IOP Publishing Inverse Problems

Inverse Problems **31** (2015) 115011 (30pp) doi:10.1088/0266-5611/31/11/115011

An algorithmic framework for Mumford–Shah regularization of inverse problems in imaging

Kilian Hohm[1], **Martin Storath**[2,4] **and Andreas Weinmann**[1,3]

[1] Department of Mathematics, Technische Universität München, Germany
[2] Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Switzerland
[3] Research Group Fast Algorithms for Biomedical Imaging, Helmholtz Zentrum München, Germany

E-mail: kilian.hohm@tum.de, martin.storath@epfl.ch and andreas.weinmann@tum.de

Received 11 March 2015, revised 10 July 2015
Accepted for publication 16 July 2015
Published 29 October 2015

Abstract

The Mumford–Shah model is a very powerful variational approach for edge preserving regularization of image reconstruction processes. However, it is algorithmically challenging because one has to deal with a non-smooth and non-convex functional. In this paper, we propose a new efficient algorithmic framework for Mumford–Shah regularization of inverse problems in imaging. It is based on a splitting into specific subproblems that can be solved exactly. We derive fast solvers for the subproblems which are key for an efficient overall algorithm. Our method neither requires *a priori* knowledge of the gray or color levels nor of the shape of the discontinuity set. We demonstrate the wide applicability of the method for different modalities. In particular, we consider the reconstruction from Radon data, inpainting, and deconvolution. Our method can be easily adapted to many further imaging setups. The relevant condition is that the proximal mapping of the data fidelity can be evaluated a within reasonable time. In other words, it can be used whenever classical Tikhonov regularization is possible.

Keywords: Mumford–Shah functional, image reconstruction, inverse problem, ADMM, dynamic programming, computed tomography, deconvolution

(Some figures may appear in colour only in the online journal)

[4] Author to whom any correspondence should be addressed.

1. Introduction

The image formation processes of many imaging modalities, such as computed tomography, positron emission tomography, and photoacoustic tomography, are ill-posed inverse problems. This means that even small perturbations in the data may cause large errors in the reconstruction. Therefore, the reconstruction process needs regularization. It is common to stabilize the inversion of the imaging operator via a variational approach: an energy functional consisting of a data fidelity term and a regularizing term is minimized. The data term is typically determined by the image acquisition device, i.e., by the imaging operator and by the type of noise. The regularizing term depends on the characteristics of the underlying image. It is well known that classical linear regularization—based on Hilbert space semi-norms-performs poorly when the underlying image has discontinuities. Much better results are obtained using the Mumford–Shah functional which is given by

$$\min_{u,C} \gamma \operatorname{length}(C) + \alpha \int_{\Omega \setminus C} |\nabla u|^q \mathrm{d}x + d(H(u), f). \tag{1}$$

Here, H is the (not necessarily linear) imaging operator, and the (quasi-)metric d measures the fidelity to the acquired data f. Often, d is the pth power of an L^p-distance $d(x, y) = \| x - y \|_p^p$. In contrast to Tikhonov-type priors, the Mumford–Shah prior penalizes the variation (measured in an L^q norm) of the argument only for the complement of a discontinuity set C. Furthermore, the 'length', i.e., the (outer) one-dimensional Hausdorff measure of this discontinuity set is penalized. The parameters $\gamma > 0$ and $\alpha > 0$ control the weight of the length penalty and the variation penalty, respectively. Closely related to (1) is the piecewise constant Mumford–Shah functional, often called the Potts functional. It corresponds to the degenerate case $\alpha = \infty$ which amounts to (1) with the second summand removed and restricted to piecewise constant priors.

Problem (1) was studied by Mumford and Shah in their seminal work [32, 33] for $H = \mathrm{id}$ using quadratic variation, and the L^2 data fidelity term. Its discrete counterpart was investigated by Blake and Zisserman [7], and the discrete piecewise constant variant by Geman and Geman [22]. Mumford–Shah functionals with L^q variation penalty have been considered by Fornasier and Ward [20]. Further (more classical) references are [2, 3, 12]. More recently, Mumford–Shah priors have gained a lot of interest for stabilization of inverse problems. The regularizing properties (in the sense of inverse problems) of Mumford–Shah functionals have been studied by Rondi [40] and by Jiang *et al* [24]. In addition the latter paper also investigated the regularizing properties with respect to the corresponding discontinuity sets. As related work, we also mention [19]. Potts functionals have been studied by Ramlau and Ring [37, 38]. In general, it turns out that additional (mild) conditions are needed for the Mumford–Shah approach to be a regularizing method. Such conditions are, e.g., pointwise boundedness, compactness of the operator, and/or additional assumptions on the regularity of the edge set such as perimeter constraints. We refer to [24, 38, 40] for a more detailed discussion. It is worth mentioning that, without additional assumptions, even the the existence of minimizers of (1) is not guaranteed in a continuous domain setting for general data terms [19, 20, 38, 41]. Such situations are not pathological as they appear even in deconvolution problems. However, with suitable assumptions, the existence of minimizers, stability, and an *a priori* parameter choice rule are available.

The Mumford–Shah problem is algorithmically challenging, because it deals with a nonsmooth and nonconvex functional. Finding a minimizer has turned out to be an NP hard problem [1]; thus, there is no hope of finding global minimizers within a reasonable time. Therefore, the goal is to find approximative strategies that give satisfactory results in practice.

For the classical image denoising setup—where $H = \mathrm{id}$ and d is the squared L^2 distance—there are various algorithmic approaches. Examples are graduated non-convexity [7], approximation by elliptic functionals [3], graph cuts [10], active contours [16, 47], and convex relaxations [36, 46], to mention only a few. Algorithms for the related Potts functional with imaging operator $H \neq \mathrm{id}$ have been recently studied in a series of tomographic problems [27–29, 37, 45], deconvolution [44], and optical flow estimation [11]. In contrast, it seems that only a few articles consider algorithms for the (piecewise smooth) Mumford–Shah problem in inverse problems: Bar *et al* [5] consider an Ambrosio–Tortorelli-type approximation based on elliptic functionals for deconvolution problems; Kim *et al* [26] use a level-set based active contour method; Rondi and Santosa [41] use the Ambrosio–Tortorelli approach for electrical impedance tomography; for general linear inverse problems with an L^2 data term, Fornasier and Ward [20] propose an iterative thresholding algorithm.

In this paper, we propose an algorithmic framework for (piecewise smooth) Mumford–Shah regularization of imaging problems. We first discuss proper discretizations of (1) that give an accurate approximation of the Euclidean length term. We base our discretization on a finite difference scheme introduced by Chambolle [13] and the refinements of the weights derived in [43, 45]. Then we show that the discrete problem has a minimizer. The key contribution of this paper is on the algorithmic side: we derive a splitting algorithm that (i) is computationally feasible, (ii) yields high quality reconstructions in practice, and (iii) is very flexible in the sense that it can be easily adapted to various imaging modalities. To this end, we employ an ADMM strategy initially considered for Potts problems in [43–45] to split the discretized problem into specific subproblems which we can solve efficiently. One of the subproblems is a standard evaluation of the proximal mapping with respect to the data term (also known as Tikhonov regularization). The remaining subproblems turn out to basically consist of univariate (discrete) Mumford–Shah or, synonymously, Blake–Zisserman problems. In the corresponding Potts setup, there are fast solvers of complexity $\mathcal{O}(n^2)$ with respect to time and $\mathcal{O}(n)$ with respect to memory [21]. In contrast, even for the case of the Blake–Zisserman problem of quadratic variation ($q = 2$), the presently available solvers are rather slow; there were only $\mathcal{O}(n^3)$ algorithms (w.r.t. time) available. Because of calling these routines $\mathcal{O}(n)$ times in each iteration, this becomes a significant bottleneck. Even for not too large images, it limits applicability tremendously, and for larger data, it makes application even infeasible. To overcome this problem, we propose exact and efficient solvers for the practically most relevant cases of quadratic variation ($q = 2$) and ordinary variation ($q = 1$). Our solver for the first case is based on a combination of the dynamic programming concept of [21, 50] with the efficient spline energy computation of [6] and a pruning strategy [25, 43]. Its worst case complexity is $\mathcal{O}(n^2)$ with respect to time and $\mathcal{O}(n)$ with respect to memory. We emphasize that, to our knowledge, this is the first exact solver of this complexity. Our solver for the latter case is based on the taut string algorithm [17, 30]. Although its worst case complexity is cubic in time and linear in memory, we obtain reasonable runtimes in practice which is due to a speedup strategy we propose. We observe that our method converges. We note that the gray-levels (or the color-levels) of a solution do not have to be fixed *a priori*. Furthermore, there are no restrictions on the discontinuity set; in particular, there is no restriction on the number of connected components. Finally, we do not require an initial guess of the solution or of the edge set. Numerical experiments demonstrate that our method achieves comparable or slightly better reconstruction quality than the Ambrosio–Tortorelli approximation [3, 5, 41] within significantly less computation time. Another major advantage of our framework is that it can be easily adapted to various imaging setups. We only need an implementation of the proximal mapping with respect to the data term. This in turn is a standard task in imaging problems because the evaluation of the proximal mapping is equivalent to classical Tikhonov regularization. We demonstrate the wide applicability of our method to several classical image recovery setups: we consider the reconstruction from Radon data, image inpainting, and deconvolution.

The paper is organized as follows. In section 2 we present our algorithmic framework for the Mumford–Shah problem. In section 3, we apply the developed method to various imaging modalities.

2. Proposed algorithmic framework for the Mumford–Shah problem

We first discuss finite difference discretizations of the Mumford–Shah functional, and we prove existence of minimizers for these discretizations (section 2.1). The splitting scheme is explained in section 2.2. We derive efficient solvers for the arising univariate Mumford–Shah problems in section 2.3, and we comment on the evaluation of the arising proximal mapping in section 2.4. In section 2.5, we show convergence of the algorithm. An extension for vector-valued images is provided in section 2.6. We summarize the complete algorithm and discuss the parameters in section 2.7.

2.1. Discretization of the Mumford–Shah functional

The Mumford–Shah functional is closely related to the discrete weak membrane energy proposed by Blake and Zisserman [7]. For $u \in \mathbb{R}^{m\times n}$, this energy is given by

$$u \mapsto \sum_{i,j} \min\left(\gamma,\, \alpha \left|u_{i+1,j} - u_{ij}\right|^2\right) + \sum_{i,j} \min\left(\gamma,\, \alpha \left|u_{i,j+1} - u_{ij}\right|^2\right) + d(H(u), f).$$

In fact, when refining the grid, these functionals Γ-converge to a variant of the Mumford–Shah functional that measures the length of the discontinuity set in terms of the ℓ^1 analogue of the Hausdorff measure [12]. In particular, if the discontinuity set is a differentiable curve c, the length is given in terms of the Manhattan metric

$$\int_0^1 |c_1'(t)| + |c_2'(t)|\, \mathrm{d}t$$

instead of the desired Euclidean length $\int_0^1 (c_1'(t)^2 + c_2'(t)^2)^{1/2}\, \mathrm{d}t$. The practical consequences are undesired block artifacts in the reconstruction (geometric staircasing). We next discuss how to avoid such anisotropy effects.

Our starting point is a more general discretization of the form

$$\min_{u\in\mathbb{R}^{m\times n}} \sum_{s=1}^{S} \omega_s \phi\left(\nabla_{a_s} u\right) + d(H(u), f), \tag{2}$$

where ϕ is a truncated q-norm given by

$$\phi(x) = \phi_{\alpha,\gamma}(x) = \sum_{i,j} \min\left(\gamma,\, \alpha \left|x_{ij}\right|^q\right).$$

(See figure 1 for an illustration of the potential functions for $q = 1, 2$.) The symbol ∇_a denotes finite differences with respect to the displacement vector a so that $\nabla_a u = u(\,\cdot + a) - u$, where $a \in \mathbb{Z}^2 \backslash \{0\}$. The displacement vectors a belong to a neighborhood system $\mathcal{N} = \{a_1, \ldots, a_S\}$ where $S \geqslant 2$, and $\omega_1, \ldots, \omega_S$ are non-negative weights. The classical anisotropic discretization corresponds to a neighborhood system made

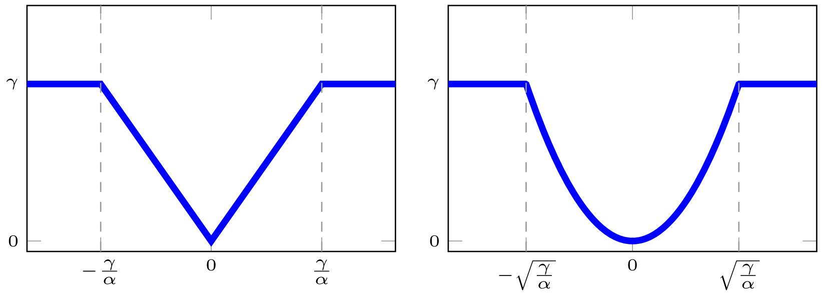


Figure 1. Truncated linear function for $q = 1$ (left) and truncated quadratic function for $q = 2$ (right).

up of the two canonical basis vectors of $\mathbb{R}^2$ along with unit weights. We here use the system

$$\mathcal{N} = \{(1,\,0),\,(0,\,1),\,(1,\,1),\,(1,\,-1),\,(-2,\,-1),\,(-2,\,1),\,(2,\,1),\,(2,\,-1)\}. \tag{3}$$

It contains finite differences in axial, diagonal, and 'knight-move' (referring to the moves of a knight in chess) directions. The discretization (2) with the above neighborhood system $\mathcal{N}$ was first used by Chambolle [13]. We here use the weights

$$\omega_s = \begin{cases} \sqrt{5} - 2, & \text{for } s = 1,\,2,\\ \sqrt{5} - \frac{3}{2}\sqrt{2}, & \text{for } s = 3,\,4,\\ \frac{1}{2}\left(1 + \sqrt{2} - \sqrt{5}\right), & \text{for } s = 5,\,\ldots,\,8. \end{cases} \tag{4}$$

which have been derived in [45]. We briefly discuss the gain of this discretization. First note that a neighborhood system along with the weights gives rise to a norm $\| a \|_{\mathcal{N}}$ given for $a \in \mathbb{R}^2$ by $\| a \|_{\mathcal{N}} = \sum_{s=1}^{S} \omega_s |\langle a,\, a_s\rangle|$. The above weights are constructed such that $\| a_s \|_{\mathcal{N}}$ coincides with the Euclidean norm $\| a_s \|_2$ for all vectors a_s in the neighborhood system [45]. (See figure 2 for an illustration of the unit circle.) With these weights, the ratio of the longest and the shortest unit vector is only about 1.03. Thus, this discretization can be considered as almost isotropic. For comparison, the ratio is $\sqrt{2}$ for the anisotropic discretization.

