparing it with a (not jet-based) discretization in terms of second order finite differences only. Further, we compared our approach to the existing approaches to the second order Mumford-Shah model qualitatively and quantitatively. Eventually, we showed the potential of the proposed method in connection with edge detection.

Finally, Chapter 7 was concerned with the *affine-linear Potts model* for the image partitioning problem. The classical Potts model approaches the image partitioning problem by segmenting the image domain into segments of constant color intensity. This can lead to oversegmentation, when applied to images with linear trends as the Potts model introduces additional (spurious) segments to account for these trends. A natural way to allow for linear trends is the affine-linear Potts model. Compared with the (classical) Potts model, the result is not constant within the segments but affine-linear. Towards an algorithm, we lifted the problem to the jet space so that the solutions corresponded to piecewise constant Taylor jets. We approached the discrete problem –which was now a piecewise constant problem in terms of Taylor jets– with an ADMM splitting approach. The emerging subproblems decomposed into univariate segmented jet problems for which we derived an efficient solver based on our solver for univariate higher order Potts problems. We conducted a quantitative comparison with the state-of-the-art method based on iterative repartitioning using graph cuts. The proposed method achieves lower mean functional values and is significantly faster.

Future work

An interesting area of future research is the application of univariate higher order Mumford-Shah and Potts models for the joint reconstruction and segmentation of signals given indirectly in terms of measured data. Towards an algorithm, our solver in Chapter 3 could be the basis for a method which minimizes a sequence of surrogate functionals [201] or for an ADMM splitting approach [182]. Further, as data can be corrupted by noise other than Gaussian white noise (e.g., Laplacian noise or impulsive noise), another interesting topic might be the extension to general data terms (e.g., a data term using the ℓ_1-norm) [182].

Similarly, one might extend our approach to the second order Mumford-Shah model in Chapter 6 to other data terms and to indirectly given images, respectively. This has already been done for the classical Mumford-Shah model in [101]. There, the additional challenge of general data terms and forward operators, respectively, was accounted for by solving an additional Tikhonov problem in each iteration. Next, one could consider other smoothing penalties than the squared (weighted) Frobenius norm of the Hessian. Examples for such smoothing penalties are the (entry-wise) ℓ_1-norm or Schatten norms of the Hessian. Furthermore, in Chapter 6 we derived jet formulations of general higher order Mumford-Shah models and discussed on how to derive corresponding discretizations. Deriving algorithmic approaches to higher order models and studying their performance are another interesting topic for future research.

Another interesting topic corresponds to the application of the proposed Potts S-CG method of Chapter 5 to further imaging problems. Examples may be blurred images, MRI, PET, image partitioning or the fusion problem for multimodal measurements. Further, it is an interesting topic to study the approach w.r.t. theoretical aspects. As stated in Remark 5.7, it is an open question whether the perturbations obtained from evaluating the proximal mapping of the block-wise Potts prior correspond to additive non-ascending directions. Finally, we think it might be rather fruitful to use the (second order) Mumford-Shah prior as well as the affine-linear Potts prior in superiorization approaches. The proposed splitting scheme in Chapter 5 might serve as a basis.

A further potential direction of future research corresponds to studying a piecewise quadratic Potts model for the image partitioning problem. In particular, it would be interesting to determine if the extension to quadratic segments could improve upon affine-linear segments, which we considered here. A reformulation in terms of Taylor jets as for the affine-linear Potts model could be a starting point. Analogously to the second order Mumford-Shah model, one could extend the approach of Chapter 7 to indirectly given images and general data terms; such extensions have been considered for the Potts model in [183].

On the theoretical side, an interesting topic of future research is to complement our empirical observation of convergence of the proposed algorithms in Chapters 6 and 7 by a theoretical convergence analysis. A starting point may be [197], which is concerned with the global convergence of the ADMM in non-convex and non-smooth optimization.

A. Appendix

In the following, we provide definitions related to the continuous Mumford-Shah and Potts models.

Definition A.1 (Hausdorff measure, [94]). Let $m < n$. Denoting the diameter $\operatorname{diam}(B) = \sup\{\|x-y\|,\ x, y \in B\}$ of a nonempty set $B \subset \mathbb{R}^n$, we define for a set $A \subset \mathbb{R}^n$,

$$\mathcal{H}^m_\varepsilon(A) = \inf \Big\{ \sum_{i=1}^{\infty} \alpha(m) \Big(\tfrac{1}{2} \operatorname{diam}(B_i) \Big)^m \ : \ A \subset \bigcup_{i=1}^{\infty} B_i,\ B_i \subset \mathbb{R}^n,\ \operatorname{diam}(B_i) < \varepsilon \Big\}, \tag{A.1}$$

where $\alpha(m)$ is the volume of m-dimensional unit sphere and the infimum is taken over all countable covers $(B_i)_{i \in \mathbb{N}}$ of A by sets B_i of diameter strictly smaller than ε. The m-dimensional Hausdorff measure of a set $A \subset \mathbb{R}^n$ is then given by the limit

$$\mathcal{H}^m(A) = \lim_{\varepsilon \to 0} \mathcal{H}^m_\varepsilon(A). \tag{A.2}$$

Please note that the limit in (A.2) exists since $\mathcal{H}^m_\varepsilon(A)$ is monotonically increasing for decreasing values of ε. However, the limit is not necessarily finite (e.g., $m = 1,\ n = 2,\ A = \mathbb{R}^2$).

