SIAM J. Imaging Sciences
Vol. 7, No. 3, pp. 1826–1852

© 2014 Society for Industrial and Applied Mathematics

Fast Partitioning of Vector-Valued Images*

Martin Storath† and Andreas Weinmann‡

Abstract. We propose a fast splitting approach to the classical variational formulation of the image partitioning problem, which is frequently referred to as the Potts or piecewise constant Mumford–Shah model. For vector-valued images, our approach is significantly faster than the methods based on graph cuts and convex relaxations of the Potts model which are presently the state-of-the-art. The computational costs of our algorithm only grow linearly with the dimension of the data space which contrasts the exponential growth of the state-of-the-art methods. This allows us to process images with high-dimensional codomains such as multispectral images. Our approach produces results of a quality comparable with that of graph cuts and the convex relaxation strategies, and we do not need an a priori discretization of the label space. Furthermore, the number of partitions has almost no influence on the computational costs, which makes our algorithm also suitable for the reconstruction of piecewise constant (color or vectorial) images.

Key words. Potts model, piecewise constant Mumford–Shah model, vector-valued image segmentation, ADMM splitting, continuous label space, image denoising, jump sparsity

AMS subject classifications. 94A08, 68U10, 65D18, 65K10, 90C26, 90C39

DOI. 10.1137/130950367

1. Introduction.

Image partitioning is an important and challenging basic task in image processing [30, 37]. Most prominently, it appears in image segmentation where the goal is to group image parts of similar characteristics such as colors or textures in order to extract essential information from the image [65, 60, 21, 25]. It is a basic building block of almost every image processing pipeline and is particularly important in medical image analysis [56] and object detection [51], to mention only two examples. The partitioning problem also appears in the context of stereo vision, where optimal partitionings are employed for the regularization of disparity maps [9, 74, 40, 57]. Furthermore, the problem of denoising cartoon-like images may also be interpreted as a partitioning problem [54, 53].

The image partitioning problem is usually formulated as a minimization problem of a certain cost functional. This functional consists of a data fidelity term providing approximation to the data and a term taking care of the regularity of the partitioning. Here a reasonable regularity term is the total boundary length of the partitioning which leads to the classical

---

[*] Received by the editors December 23, 2013; accepted for publication (in revised form) June 11, 2014; published electronically September 23, 2014. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement 267439 and the German Federal Ministry for Education and Research under SysTec Grant 0315508.
http://www.siam.org/journals/siims/7-3/95036.html

[†] Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (martin.storath@epfl.ch).

[‡] Research Group Fast Algorithms for Biomedical Imaging, Helmholtz Center Munich, Neuherberg 85764, Germany, and Department of Mathematics, Technische Universität München, München, Germany (andreas.weinmann@helmholtz-muenchen.de). This author acknowledges support by the Helmholtz Association within the Young Investigators Group VH-NG-526.

Potts model [58, 31, 49, 4, 50, 9, 69]. It is given by the minimization problem

$$u^* = \operatorname{argmin}_u \ \gamma \cdot \|\nabla u\|_0 + \|u - f\|_2^2. \tag{1.1}$$

Here, the data f is an image taking values in $\mathbb{R}^s$, and the data fidelity is measured by an L^2 norm. u is a piecewise constant function whose jump or discontinuity set encodes the boundaries of the corresponding partitioning. Since the partition boundaries agree with the support of the gradient ∇u (taken in the distributional sense), we use the symbol $\|\nabla u\|_0$ to denote the length of the partition boundaries induced by u (always assuming that these boundaries are sufficiently regular). In contrast to this rather technical interpretation in the continuous domain setting, the simplest interpretation of the symbol $\|\nabla u\|_0$ in a discrete setting is as the support size of the discrete "gradient" ∇u consisting of the directional difference operators with respect to the coordinate axes.

The empirical model parameter $\gamma > 0$ controls the balance between the two penalties. A large value of γ favors few large partitions, which is often desired in the context of segmentation. For image restoration, data fidelity is usually more important. So one chooses a smaller γ.

The partitioning model (1.1) is named after R. Potts, who introduced the jump penalty in a fully discrete setting in the context of his work on statistical mechanics [58] generalizing the Ising model. Geman and Geman [31] were the first to use such kinds of functionals in the context of image segmentation. They take a statistical point of view and interpret minimizers of (1.1) as maximum a posteriori estimates (again in a fully discrete set-up). From a calculus of variation point of view the problem (1.1) has been studied in a fully continuous setting in the seminal works of Mumford and Shah [49, 50]. For this reason, the Potts model is often also called the *piecewise constant Mumford–Shah model*. In the image processing context, further early contributions are the work of Chambolle [14] and of Greig, Porteous, and Seheult [35].

1.1. State-of-the-art minimization strategies.

The multivariate Potts problem (1.1) is nonconvex, and it is NP-hard [9] (in the discrete setting). This means that finding a global minimizer is (at least presently) a computationally intractable task. Nonetheless, the problem has significant importance in image processing. For this reason, much effort has been made to find efficient approximative strategies.

A frequently used but still NP-hard simplification is to a priori choose a set of finite labels. This means that u is only allowed to take values in an a priori chosen small finite subset of $\mathbb{R}^s$. Then this simplified problem is approached by sequentially solving binary partitioning problems. These binary partitioning problems can in turn be solved efficiently by a minimal graph cut algorithm. In this context, the α-expansion algorithm of Boykov, Veksler, and Zabih [9] is the benchmark to compete with; see, e.g., the comparative study [64].

In [38, 39, 40], Hirschmüller proposes a noniterative strategy for the Potts problem which is called cost aggregation. This method sums the energies of all piecewise constant paths coming from different directions that end in the considered pixel and that attain a given discrete label in that pixel. The label with the least aggregated costs is finally assigned to that pixel. An advantage of this single-pass algorithm is its lower computational cost. However, this comes with lower quality results in comparison with graph cuts.