Next, we show that the considered (discrete) Mumford–Shah problem (2) actually has a minimizer when assuming suitable conditions. This extends the results of [20] derived for ℓ^2 data terms using other methods.

Theorem 2.1. *We assume that either the data fidelity term is of the form* $d(H(u), f) = \| Hu - f \|_p^p$ *for some* $p \in (0,\,\infty]$ *with a (possibly non injective) linear operator H, or, for a (possibly nonlinear) operator H and a general pseudo-metric d, that the coercivity condition* $d(H(u),\, g) \to \infty$ *for* $\| u \| \to \infty$ *is fulfilled for some g. Then the discrete Mumford–Shah problem (2) has a minimizer.*

Proof. We start with the second kind of assumption of the theorem and assume that the coercivity condition in the theorem is fulfilled for the (possibly nonlinear) operator H. Then (2) is coercive which implies that the corresponding sublevel sets are compact. Together with the continuity of the functional (2), this implies the existence of minimizers.

Now we consider the first kind of assumption of the theorem. This means that we consider a linear operator H and the distance $d(H(u), f) = \| Hu - f \|_p^p$. (We note that we do

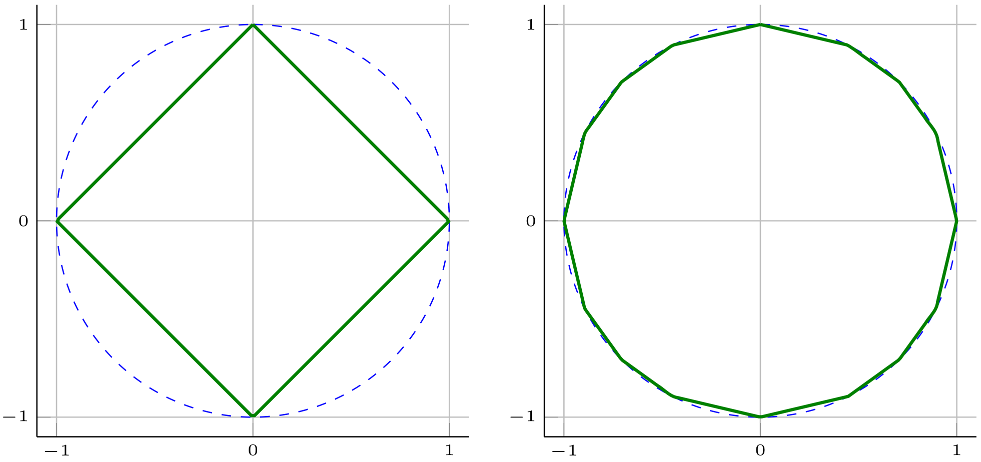


Figure 2. Unit circles of the norm induced by the anisotropic discretization (left, green solid line) and the near-isotropic discretization used in this paper (right, green solid line). In the anisotropic case, the length of the diagonals is much shorter than the length of the compass directions. The near-isotropic discretization induces a close approximation to the Euclidean unit circle.

not assume coercivity in this case.) Then the situation becomes more involved and we need some preparation. We consider a neighborhood system $\mathcal{N} = \{a_1, \ldots, a_S\}$ and use the notation r to denote a member of this system. Then we consider tuples (v, r) where $v = (i, j)$ is a coordinate in the image and $v + r$ is also a valid coordinate. Let $\mathcal{I}$ be the set of all such tuples. (Calling these objects tuples is a slight abuse of notation which causes no issues in the following.) For a subset $\mathcal{G} \subset \mathcal{I}$, we consider the problem:

$$\begin{aligned}&\min_{u\in\mathbb{R}^{m\times n}} F_{\mathcal{G}}(u)\\ &\text{subject to } |\nabla_{a_s}u(v)| \leqslant \gamma' \text{ for all } (v, a_s) \in \mathcal{G},\ s \in \{1, \ldots, S\},\end{aligned} \tag{5}$$

where

$$F_{\mathcal{G}}(u) = \sum_{s=1}^{S} \omega_s \psi_s\left(\nabla_{a_s}u\right) + \|Hu - f\|_p^p \tag{6}$$

and

$$\psi_s(x) = \sum_{\{v=(i,j):(v,a_s)\in\mathcal{G}_s\}} \alpha|x_{ij}|^q, \quad \text{and} \quad \gamma' = (\gamma/\alpha)^{1/q}.$$

Here $\mathcal{G}_s \subset \mathcal{G}$ are those tuples (v, r) in $\mathcal{G}$ having direction $r = a_s$. This means that we only consider members of the set $\mathcal{G} = \bigcup_s \mathcal{G}_s$ as indices for the sums above.

We first deal with the existence of minimizers of (5). To this end, we notice that the functional $F_{\mathcal{G}}$ in (6) consists of a sum of concatenations of affine-linear operators defined on the reconstruction domain $\mathbb{R}^{m\times n}$ with quasinorms. More precisely, the data term $d(Hu, f) = \| Hu - f\|_p^p = \sum_k |Hu_k - f_k|^p$, $p \in (0, \infty]$, is the concatenation of the linear-affine operator $u \mapsto Hu - f$ mapping from the image domain to the data space $\mathbb{R}^l$ (where l stands for the number of measurements) with the quasinorm $\| \cdot \|_p^p$, $p > 0$, defined on $\mathbb{R}^l$. The regularizing term consists of the sum of S terms, each of them being the concatenation of a linear directional difference operator $u \mapsto \nabla_{a_s}u$ mapping from the reconstruction domain

$\mathbb{R}^{m\times n}$ to the vector space $\mathbb{R}^{k_s}$ with the seminorm $\omega_s\psi_s$. Here, the number k_s is the cardinality of $\mathcal{G}_s$. The intersection of the kernels of all these linear operators—the one corresponding to the data term and the S corresponding to the regularizer—is a linear space which we call K. On K, and on the parallel affine subspaces $a + K$ for all a as well, the functional is constant. This may be seen by considering $u, v \in a + K$ and concluding from $u - v \in K$ that the value of u, v for each summand of the functional $F_\mathcal{G}$ in (6) is equal. Hence, we may restrict to a complement of the kernel and project the constraining set on this complement. On the complement, the functional is coercive and hence has a minimizer. In consequence, (5) has a minimizer whenever the admissible set is nonempty. We denote this set of minimizers by $M_\mathcal{G}$ (which is empty only if there is no admissible point).

Now we allow the sets $\mathcal{G}$ to vary. For those $\mathcal{G}$ where the corresponding problem (5) has an admissible point, we consider the quantities

$$m_\mathcal{G} = \min_u F_\mathcal{G}(u) + \gamma\sum_{s=1}^{S}\omega_s\ |\mathcal{I}_s\backslash\mathcal{G}_s|, \tag{7}$$

where $F_\mathcal{G}$ is defined in (6), and the minimum is taken over all admissible u. Here, $\mathcal{I}_s$ are those tuples in $\mathcal{I}$ having direction a_s and $|\mathcal{I}_s\backslash\mathcal{G}_s|$ denotes the cardinality of the corresponding complement of $\mathcal{G}_s$ in $\mathcal{I}_s$. We choose $\mathcal{G}'$ such that $m_{\mathcal{G}'}$ is minimal among all $\mathcal{G}$ with an admissible point, i.e.,

$$\mathcal{G}' \in \arg\min_\mathcal{G}\ m_\mathcal{G}. \tag{8}$$

This is possible since $\mathcal{I}$ is a finite set and therefore the cardinality of the set of its subsets is finite. We claim that a minimizer

$$u^* \in M_{\mathcal{G}'} \text{ is a minimizer of the Mumford–Shah functional (2)} \tag{9}$$

which would complete the proof. (The set $M_\mathcal{G}$ was defined at the end of the previous paragraph.)

In order to prove this claim we need to show the following key property of a minimizer $u^* \in M_{\mathcal{G}'}$: for u^* we have

$$|u^*_{v+a_s} - u^*_v| \geqslant \gamma' \quad \text{for all } (v, a_s) \in \mathcal{I}\backslash\mathcal{G}',\ s \in \{1, \ldots, S\}. \tag{10}$$

Suppose to the contrary that this is not the case. Then, we define $\mathcal{G}'' \supset \mathcal{G}'$ distinguishing whether $|u^*_{v+a_s} - u^*_v| \leqslant \gamma'$ or not: precisely, we let $(v, a_s) \in \mathcal{G}''_s$, if and only if $|u^*_{v+a_s} - u^*_v| \leqslant \gamma'$, and define $\mathcal{G}''$ by $\mathcal{G}'' = \bigcup_{s=1}^S\ \mathcal{G}''_s$. Then, we use the notation u'' to denote a corresponding minimizer of (5) w.r.t. $\mathcal{G}''$. We have that u^* is admissible for 5 w.r.t. $\mathcal{G}''$, and further, by the minimality of u'',

$$\begin{aligned}
&F_{\mathcal{G}''}(u'') + \gamma\sum_{s=1}^{S}\omega_s\ |\mathcal{I}_s\backslash\mathcal{G}''_s| \leqslant F_{\mathcal{G}''}\left(u^*\right) + \gamma\sum_{s=1}^{S}\omega_s\ |\mathcal{I}_s\backslash\mathcal{G}''_s| \\
&< F_{\mathcal{G}''}\left(u^*\right) + \sum_{s=1}^{S}\omega_s \sum_{\left\{v=(i,j):(v,a_s)\in\mathcal{G}''_s\backslash\mathcal{G}'_s\right\}}\left(\gamma - \alpha\left|u^*_{ij} - u^*_{(i,j)+a_s}\right|^q\right) + \gamma\sum_{s=1}^{S}\omega_s\ |\mathcal{I}_s\backslash\mathcal{G}''_s| \\
&= F_{\mathcal{G}'}\left(u^*\right) + \gamma\sum_{s=1}^{S}\omega_s\ |\mathcal{I}_s\backslash\mathcal{G}'_s|.
\end{aligned} \tag{11}$$

Here, the strict inequality is due to our assumption on u^* and the definition of $\mathcal{G}''$. More precisely, by our assumption, there is a tuple (v, a_s) in $\mathcal{I}\backslash\mathcal{G}'$ such that $|u^*_{v+a_s} - u^*_v| < \gamma'$; this tuple is in $\mathcal{G}''$ by definition, and so the second summand in the second line of (11) is strictly positive. We conclude from (11) that $m_{\mathcal{G}''} < m_{\mathcal{G}'}$ which contradicts our assumption that $m_{\mathcal{G}'}$ is minimal. Hence, (10) is true.

We finally show (9) which states that $u^* \in M_{\mathcal{G}'}$ with $\mathcal{G}'$ as in (8) is a minimizer of the Mumford–Shah functional (2). To this end, we consider an arbitrary u and show that its Mumford–Shah value is larger than or equal to $m_{\mathcal{G}'} = \min_{\mathcal{G}} m_{\mathcal{G}}$. For u, we define $\mathcal{G}''$ in analogy to the way we defined $\mathcal{G}''$ for u^* above. Precisely, we let $\mathcal{G}''_s$ be those tuples (v, a_s) where the value $|u_{v+a_s} - u_v|$ of u does not exceed γ', and let $\mathcal{G}'' = \bigcup_{s=1}^S \mathcal{G}''_s$. Then u is admissible for the problem (5) with respect to $\mathcal{G}''$. Hence $M_{\mathcal{G}''}$ is nonempty and there is a minimizer of (5) w.r.t. $\mathcal{G}''$. By the definition of $\mathcal{G}'$,

$$m_{\mathcal{G}''} \geqslant m_{\mathcal{G}'}. \tag{12}$$

In addition, by the definition of $\mathcal{G}''$, the Mumford–Shah value of u equals the value $F_{\mathcal{G}''}(u) + \gamma \sum_{s=1}^S \omega_s\, |\mathcal{I}_s \backslash \mathcal{G}''_s|$ which, by the admissibility of u for (5) with respect to $\mathcal{G}''$, is bigger than or equal to $m_{\mathcal{G}''}$. The value $m_{\mathcal{G}'}$ equals the Mumford–Shah value of u^* (see (10)). Together with (12), these two observations yield that the Mumford–Shah value of an arbitrary u is bigger than or equal to the Mumford–Shah value of u^*. This shows (9) and completes the proof. □

We remark that, for the existence of minimizers, we actually need the simpler problems (5) to have a minimizer which might be easier to show in particular for nonlinear situations not covered in the theorem. We further notice that minimizers need not be unique in general. In the above proof, this is, for instance, reflected by the non-uniqueness of the sets $\mathcal{G}'$ of (8) and the fact that the set K at the beginning of the proof formed w.r.t. a minimizing set $\mathcal{G}'$ of (8) need not be single-valued.

2.2. ADMM splitting for the Mumford–Shah problem

Our minimization strategy for the (discretized) Mumford–Shah problem (2) is based on an alternating direction method of multipliers (ADMM) approach developed in the authors' work [43, 45] in the context of Potts problems.