Definition A.2 (Distributional gradient). For an open subset $\Omega \subset \mathbb{R}^n$ we let $C_0^\infty(\Omega)$ be the space of test functions, i.e., $C_0^\infty(\Omega)$ is the space of infinitely differentiable functions on Ω with compact support. Then the space of distributions is the space of continuous linear functionals $d : C_0^\infty(\Omega) \to \mathbb{R}$, that is, the dual space $C_0^\infty(\Omega)^*$. The gradient of a distribution d is the vector of distributions defined via $\nabla d(\phi) = \Big(-d(\frac{\partial \phi}{\partial x_i})\Big)_{i=1}^n$. Since a locally integrable function $f : \Omega \to \mathbb{R}$ can be understood as a distribution d_f by $d_f(\phi) = \int_\Omega f(x)\phi(x)dx$ for all $\phi \in C_0^\infty(\Omega)$, its distributional gradient is understood as ∇d_f and we write $\nabla f = \nabla d_f$ since the distributional gradient of a differentiable function coincides with its usual gradient.

Definition A.3 (Functions of bounded variation). Let $u \in L^1(\Omega)$ be an integrable function on an open and bounded domain $\Omega \subset \mathbb{R}^2$. The total variation of u in Ω is given by

$$V(u, \Omega) = \sup \left\{ \int_\Omega u(x) \operatorname{div} \phi(x)\, dx : \phi \in C_c^1(\Omega, \mathbb{R}^2),\ \|\phi\|_{L^\infty(\Omega)} \leq 1 \right\}, \tag{A.3}$$

where $\|\cdot\|_{L^\infty(\Omega)}$ is the essential supremum norm. The space of functions of bounded variation $BV(\Omega)$ is defined as those integrable functions on Ω which have finite total variation, i.e.,

$$BV(\Omega) = \{u \in L^1(\Omega) : V(u, \Omega) < \infty\}. \tag{A.4}$$

To fix ideas, if u is continuously differentiable, then: $u \in BV(\Omega)$ if and only if $\int_\Omega \|\nabla u\|_2\, dx < \infty$.

Definition A.4 (Special functions of bounded variation)**.** Let $u \in L^1_{\text{loc}}(\Omega)$ be a locally integrable function on the open domain $\Omega \subset \mathbb{R}^2$. Then u is a special function of bounded variation, i.e., $u \in SBV(\Omega)$, if:

(i) There are measurable functions $f, g : \Omega \to \mathbb{R}^2$ such that $\int_\Omega \|f\| \, d\mathcal{H}^2 + \int_\Omega \|g\| \, d\mathcal{H}^1 < \infty$, where $\mathcal{H}^1, \mathcal{H}^2$ are the one- and two-dimensional Hausdorff measure, respectively.

(ii) The equality $\int_\Omega u \operatorname{div} \phi \, d\mathcal{H}^2 = \int_\Omega \langle \phi, f \rangle \, d\mathcal{H}^2 + \int_\Omega \langle \phi, g \rangle \, d\mathcal{H}^1$ holds for all $\phi \in C^1_c(\Omega, \mathbb{R}^2)$.

We note that $SBV(\Omega)$ is a proper subspace of $BV(\Omega)$.

Bibliography

[1] L. Ambrosio. A compactness theorem for a new class of functions of bounded variation. *Bollettino della Unione Matematica Italiana*, 3(4):857–881, 1989.

[2] L. Ambrosio, L. Faina, and R. March. Variational approximation of a second order free discontinuity problem in computer vision. *SIAM Journal on Mathematical Analysis*, 32(6):1171–1197, 2001.

[3] L. Ambrosio and V. Tortorelli. Approximation of functional depending on jumps by elliptic functional via Γ-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[5] M. Artina, M. Fornasier, and F. Solombrino. Linearly constrained nonsmooth and nonconvex minimization. *SIAM Journal on Optimization*, 23(3):1904–1937, 2013.

[6] I. Auger and C. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54, 1989.

[7] E. Bae, J. Yuan, and X.-C. Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision*, 92(1):112–129, 2011.

[8] L. Bar, T. Chan, G. Chung, M. Jung, N. Kiryati, N. Sochen, and L. Vese. Mumford and Shah model and its applications to image segmentation and image restoration. *Handbook of Mathematical Methods in Imaging*, pages 1–52, 2014.