In recent years, algorithms based on convex relaxations of the Potts problem (1.1) have gained a lot of interest; see, e.g., [57, 44, 2, 16, 10, 33, 43]. In contrast to direct approaches, they are less affected by certain metrization errors originating from the discretization of the jump penalty. In particular, they yield better results when an accurate evaluation of the boundary length is required. This is, for example, the case in the context of inpainting of large regions [57]. As a tradeoff, their computational cost is much higher than that of graph cuts [33].

The major limitation of the above methods (graph cuts, cost aggregation, and convex relaxation) lies in the discrete label space. Typically, the computational costs grow linearly with the number of discrete labels. Since the number of discrete labels scales exponentially with the dimension s of the codomain, the computational costs grow exponentially in s. For example, for partitioning a typical multispectral image with $s = 33$ channels, one has to deal with at least 2^{33} labels. A recent advance in that regard has been achieved by Strekalovskiy, Chambolle, and Cremers [63]. They propose a convex relaxation method which has a reduced scaling in terms of the label space discretization. Its complexity is governed by the sum of the squared number of labels per dimension. In addition to complexity, another drawback of discrete label spaces is that one has to choose a discretization beforehand, which requires an appropriate guess on the expected labels.

Another approach is to limit the number of values which u may take. In contrast to the above methods, the possible values of u are not a priori restricted, but only the number of different values is bounded by a certain positive integer k. For $k = 2$, Chan and Vese [20] minimize the corresponding binary Potts model using active contours. They use a level set function to represent the partitions. This level set function evolves according to the Euler–Lagrange equations of the Potts model. A globally convergent strategy for the binary segmentation problem is presented in [18]. The active contour method for $k = 2$ was extended to vector-valued images in [19] and to larger k in [66]. We refer the reader to [23] for an overview on level set segmentation.

The partitioning methods mentioned so far mainly appear in the context of image segmentation. Here good results can be achieved despite the limitations on the codomain of u. For the restoration of piecewise constant images, however, one rather deals with many small partitions, which makes the a priori choice of discrete labels a challenging problem. To overcome these limitations Nikolova et al. [54, 53] propose methods for restoration of piecewise constant images which do not require a priori information on the number of partitions and their values. They achieve this using nonconvex regularizers which are treated using a graduated nonconvexity approach. We note that the Potts problem (1.1) does not fall into the class of problems considered in [54, 53].

Another frequently appearing method in the context of restoration of piecewise constant images is total variation minimization [59]. There the jump penalty $\|\nabla u\|_0$ is replaced by the total variation $\|\nabla u\|_1$. The arising minimization problem is convex and therefore numerically tractable with convex optimization techniques [67, 3, 34, 17, 22]. However, total variation minimization tends to produce reconstructions which do not localize the boundaries as sharply as the reconstructions based on the Potts functional (cf. the experimental section and [62]). In order to sharpen the results of total variation minimization, various techniques such as iterative reweighting [13], simplex constraints [42], and iterative thresholding [11, 12] have been proposed.

Yet another approach for the nonconvex two-dimensional Potts problem is to rewrite it as an "inverse" ℓ^0 minimization problem [26, 72, 1]. Here the cost for getting an ℓ^0 problem are equality constraints in the form of discrete Schwarz conditions [26] as well as a data term of the form $\|Au - f\|^2$, where A is a full triangular matrix. For the resulting ℓ^0 minimization problems, typically iterative thresholding methods are applied; see [5, 6] for related problems without constraints as well as [26, 1] for related minimization problems with constraints. Another approach to ℓ^0 minimization problems are the penalty decomposition methods of [45, 46, 75]. They deal with more general data terms and constraints by a two-stage iterative method. The connection with iterative hard thresholding is that the inner loop of the two-stage process is usually of iterative hard thresholding type. The difference between the hard thresholding based methods and our approach in this paper is that we do not have to deal with constraints and the full matrix A but with the nonseparable regularizing term $\|\nabla u\|_0$ instead of its separable cousin $\|u\|_0$. Hence we cannot use hard thresholding. A further difference is that we use an alternating direction method of multipliers (ADMM) approach instead of a majorization-minimization type approach.

1.2. Our contribution. In this work, we present a fast strategy for the Potts problem for vector-valued images. For an $n \times m$ image f with values in $\mathbb{R}^s$, we consider a discrete domain version of (1.1) which explicitly reads

$$u^* = \operatorname*{argmin}_{u \in \mathbb{R}^{m \times n \times s}} \Big\{ \gamma \sum_{i,j} \sum_{(a,b) \in \mathcal{N}} \omega_{ab} \cdot [u_{i,j,:} \neq u_{i+a,j+b,:}] + \sum_{i,j,k} |u_{ijk} - f_{ijk}|^2 \Big\}. \tag{1.2}$$

Here $u_{i,j,:}$ is a vector sitting at pixel (i, j), and the Iverson bracket $[\cdot]$ yields one, if the expression in brackets is true, and zero otherwise. The neighborhood system $\mathcal{N}$ and the nonnegative weights ω define a discrete boundary length of the corresponding discrete partitions. In the simplest case, we may use the coordinate unit vectors as neighborhood relation and unit weights. This corresponds to the jump penalty $\|\nabla_1 u\|_0 + \|\nabla_2 u\|_0$ which counts the nonzero elements of the directional difference operators ∇_1 and ∇_2 applied to u. Since this measures the boundary length of the partitions in the (anisotropic) Manhattan metric, the results may suffer from block artifacts; see Figure 3. In order to avoid these effects, we consider larger neighborhoods and derive appropriate weights to obtain a more isotropic discretization.