Our first step is to rewrite (2) as the constrained optimization problem

$$\begin{aligned} &\text{minimize } \sum_{s=1}^{S} \omega_s \phi\left(\nabla_{a_s} u_s\right) + d\left(H(v), f\right), \\ &\text{subject to } v - u_s = 0, \;\; \text{for all } 1 \leqslant s \leqslant S, \\ &\qquad\qquad u_r - u_t = 0, \;\; \text{for all } 1 \leqslant r < t \leqslant S. \end{aligned} \tag{13}$$

The augmented Lagrangian $\mathcal{L}$ of this minimization problem is given by

$$\begin{aligned} &\mathcal{L}\Big(\{u_s\}_{s=1}^S, v, \{\lambda_s\}_{s=1}^S, \{\rho_{r,t}\}_{1 \leqslant r < t \leqslant S} \Big) \\ &= \sum_{s=1}^{S} \omega_s \phi(\nabla_{a_s} u_s) + \frac{\mu}{2} \parallel v - u_s + \frac{\lambda_s}{\mu} \parallel_2^2 - \frac{1}{2\mu} \parallel \lambda_s \parallel_2^2 \\ &\quad + \sum_{1 \leqslant r < t \leqslant S}^{S} \frac{\nu}{2} \| u_r - u_t + \frac{\rho_{r,t}}{\nu} \|_2^2 - \frac{1}{2\nu} \parallel \rho_{r,t} \parallel_2^2 + d(H(v), f). \end{aligned} \tag{14}$$

In the above form, the constraints are included in the target functional, and the parameters $\mu > 0$ and $\nu > 0$ control their weights. The λ_s, $\varrho_{s,t} \in \mathbb{R}^{m \times n}$, where $1 \leqslant s \leqslant S$ and $1 \leqslant r < t \leqslant S$, take the role of Lagrange multipliers.

Minimization of (14) with respect to all target variables simultaneously is still unfeasible. Nevertheless, our specific splitting allows us to use ADMM. That means, that we minimize $\mathcal{L}$ with respect to $u_1, \ldots, u_S$, and v. Then, we perform a gradient ascent step in the Lagrange multiplier. (See [8] for an extensive treatment on the ADMM.) To bring the expressions for $\arg\min_{u_s}\mathcal{L}$ into a convenient form we use the fact that $\sum_i x_i(a - b_i)^2 = (\sum_i x_i)(a - \frac{\sum_i b_i x_i}{\sum_i x_i})^2 + C$ for $a, b_1, \ldots, b_N \in \mathbb{R}$ and $x_1, \ldots, x_N > 0$ and a constant $C \in \mathbb{R}$ that does not depend on a. Then, we obtain by a standard algebraic manipulation that

$$\arg\min_{u_s} \mathcal{L} = \arg\min_{u_s} \frac{2\omega_s}{\mu + \nu(S-1)} \phi\left(\nabla_{a_s} u_s\right) + \| u_s - w_s \|_2^2$$

where

$$w_s = \frac{\mu v + \lambda_s + \sum_{1 \leqslant r < s} (\nu u_r + \varrho_{r,s}) + \sum_{s < t \leqslant S} (\nu u_t - \varrho_{s,t})}{\mu + \nu(S-1)}.$$

Furthermore, we rewrite the minimization problem with respect to v in terms of the the proximal mapping which is defined for a functional F by

$$\mathrm{prox}_F(z) = \arg\min_v F(v) + \frac{1}{2}\|v - z\|_2^2, \tag{15}$$

see [39], for instance. In this notation, a minimizer of $\mathcal{L}$ with respect to v reads

$$\begin{aligned} \arg\min_v \mathcal{L} &= \arg\min_v d(H(v), f) + \frac{\mu S}{2}\left\| v - \frac{1}{S}\sum_{s=1}^{S}\left(u_s - \frac{\lambda_s}{\mu}\right)\right\|_2^2 \\ &= \mathrm{prox}_{d(H(\cdot), f)/(\mu S)}\left(\frac{1}{S}\sum_{s=1}^{S}\left(u_s - \frac{\lambda_s}{\mu}\right)\right). \end{aligned}$$

Thus, we obtain the ADMM iteration

$$\begin{cases} u_1^{k+1} &= \arg\min_{u_1} \frac{2\omega_1}{\mu_k + \nu_k(S-1)}\phi\left(\nabla_{a_1}u_1\right) + \|u_1 - w_1^k\|_2^2, \\ &\vdots \\ u_S^{k+1} &= \arg\min_{u_S} \frac{2\omega_S}{\mu_k + \nu_k(S-1)}\phi\left(\nabla_{a_S}u_S\right) + \|u_S - w_S^k\|_2^2, \\ v^{k+1} &= \mathrm{prox}_{d(H(\cdot),f)/(\mu_k S)}\left(\frac{1}{S}\sum_{s=1}^{S}\left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k}\right)\right), \\ \lambda_s^{k+1} &= \lambda_s^k + \mu_k\left(v^{k+1} - u_s^{k+1}\right), \quad \text{for all} \quad 1 \leqslant s \leqslant S, \\ \varrho_{r,t}^{k+1} &= \varrho_{r,t}^k + \nu_k\left(u_r^{k+1} - u_t^{k+1}\right), \quad \text{for all} \quad 1 \leqslant r < t \leqslant S, \end{cases} \tag{16}$$

where w_s^k is given by

$$w_s^k = \frac{\mu_k v^k + \lambda_s^k + \sum_{1 \leqslant r < s}\left(\nu_k u_r^{k+1} + \varrho_{r,s}^k\right) + \sum_{s < t \leqslant S}\left(\nu_k u_t^k - \varrho_{s,t}^k\right)}{\mu_k + \nu_k(S-1)}.$$

We use as coupling parameters increasing sequences $(\mu_k)_{k \in \mathbb{N}}$ and $(\nu_k)_{k \in \mathbb{N}}$.

The important point is that the subproblems in (16) are computationally tractable. The minimization problems in $u_1, \ldots, u_S$ decompose into univariate Mumford–Shah problems with respect to the paths induced by the finite difference vectors a_s. Moreover, these univariate subproblems can be solved in parallel. The last subproblem is the evaluation of a proximal mapping with respect to the data term. In the context of inverse problems, the proximal mapping is often called a classical Tikhonov regularization. Hence, the algorithm is applicable whenever classical Tikhonov regularization of the data term $d(H(v), f)$ is possible. This is the case in many imaging setups; see section 3 for some examples. Solvers for the subproblems will be described in the following.

2.3. Efficient solvers for the univariate Mumford–Shah problems

The crucial part of the splitting approach (16) is the efficient solution of univariate Mumford–Shah/Blake–Zisserman problems of the form

$$\min_{u\in\mathbb{R}^n}\left\{ B_{\gamma,\alpha}(u) = \sum_i \min\left(\gamma,\ \alpha\left|u_{i+1} - u_i\right|^q\right) + \sum_i (u_i - y_i)^2 \right\}, \tag{17}$$

where $y \in \mathbb{R}^n$. In order to see the connection to (1), we reformulate (17) equivalently as

$$\min_{u\in\mathbb{R}^n, J\subset\{1,\ldots,n-1\}} \gamma\,|J| + \alpha\sum_{i\notin J}|u_{i+1} - u_i|^q \quad + \sum_i (u_i - y_i)^2 \tag{18}$$

where $|J|$ denotes the number of elements in the 'jump set' J. Recall from the introduction that the solver of [6] for (18) with $q = 2$ has cubic time and linear memory complexity in the worst case. For the case $q = 1$, no efficient solver has been proposed yet to our knowledge. To obtain a computationally feasible method, we next propose efficient solvers.

2.3.1. Basic dynamic programming approach to univariate Mumford–Shah problems. Our solvers are based on the dynamic programming scheme developed in [21, 50] which we explain next. Assume that we have already computed minimizers u^l of the functional $B_{\gamma,\alpha}$ associated with the partial data $(y_1, \ldots, y_l)$ for each $l = 1, \ldots, r-1$ and some $r \leqslant n$. Then, the functional $B_{\gamma,\alpha}$ associated with data $(y_1, \ldots, y_r)$ can be computed efficiently by

$$\min_{u\in\mathbb{R}^r} B_{\alpha,\gamma}(u) = \min_{l=1,\ldots,r}\left\{ B_{\alpha,\gamma}\left(u^{l-1}\right) + \gamma + \varepsilon_{l,r}\right\}, \tag{19}$$

where $\varepsilon_{l,r}$ denotes an approximation error on the interval $(l, \ldots, r)$ given by

$$\varepsilon_{l,r} = \min_{h\in\mathbb{R}^{r-l+1}} \alpha\,\|\nabla h\|_q^q + \|h - (y_l, \ldots, y_r)\|_2^2. \tag{20}$$

Here we use the convention that u^0 is the empty vector and $B_{\alpha,\gamma}(u^0) = -\gamma$. A corresponding minimizer reads $u^r = (u^{l^*-1}, h^{l^*,r})$ where l^* is a minimizing argument of the right-hand side of (19) and $h^{l^*,r}$ is a minimizer of (20). It has been shown in [21] that we obtain a minimizer u^* for full data y by successively computing u^r for each $r = 1, \ldots, n$. When using suitable data structures, the described procedure is $\mathcal{O}(n^2)$ time and $\mathcal{O}(n)$ space provided that the approximation error $\varepsilon_{l,r}$ can be computed in $\mathcal{O}(1)$ [21].

It remains to compute the $\varepsilon_{l,r}$ for all $1 \leqslant l \leqslant r \leqslant n$. Since the functional in (20) is convex for all $q \geqslant 1$ we could use generic solvers for convex optimization problems, see e.g. [9, 35]. However, generic solvers are too slow in practice, so our next goals are faster algorithms.

2.3.2. Fast computation of the approximation errors for $q = 2$. In the case of quadratic variation, we get by a standard computation that a minimizer $h^{l,r}$ of (20) satisfies the linear system of equations

$$\left(\alpha U^{(r-l+1)} + \mathrm{id}\right) h^{l,r} = (y_l, \ldots, y_r)^T,$$

where $U^{(t)} \in \mathbb{R}^{t\times t}$ is given by

$$U^{(t)} = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}$$

Since $(\alpha U^{(t)} + \mathrm{id})$ is a bandmatrix, we can use Gaussian elimination to compute a minimizer $h^{l,r}$ and then compute the approximation error using (20). This procedure is easy to derive and to implement but it amounts in total to $\mathcal{O}(n^3)$ for computing all $\varepsilon_{l,r}$. In [21], it is claimed that the $\varepsilon_{l,r}$ can be computed in $\mathcal{O}(n^2)$ time and $\mathcal{O}(n^2)$ memory using the eigenvalue decompositions of the matrices $U^{(t)}$. The authors of the present paper could not verify the claimed $\mathcal{O}(n^2)$ time complexity from the information given in [21]. We believe that, there, the costs for computing the moments have been neglected and that the given procedure grows at least cubicly, i.e., it is not an $\mathcal{O}(n^2)$ algorithm.

Here, we propose the utilization of a completely different way to compute the $\varepsilon_{l,r}$. We use a sophisticated recurrence relation which has been derived by Blake [6]. For completeness, we here provide this recurrence relation. First, the approximation error for a single point (f_l) is given by $\varepsilon_{l,l} = 0$. Then, we recursively define an auxiliary n-tuple C by

$$C_1 = 1, \quad \text{and} \quad C_t = \frac{\alpha C_{t-1}}{\left(C_{t-1} + \alpha\right)} + 1, \quad \text{for } t \geqslant 2.$$

We consider another auxiliary variable $\bar{u}_{l,\cdot}$ whose first entry is given by $\bar{u}_{l,l} = y_l$. For $r \geqslant l + 1$, the rth entry is given by the recurrence

$$\bar{u}_{l,r} = \frac{\bar{u}_{l,r-1}\left(C_{r-l+1} - 1\right) + y_r}{C_{r-l+1}},$$

and the corresponding error is given by

$$\varepsilon_{l,r} = \varepsilon_{l,r-1} + \left(C_{r-l+1} - 1\right)(\bar{u}_{l,r} - \bar{u}_{l,r-1})^2 + (\bar{u}_{l,r} - y_r)^2.$$

The updates only need constant time. Thus, we are able to compute the $\varepsilon_{l,r}$ for all $1 \leqslant l \leqslant r \leqslant n$ in $\mathcal{O}(n^2)$ time. Moreover, we observe that the computation of a specific $\varepsilon_{l,r}$ depends only on C, $\bar{u}_{l,r-1}$, $\bar{u}_{l,r}$, and $\varepsilon_{l,r-1}$. Hence, the $\varepsilon_{l,r}$ can be computed *in situ* in the dynamic program of section 2.3.1 so that we obtain $\mathcal{O}(n)$ memory complexity. A pseudocode of the proposed solver is given in algorithm 1. In summary, we have obtained:

Theorem 2.2. *Algorithm 1 computes a global minimizer for the univariate Mumford–Shah/ Blake–Zisserman problem (17) with $q = 2$. Its complexity is $\mathcal{O}(n^2)$ w.r.t. time and $\mathcal{O}(n)$ w.r.t. memory.*

To our knowledge, the proposed procedure (algorithm 1) is the first exact solver in this complexity class. We point out that the applicability of the algorithm is not limited to the considered imaging setup. Another important application is the efficient smoothing of time series with discontinuities [7, 21, 49].