[9] L. Bar, N. Sochen, and N. Kiryati. Variational pairing of image segmentation and blind restoration. In *Computer Vision - ECCV 2004*, pages 166–177. Springer, 2004.

[10] L. Bar, N. Sochen, and N. Kiryati. Semi-blind image restoration via Mumford-Shah regularization. *IEEE Transactions on Image Processing*, 15(2):483–493, 2006.

[11] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[12] R. Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–516, 1954.

[13] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[14] R. Bellman and R. Roth. Curve fitting by segmented straight lines. *Journal of the American Statistical Association*, 64(327):1079–1084, 1969.

[15] D. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[16] P. Besl and R. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):167–192, 1988.

[17] S. Beucher. Use of watersheds in contour detection. In *Proceedings of the International Workshop on Image Processing*. CCETT, 1979.

[18] J. Bigün and G. Granlund. Optimal orientation detection of linear symmetry. In *Proceedings First International Conference on Computer Vision*, pages 433–438. IEEE Computer Society Press, 1987.

[19] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 489–495, 1999.

[20] A. Blake. Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):2–12, 1989.

[21] A. Blake and A. Zisserman. *Visual reconstruction.* MIT Press Cambridge, 1987.

[22] T. Blumensath and M. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[23] T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[24] T. Boccellari and F. Tomarelli. Generic uniqueness of minimizers for Blake and Zisserman functionals. *Revista Matemática Complutense*, 26(2):361–408, 2013.

[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[26] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[27] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[28] A. Braides, A. Defranceschi, and E. Vitali. A compactness result for a second-order variational discrete model. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(2):389–410, 2012.

[29] K. Bredies and M. Holler. A total variation-based JPEG decompression model. *SIAM Journal on Imaging Sciences*, 5(1):366–393, 2012.

[30] K. Bredies and M. Holler. A TGV-based framework for variational image decompression, zooming, and reconstruction. Part I: Analytics. *SIAM Journal on Imaging Sciences*, 8(4):2814–2850, 2015.

[31] K. Bredies and M. Holler. A TGV-based framework for variational image decompression, zooming, and reconstruction. Part II: Numerics. *SIAM Journal on Imaging Sciences*, 8(4):2851–2886, 2015.

[32] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[33] E. Brown, T. Chan, and X. Bresson. Completely convex formulation of the Chan-Vese image segmentation model. *International Journal of Computer Vision*, 98(1):103–121, 2012.

[34] J. Bruce. Optimum quantization. Technical report, Massachusetts Institute of Technology, 1965.

[35] M. Burger, C. Rossmanith, and X. Zhang. Simultaneous reconstruction and segmentation for dynamic SPECT imaging. *Inverse Problems*, 32(10):104002, 2016.

[36] C. Byrne. What do simulations tell us about superiorization? *Preprint. Available on ResearchGate at: https://www. researchgate. net/publication/336361338 What Do Simulations Tell Us About Superiorization revised-October*, 8, 2019.

[37] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ^1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

[38] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[39] M. Carriero, A. Leaci, and F. Tomarelli. Strong minimizers of Blake and Zisserman functionals. *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, 25(1-2):257–285, 1997.

[40] M. Carriero, A. Leaci, and F. Tomarelli. Euler equations for Blake and Zisserman functionals. *Calculus of Variations and Partial Differential Equations*, 32(1):81–110, 2008.

[41] M. Carriero, A. Leaci, and F. Tomarelli. A candidate local minimizer of Blake and Zisserman functionals. *Journal de Mathématiques Pures et Appliquées*, 96(1):58–87, 2011.

[42] M. Carriero, A. Leaci, and F. Tomarelli. Uniform density estimates for Blake and Zisserman functionals. *Discrete and Continuous Dynamical Systems A*, 31(4):1129–1150, 2011.

[43] M. Carriero, A. Leaci, and F. Tomarelli. A survey on the Blake–Zisserman functional. *Milan Journal of Mathematics*, 83(2):397–420, 2015.

[44] Y. Censor. Superiorization and perturbation resilience of algorithms: a continuously updated bibliography. *arXiv:1506.04219*, 2015.

[45] Y. Censor, E. Garduño, E. Helou, and G. Herman. Derivative-free superiorization: Principle and algorithm. *arXiv:1908.10100*, 2019.

[46] Y. Censor, G. Herman, and M. Jiang. Special issue on superiorization: theory and applications. *Inverse Problems*, 33(4), 2017.

[47] Y. Censor, S. Petra, and C. Schnörr. Superiorization vs. accelerated convex optimization: the superiorized/regularized least squares case. *Journal of Applied and Numerical Optimization*, 2(1):15–62, 2020.

[48] A. Chambolle. Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. *SIAM Journal on Applied Mathematics*, 55(3):827–863, 1995.

[49] A. Chambolle. Finite-differences discretizations of the Mumford-Shah functional. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(02):261–288, 1999.