In order to approach the Potts problem (1.2), we reformulate it as a suitable constrained optimization problem. We then apply the ADMM. As a result, we obtain more accessible subproblems. The crucial point is that these subproblems reduce to computationally tractable optimization problems, namely univariate Potts problems. These univariate Potts problems can be solved quickly and exactly using dynamic programming. Here, we base our approach on the classical algorithms of [49, 50, 14] and the efficient implementation introduced in [70, 29]. In this work, we propose an acceleration strategy for the dynamic program which, in our experiments, resulted in a speed-up of the algorithm by a factor of four to five.

Univariate subproblems appear in a different form and a different context in the cost aggregation method. In [38, 39, 40] discretely labeled one-dimensional subproblems with fixed value at pixel p are used to determine the output at pixel p in a noniterative way. In contrast, our ADMM splitting approach naturally leads to the iterative solution of univariate subproblems. The data for these subproblems are not discretely labeled, and the problem has

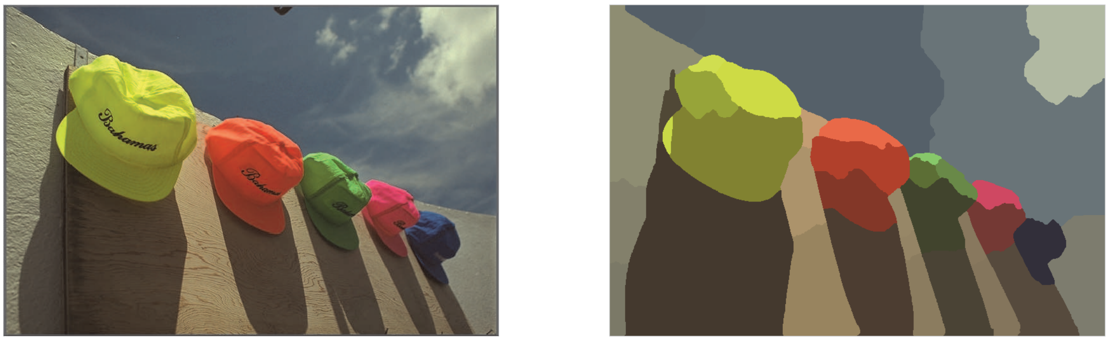


Figure 1. *The proposed method needs only* 94.0 *seconds (on a single CPU-core) for processing a large color image without having to discretize the codomain. (Left: Original,* 768×512 *pixel. Right: Our segmentation with* $\gamma = 1.0$.*) [Original image credit:* http://r0k.us/graphics/kodak/.*]*

no constraints.

The algorithm introduced in this article does not need any discretization in the codomain of u. This is an advantage compared to methods based on graph cuts, convex relaxations, and cost aggregation, which require a finite set of discrete labels.

The main feature of our algorithm is its efficiency with respect to runtime and memory consumption. One reason is that our ADMM based method for the Potts problem, in all our experiments, needs only a few iterations to converge. This observation confirms the statements in the literature which report on the high performance of ADMM methods in image processing problems [34, 52, 61, 7, 73, 62]. Another reason is that we solve the most time consuming parts of the algorithm using the highly efficient dynamic program mentioned above.

Compared to the graph cut method of [9, 8, 41] and the convex relaxation method of [57], the computational costs of our approach are significantly lower already for color images with a relatively coarsely resolved discretization of the color cube $[0,1]^3$. The advantage becomes even stronger for higher-dimensional codomains because the computational costs of our method only grow linearly in the dimension of the codomain. This contrasts with the exponential growth of the costs of the other methods. Due to the linear scaling, we can even process images taking values in a high-dimensional vector space in a reasonable time. Here a prominent example is multispectral images which may have even more than 30 channels [27, 28]. We illustrate in several experiments that our method is well suited for problems with both image segmentation (Figure 1) and the restoration of cartoon-like images (Figure 2), respectively.

We show that the proposed algorithm converges. Due to the NP-hardness of the problem, we cannot expect that the limit point is in general a minimizer of the cost function (1.2). However, in practice, we attain slightly lower functional values than graph cuts. The visual quality of our results remains at least equal to and is often even slightly better than that of graph cuts.

To support reproducibility, we provide the MATLAB/Java implementation of the algorithms under http://pottslab.de.

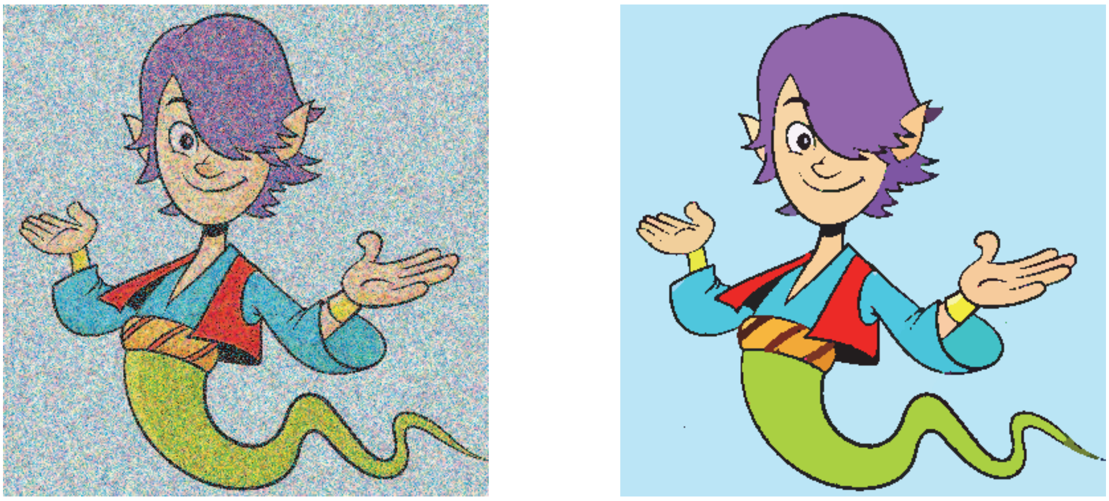


Figure 2. *Denoising a piecewise constant image using the proposed method. (Left: Noisy image with Gaussian noise of* $\sigma = 0.3$, *PSNR* 10.5. *Right: Restoration with* $\gamma = 0.9$, *PSNR* 24.4.*) ["Aladin—Genie Appear from Magic Lamp," shadowstudio* ©123RF.com*]*

1.3. Organization of the paper. We start out by presenting the basic ADMM strategy for the Potts problem in section 2. In section 3, we provide a more isotropic discretization and extensions for vector-valued images and images with missing data. In section 4, we present the dynamic program to solve the subproblems of the ADMM iteration. Numerical experiments are the subject of section 5.