Algorithm 1. Fast solver for the 1D Mumford–Shah problem ($q \in \{1, 2\}$)

```
Input: Data $y \in \mathbb{R}^n$; model parameters $\alpha, \gamma > 0$;
Output: Global minimizer of the 1D Mumford–Shah problem $u \in \mathbb{R}^n$
begin
    /* Find the optimal jump locations */
    for $r \leftarrow 1, \dots, n$ do
        $J_r \leftarrow 0$                                  /* init right-most jump */
        $B_r \leftarrow \varepsilon_{1,r}$                  /* $p = 2$: section 2.3.2; $p = 1$: section 2.3.3 */
        for $l \leftarrow r, r - 1, \dots, 2$ do
            Compute $\varepsilon_{l,r}$                     /* $p = 2$: section 2.3.2; $p = 1$: section 2.3.3 */
            $b \leftarrow B_{l-1} + \gamma + \varepsilon_{l,r}$   /* compute candidate functional value $b$ */
            if $b \leqslant B_r$ then
                $B_r \leftarrow b$                          /* store new best functional value */
                $J_r \leftarrow l - 1$                      /* update right-most jump location */
            end
            if $\varepsilon_{l,r} + \gamma > B_r$ then break end   /* Pruning, see section 2.3.4 */
        end
    end
    /* Reconstruct the minimizer from the optimal jump locations */
    $r \leftarrow n$; $l \leftarrow J_r$
    while $r > 0$ do
        $(u_{l+1}, \dots, u_r) \leftarrow$ Minimizer of (20) with data $(y_{l+1}, \dots, y_r)$
        $r \leftarrow l$; $l \leftarrow J_r$                /* go to next jump */
    end
    return u
end
```

2.3.3. Fast computation of the approximation errors for q = 1. Our strategy for the case $q = 1$ is based on two key ingredients. First, we use the taut string algorithm to solve (20) in linear time. The taut string method was introduced in [30]; some further references are [17, 23, 42]. The second key ingredient is to reuse a great portion of the computations performed for $\varepsilon_{l+1,r}$ for the efficient computation of $\varepsilon_{l,r}$. Next we describe the concrete procedure.

Let us first recall the taut string algorithm following the presentations of [17, 42]. The basic idea of the taut string method for the solution of (20) is as follows. We compute the cumulative sums Y of the data $(y_l, \dots, y_r)$ and form a tube of radius $\alpha/2$ (in ordinate direction) around that vector.

Then we introduce a string into this tube, and pull from both ends until the string is taut. The slopes of the string form the minimizer $h^{l,r}$ of (20). More precisely, the algorithm is as follows:

(i) Compute the cumulative sum $Y = (0, y_l, y_l + y_{l+1}, \dots, \sum_{i=l}^{r} y_i)^T$.

(ii) Compute the upper and lower tube bounds $Y_u = \{Y_0, Y_1 + \alpha/2, \dots, Y_{n-1} + \alpha/2, Y_n\}$ and $Y_l = \{Y_0, Y_1 - \alpha/2, \dots, Y_{n-1} - \alpha/2, Y_n\}$, where α is the smoothing parameter of the Mumford–Shah functional.

(iii) Compute iteratively the convex majorant Y_u' of the upper bound and the concave minorant Y_l' of the lower bound both starting with the point $(0, Y(0))$

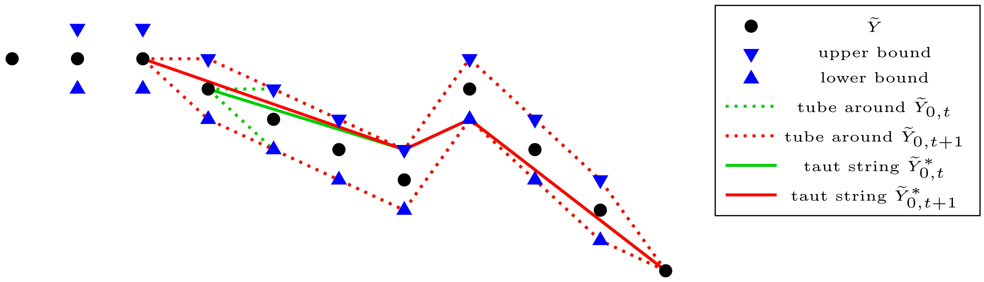

$\widetilde{Y}$
upper bound
lower bound
tube around $\widetilde{Y}_{0,t}$
tube around $\widetilde{Y}_{0,t+1}$
taut string $\widetilde{Y}^*_{0,t}$
taut string $\widetilde{Y}^*_{0,t+1}$

Figure 3. Illustration of the efficient solver for the univariate Mumford–Shah problem with q=1. We observe that the tubes associated with data $(y_l, \dots, y_r)$ and $(y_{l+1}, \dots, y_r)$ are identical up to the two last points. Reusing the identical parts saves a significant amount of computation.

(iv) Repeat step 3 until the right side derivative at 0 of the convex majorant is smaller than the one of the concave minorant.

(v) Take the point where the majorant touches the upper bound first or where the minorant touches the lower bound first as new starting point of the majorant and minorant wherever the ordinate value is smaller. Repeat this until the right side derivative of the common starting point of the majorant is again larger than of the minorant.

(vi) Do all of this until there are no new points to be processed.

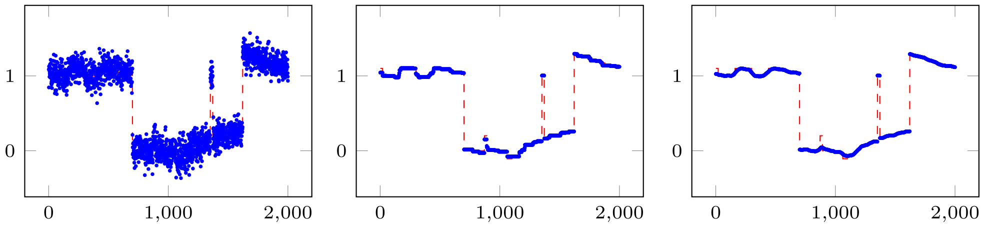
We end up with the taut string coded in the vector Y^*. The final solution $h^{l,r}$ of (20) is obtained by 'differentiating' Y^*, i.e. $h^{l,r} = \nabla Y^*$, where ∇ denotes the finite difference operator. Since the taut string has linear complexity, we obtain in total:

Theorem 2.3. *Algorithm 1 computes a global minimizer for the univariate Mumford–Shah/Blake–Zisserman problem (17) with $q = 1$. It is of $\mathcal{O}(n^3)$ time complexity and of $\mathcal{O}(n)$ memory complexity.*

Next we describe how to significantly decrease the computational costs of the solver. First, we will compute the taut string in reverse order; that is, we form a tube around the reverse cumulative sum $\widetilde{Y}$ given by

$$\widetilde{Y} = \left(0,\, y_r,\, y_r + y_{r-1},\, \dots,\, \sum_{i=1}^{r} y_{r-i+1}\right).$$

The reversed order is important for utilizing a pruning strategy which we will describe in section 2.3.4. Now let us consider the computation of $\varepsilon_{l+1,r}$ and $\varepsilon_{l,r}$ for some $l < r$. Thus, the task is to compute a solution of (20) for data $(y_{l+1},\dots,y_r)$ and $(y_l,\dots,y_r)$. Following the taut string algorithm, we form a tube around the vectors $(\widetilde{Y}_0,\dots,\widetilde{Y}_t)$ and $(\widetilde{Y}_0,\dots,\widetilde{Y}_{t+1})$ where $t = r - l$. The important observation is that the tubes around these two vectors are identical up to the last two points t and $t + 1$; see figure 3. This means that the computations of the taut string algorithm are identical until the point $\widetilde{Y}_{t-1}$ is reached. Therefore, we store the current state of the algorithm when reaching $\widetilde{Y}_{t-1}$. Then, we continue with the computation of the string in the tube around $(\widetilde{Y}_0,\dots,\widetilde{Y}_t)$. In the next iteration, we restore the previous state for the computation of the string in the tube around $(\widetilde{Y}_0,\dots,\widetilde{Y}_{t+1})$. In the worst case scenario, it still can happen that we have to run through all t data points; therefore, we are still in the linear complexity class. Nevertheless, for the signals used in this work, we observed a speedup by a factor of three to four.



(a) Noisy data, $\sigma = 0.1$ **(b)** Ordinary variation ($q = 1$) **(c)** Quadratic variation ($q = 2$)

Figure 4. Comparison of the effects of ordinary and quadratic variation on the solution of the univariate Mumford–Shah problem. The first variation preserves the jumps of low contrast, whereas the second variation produces a smoother solution. (Original signal: red dashed line.)

Eventually, in figure 4, we illustrate the effects of Mumford–Shah regularization with different variation penalties. We observe that the L^1 variation penalty preserves the jumps of low contrast better, whereas the quadratic variant produces smoother results in the low contrast regime.

2.3.4. Pruning strategy. The above solvers can be accelerated significantly in practice by pruning the search space of the basic dynamic program presented in section 2.3.1. Here, we adopt a strategy that has been proposed in similar forms in [25] and in [43] in the context of Potts problems. More precisely, we reduce the time for finding the minimizing argument l^* in (19). Let us denote $b_t = \min_{l=t,\ldots,r} B_{\alpha,\gamma}(u^{l-1}) + \gamma + \varepsilon_{l,r}$. First we observe that the approximation errors $\varepsilon_{l,r}$ and the functional values $B_{\alpha,\gamma}(u^{l-1})$ are non-negative. Assume that $b_t \leqslant \gamma + \varepsilon_{t,r}$ for some $t \leqslant r$. Since the mapping $l \mapsto \varepsilon_{l,r}$ is non-increasing it follows that $b_s \geqslant b_t$ for all $s \leqslant t$. This means that the minimizing index lies between t and r. Thus, the indices smaller than t can be skipped. In our experiments this pruning led to an overall speedup of a four to fivefold factor.

Note that the pruning strategy requires that we compute the approximation errors $\varepsilon_{l,r}$ in the order $l = r, r - 1, \ldots, 1$. This explains why we use the reverse order in our taut string based solver (cf. section 2.3.3).

2.4. Computation of the proximal point

Our method is applicable if the proximal mapping of the data term $d(H(v), f)$ can be evaluated within a reasonable time. This term $d(H(v), f)$ is determined by the properties of the image acquisition device, and so its proximal mapping depends on the imaging setup. In particular, when H is nonlinear and when d is non-convex this is typically a challenging problem in its own. However, as it corresponds to Tikhonov regularization, implementations of the proximal mapping are often available. Here, we briefly describe how to proceed in the frequently occurring situation in which H is a linear operator, i.e., $H(u) = Hu$, and the noise follows a zero mean distribution of the exponential family. Then, the natural data fidelity term is given by

$$d(H(v), f) = \frac{1}{p} \|Hv - f\|_p^p, \tag{21}$$

where H is a linear operator and $p \geqslant 1$. The canonical choice for a Gaussian noise model is $p = 2$. For Laplace noise, $p = 1$ is the appropriate choice. For the corresponding computations, the structure of the linear operator can be exploited in many practically relevant cases. For example, we can take advantage of the Fourier transform for convolution operators (see section 3). When there is no such structure available, we have to resort to iterative algorithms. Since the data term is convex for all $p \geqslant 1$, the proximal mapping $\mathrm{prox}_{d(H(\cdot),f)/\tau}$ can be evaluated using generic convex optimization techniques [9], for example the Chambolle–Pock algorithm [14]. For the practically most relevant cases $p = 1, 2$, we can also use dedicated minimization methods. For $p = 2$, the proximal point $v^* = \mathrm{prox}_{d(H(\cdot),f)/\tau}(z)$ satisfies the normal equation

$$\left(H^*H + \tau I\right) v^* = H^*f + \tau z, \tag{22}$$

where H^* denotes the adjoint of H. Thus, the computation of v^* reduces to solving a linear system of equations. For $p = 1$, one can use the semi-smooth Newton method described in [15].

To retain the efficiency in the general case, we utilize 'warmstarting'; that means, we use the solution of iteration v^k as initial guess for v^{k+1}. This makes the iterative solver typically converge within only a few iterations, thus only a few evaluations of H and H^* have to be performed.

2.5. Convergence

We show that algorithm (16) converges in the prototypical case $\nu_k = 0$ for all k (which implies that the $\varrho^k = 0$ for all k.) We leave a convergence proof for algorithm (16) with general ν_k as an open problem. The following proof is closely related to the proofs in [11, 44, 45] which show the convergence of the algorithms for the Potts problem developed in these papers. We here provide the necessary modifications to deal with the Mumford–Shah setting and possibly nonlinear operators H. Instead of pointing out where these modifications should be done, we, as a service to the reader, provide a self-contained version of the proof for the considered setup.

Theorem 2.4. *Let H be a Lipschitz continuous (not necessarily linear) operator from $\mathbb{R}^{m\times n}$ to the metric space $(X; d')$ and the data term be given by $d(H(u), f) = d'(H(u), f)^p$ for some $p \geqslant 1$. Let the sequence $(\mu_k)_{k\in\mathbb{N}}$ be increasing and satisfy $\sum_k \mu_k^{-1/2} < \infty$. Furthermore, let $\nu_k = 0$ for all k. Then, the iteration (16) converges in the sense that*

$$\left(u_1^k, \ldots, u_S^k, v^k\right) \to \left(u_1^*, \ldots, u_S^*, v^*\right) \quad \text{with} \quad u_1^* = \ldots = u_S^* = v^*,$$
$$\frac{\lambda_s^k}{\mu_k} \to 0 \quad \text{for all} \quad s \in \{1, \ldots, S\}. \tag{23}$$

Proof. We denote the S functionals appearing in the first S lines of (16) by F_s^k, i.e.,

$$F_s^k(u_s) = \frac{2\omega_s}{\mu_k}\phi\left(\nabla_{a_s}u_s\right) + \|u_s - \left(v^k + \frac{\lambda_s^k}{\mu_k}\right)\|_2^2.$$

Using this notation, we rewrite the first S lines of (16) as $u_s^{k+1} \in \arg\min_u F_s^k(u_s)$ for all $s \in \{1, \ldots, S\}$. We first estimate the distance $\|u_s^{k+1} - (v^k + \frac{\lambda_s^k}{\mu_k})\|_2$. To that end, we note that $F_s^k(u_s^{k+1}) \leqslant F_s^k(v^k + \frac{\lambda_s^k}{\mu_k})$ which holds true since u_s^{k+1} minimizes F_s^k. Applying the definition of F_s^k, we get

$$\omega_s\phi\left(\nabla_{a_s}u_s^{k+1}\right) + \frac{\mu_k}{2}\|u_s^{k+1} - \left(v^k + \frac{\lambda_s^k}{\mu_k}\right)\|_2^2 \leqslant \omega_s\phi\left(\nabla_{a_s}\left(v^k + \frac{\lambda_s^k}{\mu_k}\right)\right) \leqslant \gamma\omega_s L,$$

where $L = mn$ is the size of the considered $(m \times n)$ image. This is because $\phi(\nabla_{a_s}z) \leqslant mn\gamma$ for any a_s and any data z. Since the first summand on the left-hand side is nonnegative, we get that

$$\| u_s^{k+1} - \left(v^k + \frac{\lambda_s^k}{\mu_k} \right) \|_2^2 \leqslant \frac{\gamma \omega_s L}{\mu_k}. \tag{24}$$