[50] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal on Imaging Sciences*, 5(4):1113–1158, 2012.

[51] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[52] T. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[53] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[54] R. Chartrand and B. Wohlberg. A nonconvex ADMM algorithm for group sparsity with sparse groups. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6009–6013, 2013.

[55] C. Chen, J. Leng, and G. Xu. A general framework of piecewise-polynomial Mumford-Shah model for image segmentation. *International Journal of Computer Mathematics*, 94(10):1981–1997, 2017.

[56] X. Cheng, M. Zeng, and X. Liu. Feature-preserving filtering with L_0 gradient minimization. *Computers & Graphics*, 38:150–157, 2014.

[57] L. Condat. A direct algorithm for 1-D total variation denoising. *IEEE Signal Processing Letters*, 20(11):1054–1057, 2013.

[58] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.

[59] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[60] R. Davidi, G. Herman, and Y. Censor. Perturbation-resilient block-iterative projection methods with application to image reconstruction from projections. *International Transactions in Operational Research*, 16(4):505–524, 2009.

[61] E. De Giorgi. Free discontinuity problems in calculus of variations. *Frontiers in pure and applied Mathematics, a collection of papers dedicated to J.L. Lions on the occasion of his 60th birthday, R. Dautray ed., North Holland*, 1991.

[62] A. Delong, L. Gorelick, F. Schmidt, O. Veksler, and Y. Boykov. Interactive segmentation with super-labels. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 147–162, 2011.

[63] A. Delong, A. Osokin, H. Isack, Y. Boykov, A. Delong, H. Isack, Y. Boykov, and A. Osokin. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96:1–27, 2012.

[64] P. Dembowski. *Finite Geometries.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1968.

[65] Q. Ding, T. Niu, X. Zhang, and Y. Long. Image-domain multimaterial decomposition for dual-energy CT based on prior information of material images. *Medical Physics*, 45(8):3614–3626, 2018.

[66] D. Donoho and I. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[67] A. Drobyshev, C. Machka, M. Horsch, M. Seltmann, V. Liebscher, M. de Angelis, and J. Beckers. Specificity assessment from fractionation experiments (SAFE): a novel method to evaluate microarray probe specificity based on hybridisation stringencies. *Nucleic Acids Research*, 31(2):1–10, 2003.

[68] A. Drory, C. Haubold, S. Avidan, and F. Hamprecht. Semi-global matching: a principled derivation in terms of message passing. In *Pattern Recognition*, pages 43–53. Springer, 2014.

[69] N. Ducros, J. Abascal, B. Sixou, S. Rit, and F. Peyrin. Regularization of nonlinear decomposition of spectral x-ray projection images. *Medical Physics*, 44(9):e174–e187, 2017.

[70] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J. Olivo-Marin, and C. Zimmer. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Transactions on Image Processing*, 14(9):1396–1410, 2005.

[71] M. Ehrhardt and M. Betcke. Multicontrast MRI reconstruction with structure-guided total variation. *SIAM Journal on Imaging Sciences*, 9(3):1084–1106, 2016.

[72] H. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Springer, 1996.

[73] P. Fearnhead, R. Maidstone, and A. Letchford. Detecting changes in slope with an L_0 penalty. *Journal of Computational and Graphical Statistics*, 28(2):265–275, 2019.

[74] M. Foare, N. Pustelnik, and L. Condat. Semi-linearized proximal alternating minimization for a discrete Mumford–Shah model. *IEEE Transactions on Image Processing*, 29(1):2176–2189, 2019.

[75] M. Fornasier and R. Ward. Iterative thresholding meets free-discontinuity problems. *Foundations of Computational Mathematics*, 10(5):527–567, 2010.

[76] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305. Interlaken, 1987.

[77] D. Fortun, M. Storath, D. Rickert, A. Weinmann, and M. Unser. Fast piecewise-affine motion estimation without segmentation. *IEEE Transactions on Image Processing*, 27(11):5612–5624, 2018.

[78] K. Frick, A. Munk, and H. Sieling. Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):495–580, 2014.

[79] F. Friedrich, A. Kempe, V. Liebscher, and G. Winkler. Complexity penalized M-estimation: fast computation. *Journal of Computational and Graphical Statistics*, 17(1):201–224, 2008.

[80] K.-S. Fu and J. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1):3–16, 1981.

[81] D. Gabay and B. Mercier. *A dual algorithm for the solution of non linear variational problems via finite element approximation.* Institut de recherche d'informatique et d'automatique, 1975.

[82] H. Gao, H. Yu, S. Osher, and G. Wang. Multi-energy CT based on a prior rank, intensity and sparsity model (PRISM). *Inverse Problems*, 27(11):115012, 2011.

[83] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[84] W. Gentleman. Least squares computations by Givens transformations without square roots. *IMA Journal of Applied Mathematics*, 12:329–336, 1973.