2. ADMM splitting for the Potts problem.

In this section, we present a basic ADMM strategy for the Potts problem. For the sake of notational simplicity, we start with scalar-valued images $f \in \mathbb{R}^{m\times n}$ and simple (anisotropic) neighborhoods. (We elaborate on vector-valued images and on a more isotropic discretization of the jump penalty in section 3.)

Consider a four-connected neighborhood; i.e., two pixels are neighbors if only their horizontal or vertical indices differ by one. The neighborhood weights ω of (1.2) are uniformly equal to 1. Then the jump penalty reads

$$\|\nabla u\|_0 = \|\nabla_1 u\|_0 + \|\nabla_2 u\|_0 := \textstyle\sum_{i,j}[u_{ij} \neq u_{i+1,j}] + \sum_{i,j}[u_{ij} \neq u_{i,j+1}]. \tag{2.1}$$

Using this expression, we rewrite the Potts problem as the bivariate constrained optimization problem

$$\gamma\,\|\nabla_1 u\|_0 + \gamma\,\|\nabla_2 v\|_0 + \tfrac{1}{2}\|u-f\|_2^2 + \tfrac{1}{2}\|v-f\|_2^2 \to \min, \text{ s.t. } u - v = 0,$$

where $u, v \in \mathbb{R}^{m\times n}$. The augmented Lagrangian of this so-called consensus form (cf. [55]) is given by

$$\begin{aligned} L_\mu(u,v,\lambda) = \gamma\|\nabla_1 u\|_0 + \gamma\|\nabla_2 v\|_0 + \tfrac{1}{2}\|u-f\|_2^2 + \tfrac{1}{2}\|v-f\|_2^2 \\ + \langle\lambda, u-v\rangle + \tfrac{\mu}{2}\|u-v\|_2^2. \end{aligned} \tag{2.2}$$

The constraint $u - v = 0$ is now part of the target functional, and the parameter $\mu > 0$ regulates how strongly the difference between u and v is penalized. The dual variable λ is an $(m \times n)$-dimensional matrix of Lagrange multipliers, and the scalar product is given by $\langle x, y\rangle = \sum_{i,j} x_{ij} y_{ij}$. Completing the square in the last two terms of (2.2) yields

$$\begin{aligned} L_\mu(u,v,\lambda) = \gamma\|\nabla_1 u\|_0 + \gamma\|\nabla_2 v\|_0 + \tfrac{1}{2}\|u-f\|_2^2 + \tfrac{1}{2}\|v-f\|_2^2 \\ + \tfrac{\mu}{2}\|u - v + \tfrac{\lambda}{\mu}\|_2^2 - \tfrac{\mu}{2}\|\tfrac{\lambda}{\mu}\|_2^2. \end{aligned} \tag{2.3}$$

We approach this problem using ADMM. In the ADMM iteration we first fix v and λ, and we minimize $L_\mu(u,v,\lambda)$ with respect to u. Then we minimize $L_\mu(u,v,\lambda)$ with respect to v, keeping u and λ fixed. The third step can be interpreted as a gradient ascent step in the Lagrange multiplier λ. Thus, the ADMM for the Potts problem reads

$$\begin{cases} u^{k+1} & = \arg\min_u \gamma\|\nabla_1 u\|_0 + \tfrac{1}{2}\|u-f\|_2^2 + \tfrac{\mu}{2}\|u - (v^k - \tfrac{\lambda^k}{\mu})\|_2^2, \\ v^{k+1} & = \arg\min_v \gamma\|\nabla_2 v\|_0 + \tfrac{1}{2}\|v-f\|_2^2 + \tfrac{\mu}{2}\|v - (u^{k+1} + \tfrac{\lambda^k}{\mu})\|_2^2, \\ \lambda^{k+1} & = \lambda^k + \mu(u^{k+1} - v^{k+1}). \end{cases} \tag{2.4}$$

Using the identity

$$\begin{aligned} \nu(a-b)^2 + \mu(a-c)^2 &= (\nu+\mu)a^2 - 2a(\nu b + \mu c) + \nu b^2 + \mu c^2 \\ &= (\nu+\mu)\left(a - \frac{\nu b + \mu c}{\nu + \mu}\right)^2 - \frac{(\nu b + \mu c)^2}{\nu+\mu} + \nu b^2 + \mu c^2 \end{aligned} \tag{2.5}$$

with $\nu = 1$, we rewrite the first and the second lines of (2.4) to obtain

$$\begin{cases} u^{k+1} & = \arg\min_u \frac{2\gamma}{1+\mu}\|\nabla_1 u\|_0 + \|u - (1+\mu)^{-1}(f + \mu v^k - \lambda^k)\|_2^2, \\ v^{k+1} & = \arg\min_v \frac{2\gamma}{1+\mu}\|\nabla_2 v\|_0 + \|v - (1+\mu)^{-1}(f + \mu u^{k+1} + \lambda^k)\|_2^2, \\ \lambda^{k+1} & = \lambda^k + \mu(u^{k+1} - v^{k+1}). \end{cases} \tag{2.6}$$

We observe that the first line of (2.6) is separable into n subproblems of the form

$$u_{:,j}^{k+1} = \arg\min_{h\in\mathbb{R}^m} \frac{2\gamma}{1+\mu}\|\nabla h\|_0 + \|h - (1+\mu)^{-1}(f_{:,j} + \mu v_{:,j}^k - \lambda_{:,j}^k)\|_2^2 \tag{2.7}$$

for $j = 1, \ldots, n$. Likewise, the minimizer of the second line of (2.6) is given by

$$v_{i,:}^{k+1} = \arg\min_{h\in\mathbb{R}^n} \frac{2\gamma}{1+\mu}\|\nabla h\|_0 + \|h - (1+\mu)^{-1}(f_{i,:} + \mu u_{i,:}^{k+1} + \lambda_{i,:}^k)\|_2^2 \tag{2.8}$$

for $i = 1, \ldots, m$. The crucial point is that these subproblems are univariate Potts problems which can be solved exactly and efficiently using dynamic programming. We will elaborate on the solution algorithm for these subproblems in section 4.