In particular, for all $s \in \{1, \ldots, S\}$, we obtain that

$$\lim_{k \to \infty} u_s^{k+1} - \left(v^k + \frac{\lambda_s^k}{\mu_k} \right) = 0. \tag{25}$$

We now look at the $(S + 1)$th line of (16). We denote the corresponding functional by G^k, i.e.,

$$G^k(v) = d'(H(v), f)^p + \frac{\mu_k S}{2} \| v - \frac{1}{S} \sum_{s=1}^{S} \left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} \right) \|_2^2.$$

The minimality of v^{k+1} implies the inequality

$$G^k\left(v^{k+1}\right) \leqslant G^k\left(\frac{1}{S} \sum_{s=1}^{S} \left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} \right) \right).$$

We now apply the definition of G^k to estimate

$$\begin{aligned} & d'\left(H\left(v^{k+1}\right), f\right)^p + \frac{\mu_k S}{2} \| v^{k+1} - \left(\frac{1}{S} \sum_{s=1}^{S} \left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} \right) \right) \|_2^2 \\ & \leqslant d'\left(H\left(\frac{1}{S} \sum_{s=1}^{S} \left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} \right) \right), f \right)^p \\ & \leqslant \left(d'\left(H\left(\frac{1}{S} \sum_{s=1}^{S} \left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} \right) \right), H\left(v^k\right) \right) + d'\left(H\left(v^k\right), f \right) \right)^p \\ & \leqslant \left(K \| \frac{1}{S} \sum_{s=1}^{S} \left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} - v^k \right) \|_2 + d'\left(H\left(v^k\right), f \right) \right)^p. \end{aligned} \tag{26}$$

Here, K is the Lipschitz constant of the operator H acting on ℓ^2. Note that such a constant exists since, on a finite dimensional space, all norms are equivalent and the standard topology in Euclidean space is a norm topology. Next we combine (26) and (24) to estimate the magnitude of the residuals $d'(H(v^{k+1}), f)$. We get

$$d'\left(H\left(v^{k+1}\right), f\right) \leqslant \frac{C}{\sqrt{\mu_k}} + d'\left(H\left(v^k\right), f\right),$$

where $C > 0$ is a constant that only depends on γ, ω_s L, and K. Solving this recursion yields

$$d'\left(H\left(v^{k+1}\right), f\right) \leqslant C \sum_{j=1}^{k} \frac{1}{\sqrt{\mu_j}} + d'\left(H\left(v^0\right), f\right),$$

which shows that the sequence of residuals $(d'(H(v^{k+1}), f))_{k \in \mathbb{N}}$ is bounded.

We consider the right-hand term in the first line of (26) and we apply (26) to get

$$\begin{aligned}
&\tfrac{\mu_k S}{2} \| v^{k+1} - \frac{1}{S}\sum_{s=1}^{S}\left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k}\right)\|_2^2 \\
&\leqslant \left(d'\left(H\left(\frac{1}{S}\sum_{s=1}^{S}\left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k}\right)\right), H\left(v^k\right)\right) + d'\left(H\left(v^k\right), f\right)\right)^p \\
&\leqslant \left(K \ \|\frac{1}{S}\sum_{s=1}^{S}\left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} - v^k\right)\|_2 + C'\right)^p.
\end{aligned}$$

The last inequality is a consequence of the boundedness of the residuals where we denote the bound by the positive constant C' (which is independent of k). We now apply (25) to the first summand to conclude that the sequence (with respect to k)

$$\mu_k \left\| v^{k+1} - \frac{1}{S}\sum_{s=1}^{S}\left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k}\right)\right\|_2^2 \text{ is bounded.} \tag{27}$$

We use this fact to establish the convergence of the sequence v^k by showing that it is a Cauchy sequence. We first apply the triangle inequality to get

$$\| v^{k+1} - v^k \|_2 \leqslant \| v^{k+1} - \frac{1}{S}\sum_{s=1}^{S}\left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k}\right)\|_2 + \|\frac{1}{S}\sum_{s=1}^{S}\left(u_s^{k+1} - \frac{\lambda_s^k}{\mu_k} - v^k\right)\|_2.$$

We now apply (27) to the first summand on the right-hand side as well as (24) to the second summand on the right-hand side to obtain $\| v^{k+1} - v^k \|_2 \leqslant \frac{C''}{\sqrt{\mu_k}}$ for some constant $C'' > 0$ which is again independent of k. The assumption on the sequence μ_k guarantees that v^k is a Cauchy sequence and hence that v^k converges to some v^*.

To establish the last statement in (23), we rewrite each of the last S lines in (16) to obtain the identity

$$\frac{\lambda_s^{k+1}}{\mu_k} = \left(\frac{\lambda_s^k}{\mu_k} - u_s^{k+1} + v^k\right) + \left(v^{k+1} - v^k\right). \tag{28}$$

by (25) and (27), each term in parentheses converges to 0. Hence, $\lim_{k\to\infty}\frac{\lambda_s^{k+1}}{\mu_k} = 0$. Since we assume that the sequence μ_k is nondecreasing, we have that $\mu_k/\mu_{k+1} \leqslant 1$ and thus, for all $s = 1, \ldots, S$, $\lim_{k\to\infty}\frac{\lambda_s^k}{\mu_k} = 0$. This shows the last statement in (23). We rewrite the penultimate line of (16) as $v^{k+1} - u_s^{k+1} = (\lambda_s^{k+1} - \lambda_s^k)/\mu_k$, and we get the inequality

$$\| u_s^{k+1} - v^{k+1} \|_2 \leqslant \frac{\| \lambda_s^{k+1} \|_2}{\mu_k} + \frac{\| \lambda_s^k \|_2}{\mu_k} \to 0.$$

This means that $u_s^k - v^k \to 0$ for all $s = 1, \ldots, S$ and, since v^k converges, also each u_s^k converges and the corresponding limit u_s^* equals v^*, which completes the proof. □

2.6. Vector-valued data

The proposed method can be extended from scalar to vector valued data as follows. Let $u \in \mathbb{R}^{m\times n\times d}$ be a vector-valued image; that is, each pixel (i, j) carries a d-dimensional vector. For vector-valued data, the gradient operator with respect to the displacement vector $a \in \mathbb{Z}^2 \setminus \{0\}$ is understood in the sense

$$\left(\nabla_a u\right)_{ijk} = u_{(i,j)+a,k} - u_{ijk};$$

that is, we form the vector of componentwise differences. The corresponding potential function ϕ in (2) reads

$$\phi(x) = \phi_{\alpha,\gamma}(x) = \sum_{i,j} \min\left(\gamma,\ \alpha \sum_{k=1}^{d} \left| x_{ijk} \right|^{q}\right).$$

Then, the minimization problems in (16) can be decomposed into univariate (vector-valued) Mumford–Shah problems of the form

$$\min_{u\in\mathbb{R}^{n\times d}, J\subset\{1,\ldots,n-1\}} \gamma\,|J| + \alpha \sum_{i\notin J}\sum_{k=1}^{d} |u_{i+1,k} - u_{i,k}|^{q} + \sum_{i=1}^{n}\sum_{k=1}^{d} (u_{i,k} - y_{i,k})^{2}. \tag{29}$$

A global minimizer of this problem can be computed using algorithm 1 with the following modification. The approximation error on an interval $l, \ldots, r$ is now given by

$$\begin{aligned} \varepsilon'_{l,r} &= \min_{h'\in\mathbb{R}^{(r-l+1)\times d}} \sum_{k=1}^{d}\sum_{i=l}^{r-1} \alpha \left| h'_{i+1,k} - h'_{i,k} \right|^{q} + \sum_{k=1}^{d}\sum_{i=l}^{r} \left(h'_{i,k} - y_{i,k} \right)^{2} \\ &= \sum_{k=1}^{d} \min_{h\in\mathbb{R}^{r-l+1}} \left(\sum_{i=l}^{r-1} \alpha \left| (\nabla h)_i \right|^{q} + \sum_{i=l}^{r} \left(h_i - y_{i,k} \right)^{2} \right) \\ &= \sum_{k=1}^{d} \min_{h\in\mathbb{R}^{r-l+1}} \alpha \,\| \nabla h \|_q^q + \| h - (y_{l,d}, \ldots, y_{r,d}) \|_2^2. \end{aligned}$$

We observe that the vectorial approximation error $\varepsilon_{l,r}{}'$ is the sum of the scalar approximation error $\varepsilon_{l,r}$ (defined by (20)) over all channels $k = 1, \ldots, d$. Accordingly, the final reconstruction step in algorithm (1) is performed channelwise. We point out that only the approximation errors can be computed channelwise in (29); we emphasize that the overall solution of the univariate Mumford–Shah problem cannot be computed channelwise. In particular, all channels share a single jump set. (Considering separate Mumford–Shah problems on each channel typically leads to poor results as each channel may get a different jump set.) The computations in the vectorial case are around d-times as expensive as for the scalar case. Thus, our solvers for the univariate Mumford–Shah problem have time complexity $O(n^2 d)$ for $p = 2$, and $O(n^3 d)$ for $p = 1$. We emphasize that the complexity scales linearly with the length d of the vectorial data.

2.7. Complete algorithm and discussion of the parameters

A pseudocode of the complete proposed method is given in algorithm 2. It depends on the ordering of the a_s, the initialization of the split variables, the stopping criterion, and the choice of the sequence μ_k, which we discuss in the following.

We observed in our experiments that the initialization of the split variables and the ordering of the a_s have only a little influence on the final result when starting the iteration with sufficiently small coupling parameters. Here, we initialize the variables u_s and the Lagrange multipliers λ_s, $\varrho_{r,s}$ with 0, and v by $\mathrm{prox}_{d(H(\cdot),f)/(\mu_1 S)}(0)$. We use the ordering of the a_s as in (3). The choice of the coupling sequence has a certain impact on the solution;typically, the slower we choose the progression the better are the results. We will compare different progressions in the numerical part, section 3. The sequence $\mu_k = 10^{-6} \cdot k^{\beta}$ with $\beta = 2.01$ worked in all cases. For the denoising problems (which are not ill-posed) we can safely use the much faster geometric progression $\mu_k = 2^k \cdot 10^{-6}$. In all cases, we use $\nu_k = 2\mu_k/(S - 1)$ in order to give a balanced

Algorithm 2. Proposed algorithm for Mumford–Shah regularization

```
Input: Measured data $f$, model parameters $\gamma > 0$ and $\alpha > 0$
Output: Vector-valued image $u \in \mathbb{R}^{m\times n\times d}$
begin
    $k \leftarrow 1$
    Initialize $u_s$, $\lambda_s$, $\varrho_{r,s}$ with 0 and $v = \mathrm{prox}_{d(H(\cdot),f)/(\mu_k S)}(0)$
    repeat
        for $s = 1, \dots, S$ do
            $w \leftarrow \frac{\mu_k v + \lambda_s + \sum_{1\leqslant r<s}(\nu_k u_r + \varrho_{r,s}) + \sum_{s<t\leqslant S}(\nu_k u_t - \varrho_{s,t})}{\mu_k + \nu_k(S-1)}$
            $u_s \leftarrow$ Solution of 1D Mumford–Shah problems with data $w$, line-wise
            w.r.t. the direction $a_s$, using the parameters $\frac{2\omega_s\gamma}{\mu_k+\nu_k(S-1)}$ and $\frac{2\omega_s\alpha}{\mu_k+\nu_k(S-1)}$
            /* see algorithm 1 */
        end
        $v \leftarrow \mathrm{prox}_{d(H(\cdot),f)/(\mu_k S)}(\frac{1}{S}\sum_{s=1}^{S}(u_s - \frac{\lambda_s}{\mu_k}))$        / *Use warmstart, cf. section 2.4 */
        for $s = 1, \dots, S$ do
            $\lambda_s \leftarrow \lambda_s + \mu_k(v - u_s)$
            for $r = 1, \dots, s-1$ do
                $\varrho_{r,s} \leftarrow \varrho_{r,s} + \nu_k(u_r - u_s)$
            end
        end
        $k \leftarrow k+1$
    until stopping criterion reached (see Sec. 2.7)

    return $u = \frac{1}{S}(\sum_{s=1}^{S} u_s)$        / *return average of the split variables $u_s$ */
end
```

weight of the coupling for the split variables and the data term. We stop the iteration when the split variables u_1^k and u_2^k are close to each other;more precisely, we stop when the relative discrepancy $\| u_1^k - u_2^k \|_2 / (\| u_1^k \|_2 + \| u_2^k \|_2)$ falls below $\mathrm{TOL} = 10^{-3}$, or if $\| u_1^k \|_2 + \| u_2^k \|_2 < \mathrm{TOL}$. For the geometric progression, usually 20–30 iterations are required, for the quadratic progression around 10^3 iterations.

The model parameters α and γ are chosen empirically. A rule of thumb in the case $H = \mathrm{id}$ is as follows. Assume that f is a (scalar-valued) piecewise constant signal of length $n \gg \alpha$ with exactly one jump of height b at its center point. The minimizer candidate $u = f$ has the functional value γ. A candidate without jumps has the functional value $\varepsilon_{1,n}$. Thus, the jump is taken if $\gamma < \varepsilon_{1,n}$. As n is large compared to α, we get for $p = 1$ approximately $\varepsilon_{1,n} \approx b\alpha$. Thus, the critical value of γ for jump detection of height b is $\gamma = b\alpha$. The analogous critical value for $p = 2$ is given by $\gamma = \frac{1}{2}b^2\sqrt{\alpha}$; we refer to the book of Blake and Zisserman [7, chapter 4] for the derivation and a more detailed discussion. For vector valued data of dimension d, we multiply the critical values for γ by a factor d. The cutoff b is usually easier to guess than the other parameters. A typical value for b is around 0.1 to 0.3 for images with a dynamic range [0, 1]. Having chosen b, it only remains to adjust α. This is done empirically where a higher α means stronger smoothing.