[85] R. Giryes, M. Elad, and A. Bruckstein. Sparsity based methods for overparameterized variational problems. *SIAM Journal on Imaging Sciences*, 8(3):2133–2159, 2015.

[86] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.

[87] B. Goldluecke, E. Strekalovskiy, and D. Cremers. Tight convex relaxations for vector-valued labeling. *SIAM Journal on Imaging Sciences*, 6(3):1626–1664, 2013.

[88] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[89] B. Gonzales and D. Lalush. Full-spectrum CT reconstruction using a weighted least squares algorithm with an energy-axis penalty. *IEEE Transactions on Medical Imaging*, 30(2):173–183, 2010.

[90] R. Gonzalez and R. Woods. *Digital Image Processing.* Pearson, 2018.

[91] P. Hansen and M. Saxild-Hansen. AIR tools—a Matlab package of algebraic iterative reconstruction methods. *Journal of Computational and Applied Mathematics*, 8(236):2167–2178, 2012.

[92] R. Haralick and L. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132, 1985.

[93] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[94] F. Hausdorff. Dimension und äußeres Maß. *Mathematische Annalen*, 79(1-2):157–179, 1918.

[95] B. Heismann, B. Schmidt, and T. Flohr. *Spectral Computed Tomography.* SPIE Bellingham, WA, 2012.

[96] E. Helou, G. Herman, C. Lin, and M. Zibetti. Superiorization of preconditioned conjugate gradient algorithms for tomographic image reconstruction. *arXiv:1807.10151*, 2018.

[97] G. Herman, E. Garduño, R. Davidi, and Y. Censor. Superiorization: an optimization heuristic for medical physics. *Medical Physics*, 39(9):5532–5546, 2012.

[98] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814. IEEE, 2005.

[99] H. Hirschmüller. Stereo vision in structured environments by consistent semi-global matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2386–2393. IEEE, 2006.

[100] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[101] K. Hohm, M. Storath, and A. Weinmann. An algorithmic framework for Mumford-Shah regularization of inverse problems in imaging. *Inverse Problems*, 31(11):115011, 2015.

[102] M. Hong, Z.-Q. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[103] T. Hotz, O. Schutte, H. Sieling, T. Polupanow, U. Diederichsen, C. Steinem, and A. Munk. Idealizing ion channel recordings by a jump segmentation multiresolution filter. *IEEE Transactions on NanoBioscience*, 12(4):376–386, 2013.

[104] P. Hupé, N. Stransky, J. Thiery, F. Radvanyi, and E. Barillot. Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20(18):3413–3422, 2004.

[105] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal on Computer Vision*, 97:123–147, 2012.

[106] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[107] B. Jackson, J. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, 2005.

[108] M. Jiang, P. Maass, and T. Page. Regularizing properties of the Mumford-Shah functional for imaging applications. *Inverse Problems*, 30(3):035007, 2014.

[109] C. Joo, H. Balci, Y. Ishitsuka, C. Buranachai, and T. Ha. Advances in single-molecule fluorescence methods for molecular biology. *Annu. Rev. Biochem.*, 77:51–76, 2008.

[110] F. Jost, P. Peter, and J. Weickert. Compressing piecewise smooth images with the Mumford-Shah cartoon model. *arXiv:2003.05206*, 2020.

[111] O. Juan and Y. Boykov. Active graph cuts. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1023–1029. IEEE, 2006.

[112] Y. Kaloga, M. Foare, N. Pustelnik, and P. Jensen. Discrete Mumford–Shah on graph for mixing matrix estimation. *IEEE Signal Processing Letters*, 26(9):1275–1279, 2019.

[113] Z. Kato and T.-C. Pong. A Markov random field image segmentation model for color textured images. *Image and Vision Computing*, 24(10):1103–1114, 2006.

[114] D. Kazantsev, J. Jørgensen, M. Andersen, W. Lionheart, P. Lee, and P. Withers. Joint image reconstruction method with correlative multi-channel prior for x-ray spectral computed tomography. *Inverse Problems*, 34(6):064001, 2018.

[115] L. Kiefer and S. Petra. Performance bounds for cosparse multichannel signal recovery via collaborative-TV. In *2017 International Conference on Sampling Theory and Applications*, pages 599–603. IEEE, 2017.

[116] L. Kiefer, S. Petra, M. Storath, and A. Weinmann. Multi-channel Potts-based reconstruction for multi-spectral computed tomography. 2020, preprint.

[117] L. Kiefer, M. Storath, and A. Weinmann. An efficient algorithm for the piecewise affine-linear Mumford-Shah model based on a Taylor jet splitting. *IEEE Transactions on Image Processing*, 29:921–933, 2020.

[118] L. Kiefer, M. Storath, and A. Weinmann. Iterative Potts minimization for the recovery of signals with discontinuities from indirect measurements: the multivariate case. *Foundations of Computational Mathematics*, 2020.