We initialize the ADMM iteration with a small positive coupling parameter $\mu_0 > 0$ and increase it during the iteration by a factor $\tau > 1$. Hence, μ is given by the geometric progression

$$\mu = \mu_k = \tau^k \mu_0.$$

This strategy ensures that u and v can evolve quite independently at the beginning and that they are close to each other at the end of the iteration. We stop the iteration when the difference of u and v falls below some tolerance. We note that the increment of the coupling parameter is not used in standard ADMM approaches for convex optimization problems. However, we use such an increment since it has turned out to work well in our practical applications. In particular, the geometric progression yields satisfactory results while being very fast.

As is typical for energy minimization methods, there is no unique theoretically founded strategy for finding the regularization parameter γ. Intuitively, γ can be interpreted as a scale parameter: choosing a high value of γ results in a few large partitions, whereas a small γ value gives an approximation to the data having more jumps. In connection with one-dimensional Potts functionals, different approaches for an automated choice of γ are reported in the literature. For example, strategies based on Akaike's and Schwarz's information criterion are employed to estimate γ; see, e.g., the overview article [71]. Furthermore, strategies based on testing the residual for noise [24] or the interval method of Winkler et al. [71] are used. The latter strategy chooses the largest parameter interval for γ where the same solution persists. The drawback of the previous methods is their high computational cost or their specialization to one dimension which makes them (almost) not applicable in higher dimensions. Potential alternatives are general concepts from the theory of inverse problems such as the Morozov discrepancy principle [48] or the L-curve method [36]. In this paper we choose the parameter empirically.

Our approach to the Potts problem is summed up in Algorithm 1.

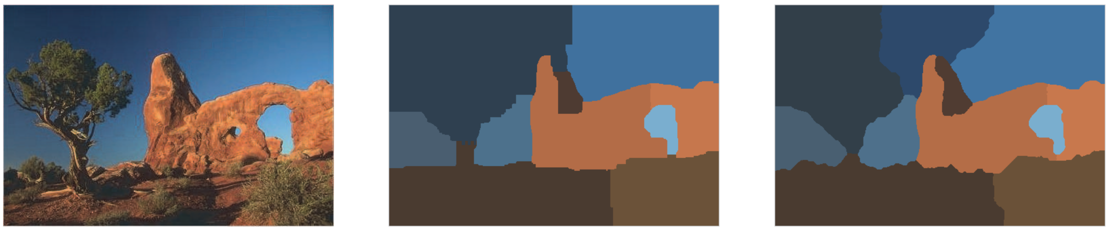
Algorithm 1: ADMM strategy to the Potts problem.

```
Input: Image $f \in \mathbb{R}^{m\times n}$, model parameter $\gamma > 0$, initial value $\mu_0 > 0$, step size $\tau > 1$.
Local: Iterated solutions $u, v \in \mathbb{R}^{m\times n}$, dual variable $\lambda \in \mathbb{R}^{m\times n}$, coupling parameter $\mu > 0$.
Output: Computed result $u \in \mathbb{R}^{m\times n}$ to the Potts problem (1.1).
begin
    $v \leftarrow f$; $\mu \leftarrow \mu_0$; $\lambda \leftarrow 0$;                                /* init */
    repeat
        for $j \leftarrow 1$ to $n$ do
            $u_{:,j} \leftarrow$ Minimizer of subproblem (2.7) using Algorithm 2;   /* cf. (section 4) */
        end
        for $i \leftarrow 1$ to $m$ do
            $v_{i,:} \leftarrow$ Minimizer of subproblem (2.8) using Algorithm 2;   /* cf. (section 4) */
        end
        $\lambda \leftarrow \lambda + \mu(u - v)$ ;                                  /* update of dual variable */
        $\mu \leftarrow \tau \cdot \mu$ ;                                         /* increase coupling parameter */
    until reached stopping criterion;
end
```

We eventually show the convergence of our ADMM strategy.

Theorem 1. *Algorithm 1 converges in the sense that there exists a u^* such that $u^k \to u^*$ and $v^k \to u^*$.*

The proof is given in Appendix A.



(a) Original (481 × 321 pixel). (b) Anisotropic discretization (13.4 sec). (c) Near-isotropic discretization (20.5 sec).

Figure 3. *The near-isotropic discretization produces smoother boundaries at the cost of about double computation time ($\gamma = 2$). [Original image credit:* http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/*.]*

3. Extensions for the ADMM splitting of the Potts problem.

We so far have explained our ADMM approach to the Potts problem with scalar-valued data and a simple neighborhood relation. We now present modifications for vector-valued images (e.g., color images) as well as a more isotropic discretization of the boundary length term $\|\nabla u\|_0$ of (1.1). We further deal with missing data points.

3.1. Increasing isotropy.

In the last chapter we used the simple anisotropic discretization of $\|\nabla u\|_0$ given by (2.1). That discretization favors region boundaries which are minimal in the Manhattan metric (compare with [14]). This may lead to block artifacts in the reconstructions; see, for example, Figure 3(b).