Following the observations of figure 4, we use the following criterion for the choice of the exponent q in the variation penalty. On the one hand, if we aim for visually pleasing results, then the quadratic variation penalty ($q = 2$) is a suitable choice, because it favors smoothing in the low contrast regime. On the other hand, if we want to recover small structures of low contrast, which might be of interest in medical imaging, then $q = 1$ is favorable.

3. Numerical results and applications

In this section, we illustrate the wide applicability of our algorithmic framework. To this end, we apply our method to classical imaging problems; namely, we consider deconvolution, reconstruction from Radon data, and inpainting. Furthermore, we consider various noise models such as Gaussian, Laplacian, and impulsive noise. We emphasize that only the proximal mapping of the data term has to be adapted to the specific imaging setup; all other parts of the algorithm remain unchanged.

3.1. Experimental setup

In all our experiments, we use algorithm 2 with the proximal mappings corresponding to the problem under consideration. The experiments were conducted on a desktop computer (Intel Xeon E5, 6 cores, 3.5 GHz, 32 GB RAM).

In the synthetic experiments, we simulate Gaussian noise of zero mean and standard deviation σ, if not indicated otherwise. A noise level of r means $\sigma = r \cdot \| f' \|_\infty$ where f' is the clean data. The images in figures 6, 11, and 12 were taken from the Berkeley Segmentation Dataset [31]; their dynamic range is the color cube $[0,1]^3$. The PET (figure 8) dataset was taken from [18].

As a measure for reconstruction quality, we use the *mean structural similarity index* (MSSIM) [48]. The MSSIM is better suited to perceived visual quality than measures based on the L^2 error (such as, e.g., PSNR) as it takes the similarity of local structures into account. The MSSIM is bounded from above by 1 and a higher MSSIM means a better reconstruction quality. Here, we use Matlab's function `ssim` with standard parameters for the computation of the MSSIM.

3.2. Deconvolution

We first look at blurred images; that is, the measured image is convolved by a kernel K, for example a Gaussian kernel or a motion blur. Then, the corresponding data term reads

$$d(H(v), f) = \frac{1}{2} \| K * v - f \|_2^2. \tag{30}$$

We compare our method with the Ambrosio–Tortorelli approximation [3] of the Mumford–Shah functional (with $q = 2$). For deconvolution problems, it is given by the bivariate functional

$$\mathcal{A}_\varepsilon(u, v) = \gamma \int \varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \, \mathrm{d}x + \alpha \int v^2 |\nabla u|^2 \, \mathrm{d}x + \frac{1}{2} \int (K * u - f)^2 \, \mathrm{d}x,$$

see [5]. The additional variable v takes the role of an edge indicator. The parameter $\varepsilon > 0$ is an edge smoothing parameter which is chosen empirically. The approximation $\mathcal{A}_\varepsilon$ converges to the functional (1) in the Γ-sense as $\varepsilon \to 0$; see [4, 41]. We implemented the numerical scheme proposed in [5].[5] It consists of alternately minimizing $\mathcal{A}_\varepsilon$ with respect to u and v using

[5] Our implementation is partially based on the implementation by Stefan Fürtinger for the case $H = \mathrm{id}$, available at http://uni-graz.at/people/fuertins/codes.html.

iterative solvers on the Euler–Lagrange equations given by

$$\begin{aligned} 2\alpha v|\nabla u|^2 + \gamma\frac{v-1}{2\varepsilon} - 2\varepsilon\gamma\nabla^2 v = 0, \\ (K * u - f) * \widetilde{K} - 2\alpha\, \mathrm{div}\big(v^2\nabla u\big) = 0, \end{aligned} \tag{31}$$

where $\widetilde{K}(x) = K(-x)$. The first line is solved w.r.t. v using the MINRES solver and the second line is solved using conjugate gradients [5]. In both cases we used the solvers provided by Matlab with the standard tolerance and maximum 10000 iterations. We iterate until the two variables do not change any further; more precisely, if $\| u^{k+1} - u^k \|_2/(\| u^k \|_2 + 10^{-6}) \leqslant 10^{-3}$ and $\| v^{k+1} - v^k \|_2/(\| v^k \|_2 + 10^{-6}) \leqslant 10^{-3}$. From figure 5 we conclude that the Ambrosio–Tortorelli approximation is computationally demanding even for moderate filter sizes. We further observe a visible dependence on the choice of the edge smoothing parameter ε.

In contrast to the Ambrosio–Tortorelli approach (31), the proposed method (16) is based on proximal mapping which comes with some advantages: first, the method can be easily adapted to many imaging setups as implementations for proximal mapping are available for virtually every imaging setup. Second, proximal mapping can frequently be evaluated very efficiently. For deconvolution, we utilize the fast Fourier transform:

$$\mathrm{prox}_{\frac{1}{2}\|K*\cdot-f\|_2^2/\tau}(z) = \arg\min_v \frac{1}{2}\|K * v - f\|_2^2 + \frac{\tau}{2}\|v - z\|_2^2 = \mathcal{F}^{-1}\{y\}, \tag{32}$$

where

$$y_{ij} = \frac{\hat{f}_{ij}\overline{\hat{K}_{ij}} + \tau\hat{z}_{ij}}{|\hat{K}_{ij}|^2 + \tau}.$$

Here, $\hat{K}$ denotes the (discrete) Fourier transform of K, and $\mathcal{F}^{-1}\{y\}$ denotes the inverse Fourier transform of y. Figure 5 demonstrates that our method is significantly faster even for the slowest progression w.r.t. the coupling parameter μ.

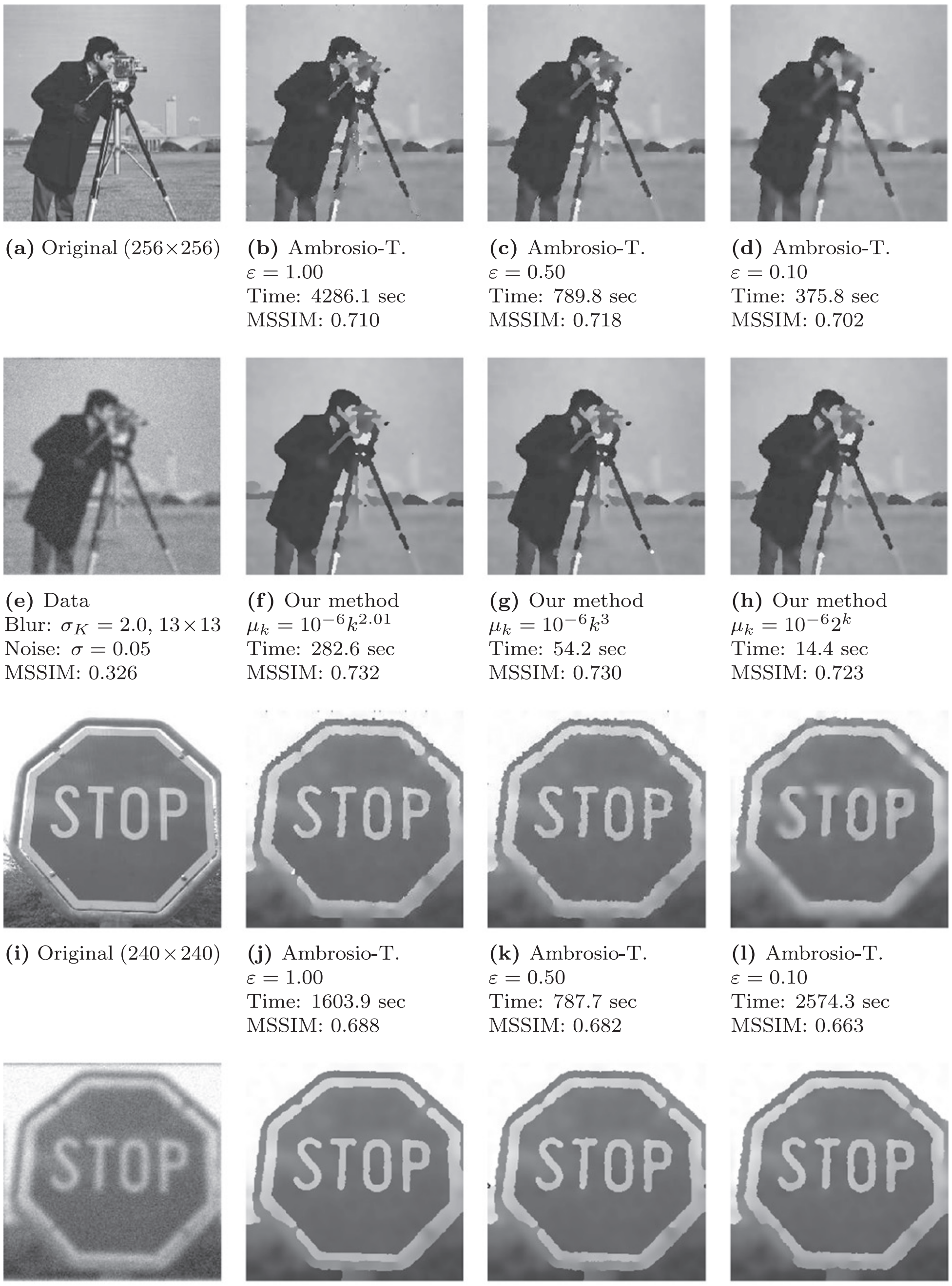
Figure 5 also shows a comparison of the different progressions w.r.t. μ_k. We obtain the best reconstruction quality for the slowest progression, $\mu_k = 10^{-6}k^{2.01}$, and the lowest run-times for the faster geometric progression, $\mu_k = 10^{-6}2^k$. In the following, we will focus on optimal reconstruction quality; therefore we will use the quasi-quadratic progression $\mu_k = k^{2.01}10^{-6}$. In the denoising experiments later on, we will utilize the fast progression $\mu_k = 2^k 10^{-6}$ because the difference of quality is negligible there.

A deconvolution example of a vector-valued image is given in figure 6. There, we compare with the classical Wiener deconvolution using the Matlab function `deconvwnr` with the regularization parameter optimized with respect to MSSIM. The CPU time is 532.4 seconds. Our method removes the noise reliably and it restores the edges.

3.3. Reconstruction from Radon data

Next, we consider reconstruction from Radon data which appears for example in x-ray computed tomography (CT) and positron emission tomography (PET). Assuming a Gaussian noise model, the according data term is given by

$$d(H(v), f) = \frac{1}{2}\|\mathcal{R}v - f\|_2^2. \tag{33}$$



(a) Original (256×256)

(b) Ambrosio-T.
$\varepsilon = 1.00$
Time: 4286.1 sec
MSSIM: 0.710

(c) Ambrosio-T.
$\varepsilon = 0.50$
Time: 789.8 sec
MSSIM: 0.718

(d) Ambrosio-T.
$\varepsilon = 0.10$
Time: 375.8 sec
MSSIM: 0.702

(e) Data
Blur: $\sigma_K = 2.0$, 13×13
Noise: $\sigma = 0.05$
MSSIM: 0.326

(f) Our method
$\mu_k = 10^{-6}k^{2.01}$
Time: 282.6 sec
MSSIM: 0.732

(g) Our method
$\mu_k = 10^{-6}k^3$
Time: 54.2 sec
MSSIM: 0.730

(h) Our method
$\mu_k = 10^{-6}2^k$
Time: 14.4 sec
MSSIM: 0.723

(i) Original (240×240)

(j) Ambrosio-T.
$\varepsilon = 1.00$
Time: 1603.9 sec
MSSIM: 0.688

(k) Ambrosio-T.
$\varepsilon = 0.50$
Time: 787.7 sec
MSSIM: 0.682

(l) Ambrosio-T.
$\varepsilon = 0.10$
Time: 2574.3 sec
MSSIM: 0.663

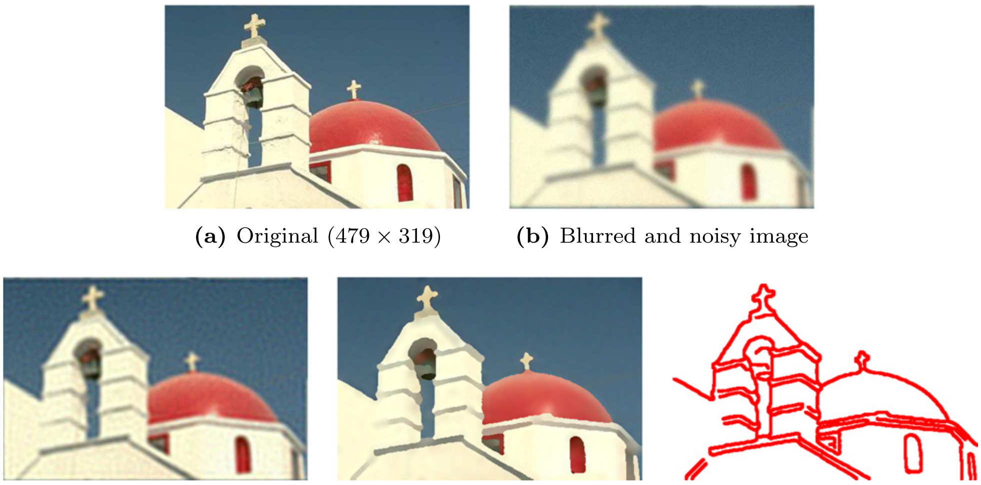
(m) Data
Blur: $\sigma_K = 3.0$, 19×19
Noise: $\sigma = 0.05$
MSSIM: 0.324

(n) Our method
$\mu_k = 10^{-6}k^{2.01}$
Time: 441.2 sec
MSSIM: 0.717

(o) Our method
$\mu_k = 10^{-6}k^3$
Time: 53.2 sec
MSSIM: 0.717

(p) Our method
$\mu_k = 10^{-6}2^k$
Time: 15.4 sec
MSSIM: 0.702

Figure 5. Comparison of our method (rows 2 and 4) with the Ambrosio–Tortorelli approximation (rows 1 and 3). The image is blurred by a Gaussian kernel of width σ_K and corrupted by Gaussian noise of level σ. Our method produces comparable or slightly better results within a significantly shorter computational time.