[119] L. Kiefer, M. Storath, and A. Weinmann. PALMS image partitioning – a new parallel algorithm for the piecewise affine-linear Mumford-Shah model. *Image Processing On Line*, 2020, accepted.

[120] L. Kiefer, M. Storath, and A. Weinmann. An algorithm for second order Mumford-Shah models based on a Taylor jet formulation. 2020, preprint.

[121] R. Killick, P. Fearnhead, and I. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[122] J. Kim, A. Tsai, M. Cetin, and A. Willsky. A curve evolution-based variational approach to simultaneous image restoration and segmentation. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages I–109. IEEE, 2002.

[123] E. Klann. A Mumford-Shah-like method for limited data tomography with an application to electron tomography. *SIAM Journal on Imaging Sciences*, 4(4):1029–1048, 2011.

[124] E. Klann, R. Ramlau, and W. Ring. A Mumford-Shah level-set approach for the inversion and segmentation of SPECT/CT data. *Inverse Problems and Imaging*, 5(1):137–166, 2011.

[125] J. Kleinberg and E. Tardos. *Algorithm Design.* Pearson Education India, 2006.

[126] V. Kolmogorov, T. Pock, and M. Rolinek. Total variation on a tree. *SIAM Journal on Imaging Sciences*, 9(2):605–636, 2016.

[127] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[128] G. Lan and R. Monteiro. Iteration-complexity of first-order penalty methods for convex programming. *Mathematical Programming*, 138(1-2):115–139, 2013.

[129] L. Landweber. An iteration formula for Fredholm integral equations of the first kind. *American Journal of Mathematics*, 73(3):615–624, 1951.

[130] E. Larson and D. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006, 2010.

[131] F. Lauze, Y. Quéau, and E. Plenge. Simultaneous reconstruction and segmentation of CT scans with shadowed data. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 308–319. Springer, 2017.

[132] P. Le Callet and F. Autrusseau. Subjective quality assessment IRCCyN/IVC database, 2005. http://www.irccyn.ec-nantes.fr/ivcdb/.

[133] S. Lefkimmiatis, A. Bourquard, and M. Unser. Hessian-based norm regularization for image restoration with biomedical applications. *IEEE Transactions on Image Processing*, 21(3):983–995, 2012.

[134] J. Lellmann and C. Schnörr. Continuous multiclass labeling approaches and algorithms. *SIAM Journal on Imaging Sciences*, 4(4):1049–1096, 2011.

[135] D. Lemire. A better alternative to piecewise linear time series segmentation. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 545–550. SIAM, 2007.

[136] G. Li and T. Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

[137] M. Little and N. Jones. Generalized methods and solvers for noise removal from piecewise constant signals. I. Background theory. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 467(2135):3088–3114, 2011.

[138] M. Little and N. Jones. Generalized methods and solvers for noise removal from piecewise constant signals. II. New methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 467(2135):3115–3140, 2011.

[139] W. Liu, W. Xu, X. Chen, X. Huang, C. Shen, and J. Yang. Edge-preserving piecewise linear image smoothing using piecewise constant filters. *arXiv:1801.06928*, 2018.

[140] X. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994.

[141] Y. Long and J. Fessler. Multi-material decomposition using statistical image reconstruction for spectral CT. *IEEE Transactions on Medical Imaging*, 33(8):1614–1626, 2014.

[142] Z. Lu. Iterative hard thresholding methods for l_0 regularized convex cone programming. *Mathematical Programming*, 147(1-2):125–154, 2014.

[143] Z. Lu and Y. Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization*, 23(4):2448–2478, 2013.

[144] S. Mallat. *A Wavelet Tour of Signal Processing: the Sparse Way*. Academic press, 2008.

[145] J. Marroquin. Surface reconstruction preserving discontinuities. *Artificial Intelligence Lab., Massachusetts Institute of Technology*, 1984.

[146] C. McCollough, S. Leng, L. Yu, and J. Fletcher. Dual-and multi-energy CT: principles, technical approaches, and clinical applications. *Radiology*, 276(3):637–653, 2015.

[147] S. Müller, J. Weickert, and N. Graf. Automatic brain tumor segmentation with a fast Mumford-Shah algorithm. In *Medical Imaging 2016: Image Processing*, volume 9784, page 97842S. International Society for Optics and Photonics, 2016.

[148] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 17, pages 137–154, 1985.

[149] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[150] F. Natterer. *The Mathematics of Computerized Tomography*, volume 32. Siam, 1986.

[151] D. Needell and R. Ward. Near-optimal compressed sensing guarantees for total variation minimization. *IEEE Transactions on Image Processing*, 22(10):3941–3949, 2013.

[152] D. Needell and R. Ward. Stable image reconstruction using total variation minimization. *SIAM Journal on Imaging Sciences*, 6(2):1035–1058, 2013.