In order to better approximate the Euclidean length we pass to eight-connected neighborhoods. Due to symmetry, this neighborhood is given by the following four vectors: $\mathcal{N} = \{(1,0),(0,1),(-1,1),(1,1)\}$. We now derive appropriate weights ω for formula (1.2). A minimal requirement is that jumps along straight lines with respect to the compass and the diagonal directions are penalized by their Euclidean length. For reasons of symmetry, we have, for the compass weights, $\omega_c := \omega_{1,0} = \omega_{0,1}$ and, for the diagonal weights, $\omega_d := \omega_{-1,1} = \omega_{1,1}$. Now consider an $(n \times n)$ binary image u which has a jump along a straight line into a compass direction. We are searching for weights ω_c, ω_d such that the jump penalty for this image amounts to γn. Plugging u into the Potts model (1.2), we get for large n the jump penalty

$$\gamma \sum_{i,j} \sum_{(a,b)\in\mathcal{N}} \omega_{ab} \cdot [u_{ij} \neq u_{i+a,j+b}] \sim \gamma n(\omega_c + 2\omega_d).$$

Letting the right-hand side equal the desired penalty γn, we obtain the condition

$$\omega_c + 2\omega_d = 1.$$

Proceeding in the same way with the diagonal directions, we get

$$2(\omega_c + \omega_d) = \sqrt{2}.$$

Solving this linear system of equations, we obtain the weights

$$\omega_c = \sqrt{2} - 1 \quad \text{and} \quad \omega_d = 1 - \frac{\sqrt{2}}{2}.$$

We see that the above requirements already determine the weights ω_c and ω_d uniquely.

Using the above weights, we can write the jump penalty as

$$\begin{aligned}\|\nabla u\|_0 &= \omega_c\left(\|\nabla_1 u\|_0 + \|\nabla_2 u\|_0\right) + \omega_d\left(\|\nabla_{12} u\|_0 + \|\nabla_{21} u\|_0\right)\\ &:= \omega_c\left(\textstyle\sum_{i,j}[u_{ij} \neq u_{i+1,j}] + \sum_{i,j}[u_{ij} \neq u_{i,j+1}]\right)\\ &\quad + \omega_d\left(\textstyle\sum_{i,j}[u_{ij} \neq u_{i-1,j+1}] + \sum_{i,j}[u_{ij} \neq u_{i+1,j+1}]\right).\end{aligned} \tag{3.1}$$

Plugging this into (1.1), we get the following splitting of the Potts problem:

$$\gamma\omega_c\left(\|\nabla_1 u\|_0 + \|\nabla_2 v\|_0\right) + \gamma\omega_d\left(\|\nabla_{12} w\|_0 + \|\nabla_{21} z\|_0\right) + \tfrac{1}{4}(\|u-f\|_2^2 + \|v-f\|_2^2 + \|w-f\|_2^2 + \|z-f\|_2^2) \to \min,$$

subject to the constraints

$$\begin{aligned} u-v=0,\quad u-w=0,\quad u-z=0,\\ v-w=0,\quad v-z=0,\quad w-z=0.\end{aligned}$$

Notice that we now have four coupled variables u, v, w, z (instead of the two variables in section 2) and that each of these are pairwise coupled. The augmented Lagrangian then reads

$$\begin{aligned} L_\mu = {} & \gamma\omega_c\left(\|\nabla_1 u\|_0 + \|\nabla_2 v\|_0\right) + \gamma\omega_d\left(\|\nabla_{12} w\|_0 + \|\nabla_{21} z\|_0\right)\\ & + \tfrac{1}{4}(\|u-f\|_2^2 + \|v-f\|_2^2 + \|w-f\|_2^2 + \|z-f\|_2^2)\\ & + \langle\lambda_1, u-v\rangle + \tfrac{\mu}{2}\|u-v\|_2^2 + \langle\lambda_2, u-w\rangle + \tfrac{\mu}{2}\|u-w\|_2^2\\ & + \langle\lambda_3, u-z\rangle + \tfrac{\mu}{2}\|u-z\|_2^2 + \langle\lambda_4, v-w\rangle + \tfrac{\mu}{2}\|v-w\|_2^2\\ & + \langle\lambda_5, v-z\rangle + \tfrac{\mu}{2}\|v-z\|_2^2 + \langle\lambda_6, w-z\rangle + \tfrac{\mu}{2}\|w-z\|_2^2\end{aligned} \tag{3.2}$$

with the six Lagrange multipliers $\lambda_i \in \mathbb{R}^{m\times n}$, $i = 1, \dots, 6$. Some algebraic manipulation yields the ADMM iteration

$$\begin{cases} u^{k+1} &= \arg\min_u \frac{4\gamma\omega_c}{1+6\mu}\|\nabla_1 u\|_0 + \|u-u'\|_2^2,\\ w^{k+1} &= \arg\min_w \frac{4\gamma\omega_d}{1+6\mu}\|\nabla_{12} w\|_0 + \|w-w'\|_2^2,\\ v^{k+1} &= \arg\min_v \frac{4\gamma\omega_c}{1+6\mu}\|\nabla_2 v\|_0 + \|v-v'\|_2^2,\\ z^{k+1} &= \arg\min_z \frac{4\gamma\omega_d}{1+6\mu}\|\nabla_{21} z\|_0 + \|z-z'\|_2^2,\\ \lambda_i^{k+1} &= \lambda_i^k + \mu a_i,\end{cases} \tag{3.3}$$

with the data

$$\begin{aligned} u' &= \tfrac{1}{1+6\mu}[f + 2\mu(v^k + w^k + z^k) + 2(-\lambda_1^k - \lambda_2^k - \lambda_3^k)],\\ w' &= \tfrac{1}{1+6\mu}[f + 2\mu(u^{k+1} + v^k + z^k) + 2(\lambda_2^k + \lambda_4^k - \lambda_6^k)],\\ v' &= \tfrac{1}{1+6\mu}[f + 2\mu(u^{k+1} + w^{k+1} + z^k) + 2(\lambda_1^k - \lambda_4^k - \lambda_5^k)],\\ z' &= \tfrac{1}{1+6\mu}[f + 2\mu(u^{k+1} + v^{k+1} + w^{k+1}) + 2(\lambda_3^k + \lambda_5^k + \lambda_6^k)]\end{aligned}$$

and the updates

$$\begin{aligned} a_1 &= u^{k+1} - v^{k+1}, & a_2 &= u^{k+1} - w^{k+1}, \\ a_3 &= u^{k+1} - z^{k+1}, & a_4 &= v^{k+1} - w^{k+1}, \\ a_5 &= v^{k+1} - z^{k+1}, & a_6 &= w^{k+1} - z^{k+1}. \end{aligned}$$