(a) Original (479 × 319) **(b)** Blurred and noisy image

(c) Wiener deconvolution (MSSIM: 0.883) **(d)** Our result (MSSIM: 0.920) **(e)** Induced edge set

Figure 6. Deconvolution of an image blurred by a Gaussian kernel and corrupted by Gaussian noise of level $\sigma = 0.025$. The Mumford–Shah approach (quadratic variation) removes the noise reliably and preserves the sharp edges.

Here, $\mathcal{R}$ is the Radon transform defined by

$$\mathcal{R}u(\theta, s) = \int_{-\infty}^{\infty} u\left(s\theta + t\theta^{\perp}\right) \mathrm{d}t,$$

where $s \in \mathbb{R}$, $\theta \in S^1$, and $\theta^{\perp}$ is the unit vector $\pi/2$ radians counterclockwise from θ; see e.g. [34].

The proximal mapping of the data term can be written in closed form. It is a filtered backprojection algorithm with a filter function given in the following. If $f = \mathcal{R}w$ for some $w \in L^2(\Omega)$, then the proximal mapping reads

$$\operatorname{prox}_{\frac{1}{2}\|\mathcal{R}\cdot - f\|_2^2/\tau}(z) = \mathcal{R}^*\Psi_\tau(f - \mathcal{R}z), \tag{34}$$

where the filtering operator Ψ_τ is defined by

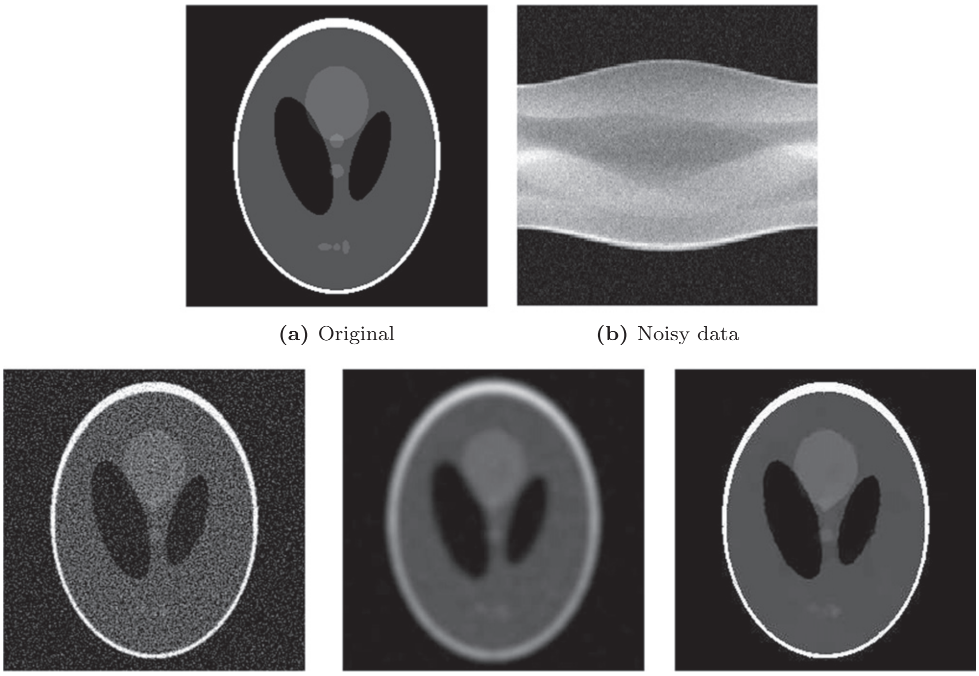
$$\Psi_\tau f = \mathcal{F}_s^{-1}\left(\psi_\tau \mathcal{F}_s f\right)$$

with the filter function

$$\psi_\tau(r) = \frac{|r|}{4\pi + \tau\,|r|}.$$

Above, $\mathcal{F}_s$ (and $\mathcal{F}_s^{-1}$) denotes the one dimensional Fourier transform (and its inverse) of a function $f(\theta, s)$ with respect to the parameter s. (A proof of (34) can be found e.g. in [45]).

In the experiments, we show the filtered backprojection (FBP) reconstructions using the standard Ram–Lak ramp filter. In figure 7, we see a reconstruction of the Shepp–Logan phantom from noisy Radon data. The CPU time was 143.7 min. The FBP amplifies the noise of the sinogram. The regularized FBP (Hamming window, cutoff frequency optimized with respect to MSSIM) smoothes out the edges and reduces the contrast. Our method removes the noise and preserves the edges.



(a) Original **(b)** Noisy data

(c) FBP (MSSIM: 0.135) **(d)** Optimized FBP (MSSIM: 0.701) **(e)** Our result (MSSIM: 0.950)

Figure 7. Reconstruction of the Shepp–Logan head phantom from Radon data (1000 projection angles, 10% noise). The filtered backprojection propagates the noise to the reconstruction; the optimized version smoothes out the edges and reduces the contrast. The Mumford–Shah method ($q = 1$) removes the noise and it preserves the edges.

In figure 8, we reconstruct a physical thorax phantom from real PET data. The CPU time was 36.4 minutes. Our method reliably removes the noise. We also see that the induced edge set gives a segmentation of the anatomic structures.

3.4. Inpainting

Let us assume that we are given high-quality data on a set of pixels Ω but the data on the complement of Ω is missing or unreliable. Then, a natural data term is given by

$$d(H(v), f) = \begin{cases} 0, & \text{if } v = f \text{ on } \Omega, \\ \infty, & \text{else}. \end{cases}$$

The proximal mapping of this functional has the simple form

$$\operatorname{prox}_{d(H(\cdot),f)/\tau}(z)_{ij} = \begin{cases} f_{ij}, & \text{if } (i, j) \in \Omega, \\ z_{ij}, & \text{else}. \end{cases} \tag{35}$$

In figure 9, we see the inpainting of a large region using the Mumford–Shah functional. We see that the geometric structures are continued reasonably and also color gradients propagate to the inpainted region. The triple junction and the crack tip are typical results of Mumford–Shah priors.

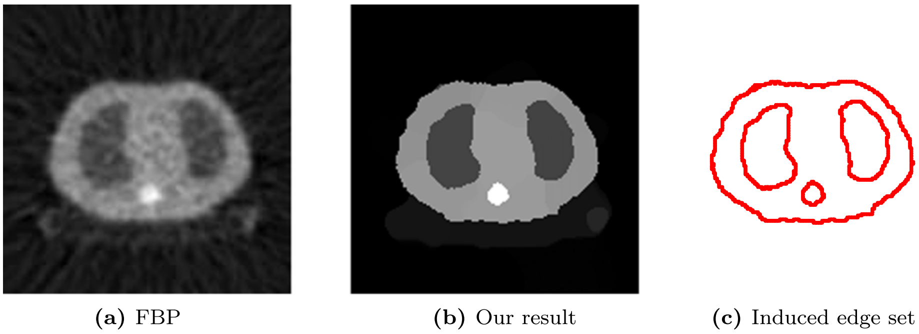

(a) FBP (b) Our result (c) Induced edge set

Figure 8. Reconstruction of a physical thorax phantom from PET data [18]. The filtered backprojection propagates the noise of the data onto the reconstruction. The Mumford–Shah method ($q = 1$) suppresses the noise and preserves the edges. The edge set can be used for segmentation of the anatomic structures (lung, spine, and thorax body).

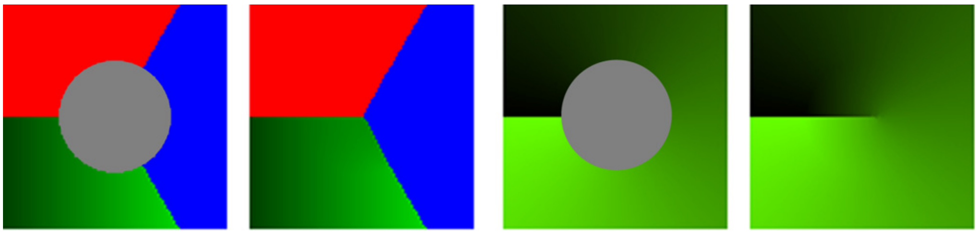


Figure 9. Inpainting of the gray regions using the Mumford–Shah functional. *Left:* the jumps are reasonably continued and they meet at the center in a triple junction. Note also the continuation of the color gradient in the green part. *Right:* the inpainting results in a so called crack tip. Both figures are typical effects of the Mumford–Shah prior.

If data on Ω is additionally corrupted by noise, then a joint inpainting and denoising is desirable. For Gaussian noise, this can be modeled by a weighted L^2 norm as data term. It is defined by

$$d(H(v), f) = \frac{1}{2}\|v - f\|_{2,w}^2 = \frac{1}{2}\sum_{i,j} w_{ij}\, |v_{ij} - f_{ij}|^2.$$

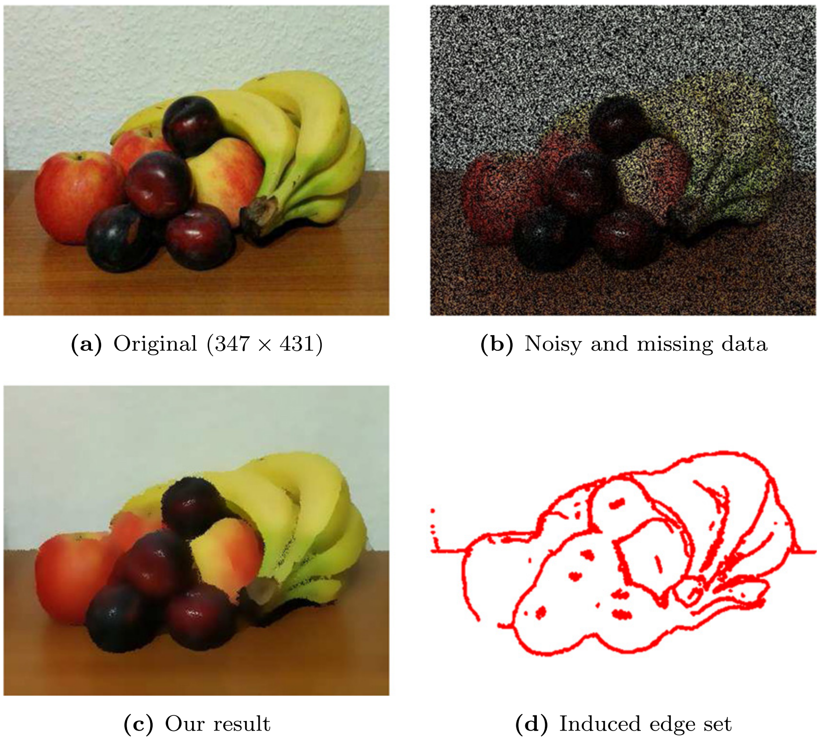
Here w is an array of nonnegative weights. We set the weight w_{ij} equal to zero whenever a pixel (i, j) is marked as missing and equal to one otherwise. The corresponding proximal mapping reads

$$\operatorname{prox}_{\frac{1}{2}\|\cdot - f\|_{2,w}^2/\tau}(z) = \arg\min_v \frac{1}{2}\| v - f \|_{2,w}^2 + \frac{\tau}{2}\|v - z\|_2^2.$$

Pointwise minimization leads to

$$\left(\operatorname{prox}_{\frac{1}{2}\|\cdot - f\|_{2,w}^2/\tau}(z)\right)_{ij} = \frac{w_{ij}f_{ij} + \tau z_{ij}}{w_{ij} + \tau},$$

which is a weighted average of f and z. In figure 10, we see the reconstruction of a noisy image where a large portion of the data is missing. Using the Mumford–Shah functional we obtain a smooth result while the edges remain sharp. The CPU time was 82.4 s.



(a) Original (347×431) **(b)** Noisy and missing data

(c) Our result **(d)** Induced edge set

Figure 10. Joint inpainting and denoising using the Mumford–Shah functional ($\sigma = 0.02$, 60% missing data). The image is continued smoothly to the missing data and the large edges remain sharply localized.

3.5. Non-Gaussian noise

We now look at non-Gaussian noise models, where we focus on the case $H = \text{id}$. For non-Gaussian noise, other data fidelities than the L^2 norm typically perform better.

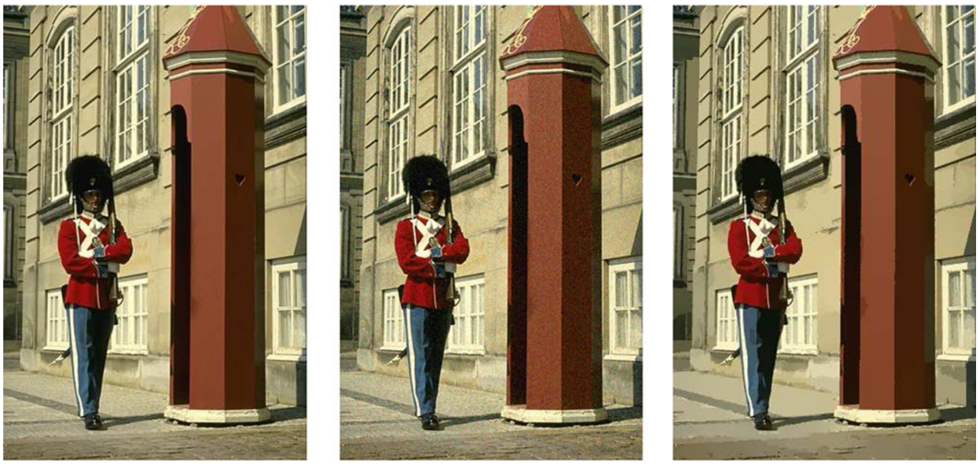
For noise distributions with heavy tails, e.g. the Laplace distribution, L^1 data fitting terms are more robust. We here consider a weighted L^1 norm, i.e.,

$$d(H(v), f) = \| v - f \|_{1,w} = \sum_{i,j} w_{ij} |v_{ij} - f_{ij}|$$

where w is an array of nonnegative weights. The corresponding proximal mapping is given by

$$\left(\text{prox}_{\|\cdot - f\|_{1,w}/\tau}(z)\right)_{ij} = T_{w_{ij}/\tau}\left(z_{ij} - f_{ij}\right) + f_{ij}$$

with the soft threshold function $T_r(x) = \max(|x| - r, 0)\text{sign}(x)$ for $r > 0$, $x \in \mathbb{R}$. We refer to [15] for a method to compute the proximal point for the case of H being a linear operator. In figure 11, we see the restoration from Laplacian noise using the Mumford–Shah functional with $q = 1$. The CPU time was 520.6 s. We observe that the noise is removed while fine details are preserved.