[153] M. Ng, P. Weiss, and X. Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. *SIAM Journal on Scientific Computing*, 32(5):2710–2736, 2010.

[154] R. Nguyen and M. Brown. Fast and effective L_0 gradient minimization by region fusion. In *Proceedings of the IEEE ICCV*, pages 208–216, 2015.

[155] M. Nikolova, M. Ng, and C.-P. Tam. Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction. *IEEE Transactions on Image Processing*, 19(12):3073–3088, 2010.

[156] M. Nikolova, M. Ng, S. Zhang, and W. Ching. Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization. *SIAM Journal on Imaging Sciences*, 1(1):2–25, 2008.

[157] A. Nord, E. Gachon, R. Perez-Carrasco, J. Nirody, A. Barducci, R. Berry, and F. Pedaci. Catch bond drives stator mechanosensitivity in the bacterial flagellar motor. *Proceedings of the National Academy of Sciences*, 114(49):12952–12957, 2017.

[158] S. Ono. L_0 gradient projection. *IEEE Transactions on Image Processing*, 26(4):1554–1564, 2017.

[159] X. Pan, E. Sidky, and M. Vannier. Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? *Inverse Problems*, 25(12):123009, 2009.

[160] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[161] D. Pham, C. Xu, and J. Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2(1):315–337, 2000.

[162] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 810–817. IEEE, 2009.

[163] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *IEEE International Conference on Computer Vision*, pages 1133–1140, 2009.

[164] R. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48(1):106–109, 1952.

[165] R. Ramlau and W. Ring. A Mumford-Shah level-set approach for the inversion and segmentation of X-ray tomography data. *Journal of Computational Physics*, 221(2):539–557, 2007.

[166] R. Ramlau and W. Ring. Regularization of ill-posed Mumford–Shah models with perimeter penalization. *Inverse Problems*, 26(11):115001, 2010.

[167] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846, 1971.

[168] D. Rigie and P. La Rivière. Joint reconstruction of multi-channel, spectral CT data via constrained total nuclear variation minimization. *Physics in Medicine & Biology*, 60(5):1741, 2015.

[169] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[170] K. Sauer and C. Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Transactions on Signal Processing*, 41(2):534–548, 1993.

[171] O. Semerci, N. Hao, M. Kilmer, and E. Miller. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. *IEEE Transactions on Image Processing*, 23(4):1678–1693, 2014.

[172] P. Shikhaliev. Energy-resolved computed tomography: first experimental results. *Physics in Medicine & Biology*, 53(20):5595, 2008.

[173] J. Siewerdsen, A. Waese, D. Moseley, S. Richard, and D. Jaffray. Spektr: a computational tool for x-ray spectral analysis and imaging system optimization. *Medical Physics*, 31(11):3057–3067, 2004.

[174] J. Simonoff. *Smoothing methods in statistics.* Springer Science & Business Media, 2012.

[175] A. Snijders, N. Nowak, R. Segraves, et al. Assembly of microarrays for genome-wide measurement of DNA copy number by CGH. *Nature Genetics*, 29:263–264, 2001.

[176] Y. Sowa, A. Rowe, M. Leake, T. Yakushi, M. Homma, A. Ishijima, and R. Berry. Direct observation of steps in rotation of the bacterial flagellar motor. *Nature*, 437(7060):916–919, 2005.

[177] G. Steidl and T. Teuber. Removing multiplicative noise by Douglas-Rachford splitting methods. *Journal of Mathematical Imaging and Vision*, 36(2):168–184, 2010.

[178] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer Science & Business Media, 2013.

[179] M. Storath, L. Kiefer, and A. Weinmann. Smoothing for signals with discontinuities using higher order Mumford–Shah models. *Numerische Mathematik*, 143(2):423–460, 2019.

[180] M. Storath, D. Rickert, M. Unser, and A. Weinmann. Fast segmentation from blurred data in 3D fluorescence microscopy. *IEEE Transactions on Image Processing*, 26(10):4856–4870, 2017.

[181] M. Storath and A. Weinmann. Fast partitioning of vector-valued images. *SIAM Journal on Imaging Sciences*, 7(3):1826–1852, 2014.

[182] M. Storath, A. Weinmann, and L. Demaret. Jump-sparse and sparse recovery using Potts functionals. *IEEE Transactions on Signal Processing*, 62(14):3654–3666, 2014.

[183] M. Storath, A. Weinmann, J. Frikel, and M. Unser. Joint image reconstruction and segmentation using the Potts model. *Inverse Problems*, 31(2):025003, 2015.

[184] M. Storath, A. Weinmann, and M. Unser. Jump-penalized least absolute values estimation of scalar or circle-valued signals. *Information and Inference*, 6(3):225–245, 2017.

[185] E. Strekalovskiy, A. Chambolle, and D. Cremers. A convex representation for the vectorial Mumford-Shah functional. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1712–1719, 2012.