Compared with the anisotropic version (2.6), each iteration has additional steps which however consist of the same building blocks as before. The essential difference is that we additionally solve univariate Potts problems with respect to the diagonal directions (lines 2 and 4 of (3.3)). In Figure 3, we see that the extra computational effort pays off in a visual improvement; the segment boundaries are much smoother using this discretization. Convergence of the above algorithm can be shown under the same conditions as for Algorithm 1. We omit the proof since it uses the same arguments as the proof of Theorem 1. Due to the additional Lagrange multipliers it would become excessively lengthy (without giving new insights).

We note that the isotropy can be further increased by incorporating "knight move" finite differences such as $u_{i+2,j+1} - u_{i,j}$. The corresponding neighborhood system is given by $\mathcal{N}' = \{(1,0),\ (0,1),\ (-1,1),\ (1,1),\ (-2,1),\ (-1,2),\ (1,2),\ (2,1)\}$. Due to symmetry, this system comes with three weights ω_1, ω_2, ω_3—one for the compass directions, one for the diagonal directions, and one for the knight move directions. We postulate that jumps with respect to those basic directions are measured by their Euclidean length. This leads to the system of equations

$$\begin{aligned} \omega_1 + 2\omega_2 + \ \ 6\omega_3 &= 1, \\ 2\omega_1 + 2\omega_2 + \ \ 8\omega_3 &= \sqrt{2}, \\ 3\omega_1 + 4\omega_2 + 12\omega_3 &= \sqrt{5}, \end{aligned} \tag{3.4}$$

which has the solution $\omega_1 = \sqrt{5} - 2$, $\omega_2 = \sqrt{5} - \frac{3}{2}\sqrt{2}$, and $\omega_3 = \frac{1}{2}(1 + \sqrt{2} - \sqrt{5})$. An ADMM iteration can be derived in analogy to (3.3). The iteration involves solving univariate Potts problems with respect to eight directions (two compass, two diagonal, and four knight move directions) and updating 28 Lagrange multipliers in each step.

Including finite differences with respect to diagonal directions was first proposed by Chambolle [15]. There, the quality of a set of weights is measured by the anisotropy, which is understood as the ratio between the lengths of the longest and the shortest unit vectors. It turns out that the weights derived for the eight-neighborhood coincide with the weights we have obtained with our approach (up to a normalization factor). However, when incorporating "knight move" finite differences, the weights derived by our approach are different from the weights of [15]. Using the anisotropy definition of [15], we even obtain a value of 1.03 for our weights in contrast to 1.05 for the weights in [15]. (For comparison, the eight-neighborhood weights have anisotropy 1.08.)

This approach can be extended to a general method for further increasing isotropy by passing to larger neighborhood systems. The rule for including new neighbors is to add directions to the neighborhood system if they are not yet covered. Here a direction (i, j) is not covered if its slope i/j does not yet appear in the neighborhood system. For example, the eight-neighborhood covers the slopes $0, 1, -1, \infty$. Hence, the knight moves are not yet covered since its slopes are $\pm\frac{1}{2}$ and ± 2 and they can be added to the system. The next vectors to include in the neighborhood system are $(\pm 1, 3)$, $(\pm 3, 1)$ and $(\pm 2, 3)$, $(\pm 3, 2)$, and so on. The general scheme corresponds to the standard enumeration of the rational numbers. Conditions for the weights can be derived as follows. Let us assume that u is a binary $(n \times n)$ image with an ideal jump along the direction $(x, y) \in \mathcal{N}$. We first look at lines with a slope x/y between -1 and 1 going from the left to the right boundaries of the image. (If the slope of (x, y) is not in the interval $[-1, 1]$, then we look at the $\pi/2$-rotated image and exchange the roles of x and y.) The Euclidean length of this line is given by $n\sqrt{x^2 + y^2}/x$. Since we want the total jump penalty of this image to equal that Euclidean length, we get a condition for the weights of the form

$$\gamma \sum_{i,j} \sum_{(a,b)\in\mathcal{N}} \omega_{ab} \cdot [u_{ij} \neq u_{i+a,j+b}] = n\frac{\sqrt{x^2 + y^2}}{x}.$$

It remains to evaluate the left-hand side of the equation, which can be done either manually for small neighborhood systems or with the help of a computer program for larger neighborhood systems. (When counting the weights we assume n to be very large so that boundary effects are negligible.) We then obtain a system of $|\mathcal{N}|$ equations for the $|\mathcal{N}|$ unknowns. The dimension of the system can be reduced by exploiting symmetries in the weights. We used this, for example, in (3.4) to reduce the system from 8 to 3 unknowns using that $\omega_1 = \omega_{1,0} = \omega_{0,1}$, $\omega_2 = \omega_{1,\pm 1}$, and $\omega_3 = \omega_{2,\pm 1} = \omega_{\pm 2,1}$.