(a) Original (481 × 321)

(b) Laplacian noise $\sigma = 0.05$, (MSSIM: 0.893)

(c) Our result (MSSIM: 0.958)

Figure 11. Restoration of an image from Laplacian noise using the Mumford–Shah functional ($q = 1$) and L^1 data term. The noise is removed and structures of small contrast are preserved.

If we have impulsive noise, for example salt and pepper noise, we consider an ℓ^0 data fidelity of the form

$$d(H(v), f) = \| v - f \|_{0,w} = \sum_{i,j} w_{ij} \left(1 - \delta_{v_{ij} - f_{ij}} \right) \tag{36}$$

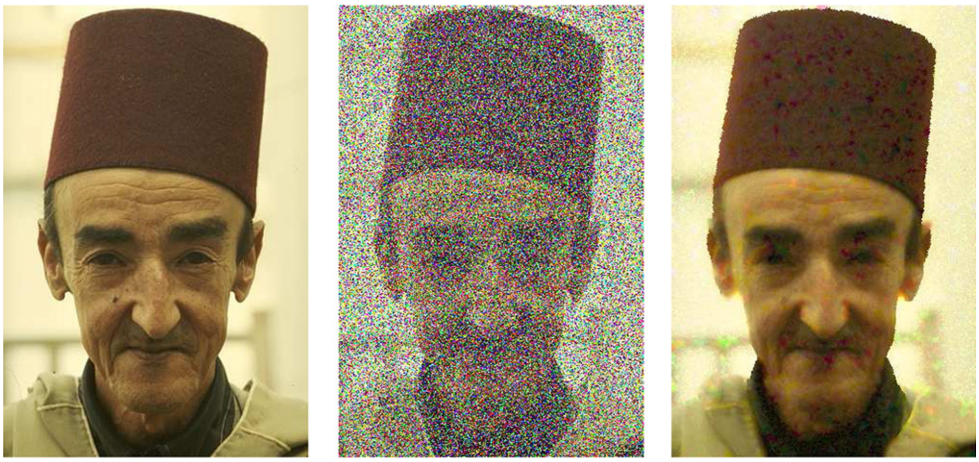
where δ denotes the Kronecker delta, i.e., $\delta_0 = 1$ and $\delta_t = 0$ for all $t \neq 0$. Note that the data fidelity is non-smooth and non-convex. So computing the proximal point is a challenging problem on its own for general imaging operators. Nevertheless, in the considered case $H = \mathrm{id}$, the proximal mapping has a closed form, which we obtain as follows. The corresponding minimization problem is separable. So the computation of the proximal mapping reduces to computing $\arg\min_{r\in\mathbb{R}} 2w_{ij}(1 - \delta_{r-f_{ij}})/\tau + (z_{ij} - r)^2$ for all i, j separately. For $r = f_{ij}$, the first term equals zero and the functional value equals $(z_{ij} - f_{ij})^2$. Otherwise, the functional value is given by $2w_{ij}/\tau + (z_{ij} - r)^2$ which attains its minimal value $2w_{ij}/\tau$ at $r = z_{ij}$. Hence, we get, for the proximal mapping of the ℓ^0 data fidelity, that

$$\left(\mathrm{prox}_{\|\cdot - f\|_{0,w}/\tau}(z) \right)_{ij} = \begin{cases} z_{ij}, & \text{if } \left(z_{ij} - f_{ij} \right)^2 \geqslant 2w_{ij}/\tau, \\ f_{ij}, & \text{else.} \end{cases}$$

This is a hard thresholding operation. In figure 12 we see the restoration from salt and pepper noise. Although the high noise level creates some color distortions, we obtain a reconstruction where many details can be recognized. The CPU time was 96.5 s.

4. Conclusion and outlook

We have established an algorithmic framework for Mumford–Shah regularization of inverse problems in imaging. Our algorithms were based on a specific splitting into tractable subproblems. For those subproblems, we have derived efficient solvers. We have obtained very



(a) Original (475 × 321) (b) 50 % salt and pepper noise (MSSIM: 0.076) (c) Our result (MSSIM: 0.880)

Figure 12. Restoration from salt and pepper noise using the Mumford–Shah functional ($q = 2$) with ℓ^0 data term. Many small details such as the wrinkles are recovered and the segment boundaries are sharp. The high noise level causes some color artifacts.

satisfactory results for classical image recovery problems. In particular, we have demonstrated the wide applicability of the method. It can be easily applied to further imaging problems because only the proximal mapping of the data term has to be adapted. Some potential fields of further application are magnetic particle imaging, photoacoustic tomography, or disparity estimation. Our future research includes a further speedup of the solvers and parallelization on the GPU.

Acknowledgments

Martin Storath has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013)/ERC grant agreement no. 267439. Andreas Weinmann acknowledges the support of the Helmholtz Association within the young investigator group VH-NG-526. They also acknowledge the support by the DFG scientific network Mathematical Methods in Magnetic Particle Imaging. We would like to thank Jeffrey Fessler for making his PET dataset publicly available and Stefan Fürtinger for making his implementation of the Ambrosio–Tortorelli approximation available.

References

[1] Alexeev B and Ward R 2010 On the complexity of Mumford–Shah-type regularization, viewed as a relaxed sparsity constraint *IEEE Trans. Image Process.* **19** 2787–9

[2] Ambrosio L, Fusco N and Pallara D 2000 *Functions of Bounded Variation and Free Discontinuity Problems* (Oxford: Clarendon Press)

[3] Ambrosio L and Tortorelli V 1990 Approximation of functional depending on jumps by elliptic functional via Γ-convergence *Commun. Pure Appl. Math.* **43** 999–1036

[4] Ambrosio L and Tortorelli V 1992 On the approximation of functionals depending on jumps by quadratic, elliptic functionals *Boll. Unione Mat. Ital.* **6-B** 105–23

[5] Bar L, Sochen N and Kiryati N 2006 Semi-blind image restoration via Mumford–Shah regularization *IEEE Trans. Image Processing* **15** 483–493

[6] Blake A 1989 Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction *IEEE Trans. Pattern Anal. Mach. Intell.* **11** 2–12

[7] Blake A and Zisserman A 1987 *Visual Reconstruction* (Cambridge: MIT Press)

[8] Boyd S, Parikh N, Chu E, Peleato B and Eckstein J 2011 Distributed optimization and statistical learning via the alternating direction method of multipliers *Found. Trends Mach. Learn.* **3** 1–122

[9] Boyd S and Vandenberghe L 2004 *Convex Optimization* (Cambridge: Cambridge University Press)

[10] Boykov Y, Veksler O and Zabih R 2001 Fast approximate energy minimization via graph cuts *IEEE Trans. Pattern Anal. Mach. Intell.* **23** 1222–39

[11] Cai X, Fitschen J, Nikolova M, Steidl G and Storath M 2015 Disparity and optical flow partitioning using extended Potts priors *Inf. Inference* **4** 43–62

[12] Chambolle A 1995 Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations *SIAM J. Appl. Math.* **55** 827–63

[13] Chambolle A 1999 Finite-differences discretizations of the Mumford–Shah functional *ESAIM: Math. Modelling Numer. Anal.* **33** 261–88

[14] Chambolle A and Pock T 2011 A first-order primal-dual algorithm for convex problems with applications to imaging *J. Math. Imaging Vis.* **40** 120–45

[15] Clason C, Jin B and Kunisch K 2010 A semismooth Newton method for L^1 data fitting with automatic choice of regularization parameters and noise calibration *SIAM J. Imaging Sci.* **3** 199–231

[16] Cremers D, Rousson M and Deriche R 2007 A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape *Int. J. Comput. Vis.* **72** 195–215

[17] Davies P and Kovac A 2001 Local extremes, runs, strings and multiresolution *Ann. Stat.* **29** 1–48

[18] Fessler J, Ficaro E, Clinthorne N and Lange K 1997 Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction *IEEE Trans. Med. Imaging* **16** 166–75 http://web.eecs.umich.edu/~fessler/result/tr/pet,trans,2d,sino/

[19] Fornasier M, March R and Solombrino F 2013 Existence of minimizers of the Mumford–Shah functional with singular operators and unbounded data *Ann. Mat. Pura Appl.* **192** 361–91

[20] Fornasier M and Ward R 2010 Iterative thresholding meets free-discontinuity problems *Found. Comput. Math.* **10** 527–67

[21] Friedrich F, Kempe A, Liebscher V and Winkler G 2008 Complexity penalized M-estimation *J. Comput. Graph. Stat.* **17** 201–24

[22] Geman S and Geman D 1984 Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images *IEEE Trans. Pattern Anal. Mach. Intell.* **6** 721–41

[23] Grasmair M 2007 The equivalence of the taut string algorithm and BV-regularization *J. Math. Imaging Vis.* **27** 59–66

[24] Jiang M, Maass P and Page T 2014 Regularizing properties of the Mumford–Shah functional for imaging applications *Inverse Problems* **30** 035007

[25] Killick R, Fearnhead P and Eckley I 2012 Optimal detection of changepoints with a linear computational cost *J. Am. Stat. Assoc.* **107** 1590–8

[26] Kim J, Tsai A, Cetin M and Willsky A 2002 A curve evolution-based variational approach to simultaneous image restoration and segmentation *Proc. of the IEEE Int. Conf. on Image Processing* **1** 109

[27] Klann E 2011 A Mumford–Shah-like method for limited data tomography with an application to electron tomography *SIAM J. Imaging Sci.* **4** 1029–48

[28] Klann E, Ramlau R and Ring W 2011 A Mumford–Shah level-set approach for the inversion and segmentation of SPECT/CT data *Inverse Probl. Imaging* **5** 137–66

[29] Kolehmainen V, Lassas M and Siltanen S 2008 Limited data x-ray tomography using nonlinear evolution equations *SIAM J. Sci. Comput.* **30** 1413–29

[30] Mammen E and van de Geer S 1997 Locally adaptive regression splines *Ann. Stat.* **25** 387–413

[31] Martin D, Fowlkes C, Tal D and Malik J 2001 A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics *Proc. of the 8th Int. Conf. on Computer Vision* 416–23 http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

[32] Mumford D and Shah J 1985 Boundary detection by minimizing functionals *IEEE Conf. on Computer Vision and Pattern Recognition* **17** 137–54

[33] Mumford D and Shah J 1989 Optimal approximations by piecewise smooth functions and associated variational problems *Commun. Pure Appl. Math.* **42** 577–685

[34] Natterer F 1986 *The Mathematics of Computerized Tomography* (Stuttgart: Teubner)

[35] Nocedal J and Wright S 2006 *Numerical Optimization* (New York: Springer)

[36] Pock T, Cremers D, Bischof H and Chambolle A 2009 An algorithm for minimizing the Mumford–Shah functional *IEEE Int. Conf. on Computer Vision* 1133–40

[37] Ramlau R and Ring W 2007 A Mumford–Shah level-set approach for the inversion and segmentation of X-ray tomography data *J. Comput. Phys.* **221** 539–57

[38] Ramlau R and Ring W 2010 Regularization of ill-posed Mumford–Shah models with perimeter penalization *Inverse Problems* **26** 115001

[39] Rockafellar R 1976 Monotone operators and the proximal point algorithm *SIAM J. Control Optim.* **14** 877–98

[40] Rondi L 2008 On the regularization of the inverse conductivity problem with discontinuous conductivities *Inverse Probl. Imaging* **2** 397–409

[41] Rondi L and Santosa F 2001 Enhanced electrical impedance tomography via the Mumford–Shah functional *ESAIM: Control Optim. Calc. Var.* **6** 517–38

[42] Steidl G, Didas S and Neumann J 2005 Relations between higher order TV regularization and support vector regression *In Scale Space and PDE Methods in Computer Vision* (Berlin: Springer) 515–27

[43] Storath M and Weinmann A 2014 Fast partitioning of vector-valued images *SIAM J. Imaging Sci.* **7** 1826–52

[44] Storath M, Weinmann A and Demaret L 2014 Jump-sparse and sparse recovery using Potts functionals *IEEE Trans. Signal Process.* **62** 3654–66

[45] Storath M, Weinmann A, Frikel J and Unser M 2015 Joint image reconstruction and segmentation using the Potts model *Inverse Problems* **31** 025003

[46] Strekalovskiy E, Chambolle A and Cremers D 2012 A convex representation for the vectorial Mumford–Shah functional *IEEE Conf. on Computer Vision and Pattern Recognition* (*CVPR*) 1712–9

[47] Tsai A, Yezzi A Jr and Willsky A 2001 Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification *IEEE Trans. Image Process.* **10** 1169–86

[48] Wang Z, Bovik A, Sheikh H and Simoncelli E 2004 Image quality assessment: from error visibility to structural similarity *IEEE Trans. Image Process.* **13** 600–12

[49] Weinmann A and Storath M 2015 Iterative Potts and Blake–Zisserman minimization for the recovery of functions with discontinuities from indirect measurements *Proc. R. Soc.* A **471** 2176

[50] Winkler G and Liebscher V 2002 Smoothers for discontinuous signals *J. Nonparametr. Stat.* **14** 203–22