[186] E. Strekalovskiy and D. Cremers. Real-time minimization of the piecewise smooth Mumford-Shah functional. In *European Conference on Computer Vision*, pages 127–141. Springer, 2014.

[187] J. Toivanen, A. Meaney, S. Siltanen, and V. Kolehmainen. Joint reconstruction in low dose multi-energy CT. *Inverse Problems and Imaging*, 14(4):607–629, 2020.

[188] J. Tuszynski. 2006-2016 PhotonAttenuation–Software for modeling of photons passing through different materials. https://uk.mathworks.com/matlabcentral/fileexchange/12092-photonattenuation.

[189] W. Van Aarle, W. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, J. Batenburg, and J. Sijbers. The ASTRA toolbox: a platform for advanced algorithm development in electron tomography. *Ultramicroscopy*, 157:35–47, 2015.

[190] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Cornell University, 1999.

[191] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[192] C. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.

[193] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[194] F. Wang, W. Cao, and Z. Xu. Convergence of multi-block Bregman ADMM for nonconvex composite problems. *Science China Information Sciences*, 61(12):122101, 2018.

[195] F. Wang, Z. Xu, and H.-K. Xu. Convergence of Bregman alternating direction method with multipliers for nonconvex composite problems. *arXiv:1410.8625*, 2014.

[196] Y. Wang, S. Xiang, C. Pan, L. Wang, and G. Meng. Level set evolution with locally linear classification for image segmentation. *Pattern Recognition*, 46(6):1734–1746, 2013.

[197] Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

[198] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[199] J. Weickert. *Anisotropic Diffusion in Image Processing*, volume 1. Teubner Stuttgart, 1998.

[200] A. Weinmann, L. Demaret, and M. Storath. Mumford-Shah and Potts regularization for manifold-valued data. *Journal of Mathematical Imaging and Vision*, 55(3):428–445, 2016.

[201] A. Weinmann and M. Storath. Iterative Potts and Blake-Zisserman minimization for the recovery of functions with discontinuities from indirect measurements. *Proceedings of the Royal Society A*, 471(2176), 2015.

[202] A. Weinmann, M. Storath, and L. Demaret. The L^1-Potts functional for robust jump-sparse reconstruction. *SIAM Journal on Numerical Analysis*, 53(1):644–673, 2015.

[203] Z. Wen, C. Yang, X. Liu, and S. Marchesini. Alternating direction methods for classical and ptychographic phase retrieval. *Inverse Problems*, 28(11):115010, 2012.

[204] E. Whittaker. On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75, 1922.

[205] J. Wilkinson. *The Algebraic Eigenvalue Problem.* Monographs on numerical analysis. Clarendon Press, Oxford, 1965.

[206] G. Winkler and V. Liebscher. Smoothers for discontinuous signals. *Journal of Nonparametric Statistics*, 14(1-2):203–222, 2002.

[207] G. Winkler, O. Wittich, V. Liebscher, and A. Kempe. Don't shed tears over breaks. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 107:57–87, 2005.

[208] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via L_0 gradient minimization. *ACM Transactions on Graphics (TOG)*, 30(6):174, 2011.

[209] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *International Conference on Machine Learning*, pages 1669–1678, 2015.

[210] L. Xu, S. Zheng, and J. Jia. Unnatural L_0 sparse representation for natural image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013.

[211] Z. Xu, S. De, M. Figueiredo, C. Studer, and T. Goldstein. An empirical study of ADMM for nonconvex problems. *arXiv:1612.03349*, 2016.

[212] J. Yang and H. Li. Dense, accurate optical flow estimation with piecewise parametric model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1027, 2015.

[213] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33(1):250–278, 2011.

[214] L. Yang, T. Pong, and X. Chen. Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. *SIAM Journal on Imaging Sciences*, 10(1):74–110, 2017.

[215] M. Zanetti and L. Bruzzone. Piecewise linear approximation of vector-valued images and curves via second-order variational model. *IEEE Transactions on Image Processing*, 26(9):4414–4429, 2017.

[216] M. Zanetti, V. Ruggiero, and M. Miranda Jr. Numerical minimization of a second-order functional for image segmentation. *Communications in Nonlinear Science and Numerical Simulation*, 36:528–548, 2016.

[217] Y. Zhang, B. Dong, and Z. Lu. ℓ_0 minimization for wavelet frame based image restoration. *Mathematics of Computation*, 82(282):995–1015, 2013.

[218] M. Zibetti, C. Lin, and G. Herman. Total variation superiorized conjugate gradient method for image reconstruction. *Inverse Problems*, 34(3):034001, 2018.

[219] C. Zimmer and J. Olivo-Marin. Coupled parametric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1838–1842, 2005.

[220] M. Zisler, F. Åström, S. Petra, and C. Schnörr. Image reconstruction by multilabel propagation. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 247–259. Springer, 2017.