3.2. Missing data. We now consider the case where some of the pixels of the image f are missing or destroyed. Since we do not want them to affect the reconstruction, we exclude them from the data penalty term. This can be formulated conveniently using a weighted L^2 norm defined by $\|x\|_w^2 = \sum_{i,j} w_{ij}|x_{ij}|^2$ with nonnegative weights w_{ij} as follows. For missing pixels $Q \subset \{1, \dots, m\} \times \{1, \dots, n\}$ the associated Potts problem is given by

$$\min_{u\in\mathbb{R}^{m\times n}} \gamma \cdot \|\nabla u\|_0 + \|u - f\|_w^2, \tag{3.5}$$

where the missing pixels are weighted by zero, i.e.,

$$w_{ij} = \begin{cases} 0 & \text{if } (i,j) \in Q, \\ 1 & \text{else.} \end{cases}$$

In analogy to (2.4) we obtain the ADMM iteration

$$\begin{cases} u^{k+1} & = \arg\min_u \gamma\|\nabla_1 u\|_0 + \frac{1}{2}\|u - f\|_w^2 + \frac{\mu}{2}\|u - (v^k - \frac{\lambda^k}{\mu})\|_2^2, \\ v^{k+1} & = \arg\min_v \gamma\|\nabla_2 v\|_0 + \frac{1}{2}\|v - f\|_w^2 + \frac{\mu}{2}\|v - (u^{k+1} + \frac{\lambda^k}{\mu})\|_2^2, \\ \lambda^{k+1} & = \lambda^k + \mu(u^{k+1} - v^{k+1}). \end{cases}$$

Here, the weights w influence only the data fidelity terms. Using identity (2.5) with $\nu = w_{ij}$, we get

$$\begin{cases} u^{k+1} & = \arg\min_u 2\gamma\|\nabla_1 u\|_0 + \|u - u'\|_{w+\mu}^2, \\ v^{k+1} & = \arg\min_v 2\gamma\|\nabla_2 v\|_0 + \|v - v'\|_{w+\mu}^2, \\ \lambda^{k+1} & = \lambda^k + \mu(u^{k+1} - v^{k+1}), \end{cases} \tag{3.6}$$

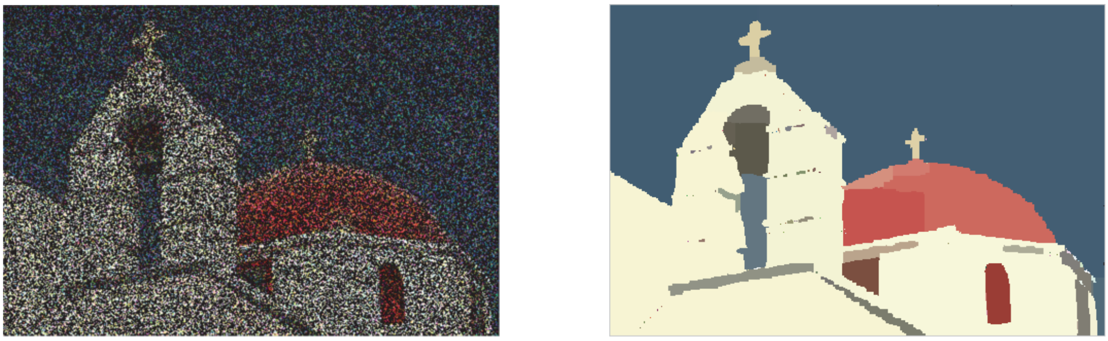


Figure 4. *Left: Image corrupted by Gaussian noise of $\sigma = 0.2$; 60% of the pixels are missing (marked as black). Right: Restoration using our method ($\gamma = 0.3$). [Original image credit:* http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/*.]*

where $u'_{ij} = \frac{w_{ij} f_{ij} - \mu v^k_{ij} - \lambda^k_{ij}}{w_{ij}+\mu}$ and $v'_{ij} = \frac{w_{ij} f_{ij} - \mu u^k_{ij} + \lambda^k_{ij}}{w_{ij}+\mu}$. Now both subproblems are univariate Potts problems with a weighted L^2 norm as the data penalty term. Figure 4 shows a restoration problem with missing and noisy data.

3.3. Vector-valued images. We now extend the proposed method to vector-valued images of the form $f \in \mathbb{R}^{m \times n \times s}$ with an $s \geq 1$. Prominent examples are color and multispectral images (Figure 5). Most of our experiments deal with color images where we have $s = 3$ and the arrays $f_{:,:,k}$, $k = 1, \dots, 3$, correspond to the red, green, and blue channels. For a multispectral image we can have about 30 channels; each channel $f_{:,:,k}$ corresponds to a certain wavelength.

For vector-valued data, the data term and the jumps term are given by

$$\|u - f\|_2^2 = \sum_{i,j,k} |u_{ijk} - f_{ijk}|^2 \quad \text{and} \quad \|\nabla u\|_0 = \sum_{(a,b)\in\mathcal{N}} \omega_{a,b} \cdot [u_{i,j,:} \neq u_{i+a,j+b,:}].$$

Here, we have a jump if two neighboring vectors $u_{i,j,:}$ and $u_{i+a,j+b,:}$ are not equal. We note that the jump penalty cannot be evaluated componentwise. This leads to the following modifications of the ADMM algorithms (2.6) and (3.3). The intermediate solutions u, v, w, z and the multipliers λ_i are in $\mathbb{R}^{m \times n \times s}$. The updates of the Lagrange multipliers can be carried out componentwise. The univariate Potts problems can be solved using the same dynamic program as for the scalar-valued case with the modifications explained in section 4.

4. Efficient solution of univariate Potts problems.

A basic building block of our ADMM algorithm is the solver of univariate Potts problems. In particular, the speed of our method heavily depends on the time consumed for solving these subproblems. In this section, we first review the basic dynamic program for solving univariate Potts problems. Then we extend the dynamic program to weighted L^2 norms (which are needed in the context of missing data) and to vector-valued data. Finally, we introduce a new effective acceleration strategy, which decreased the runtime in our experiments by a factor of about five.

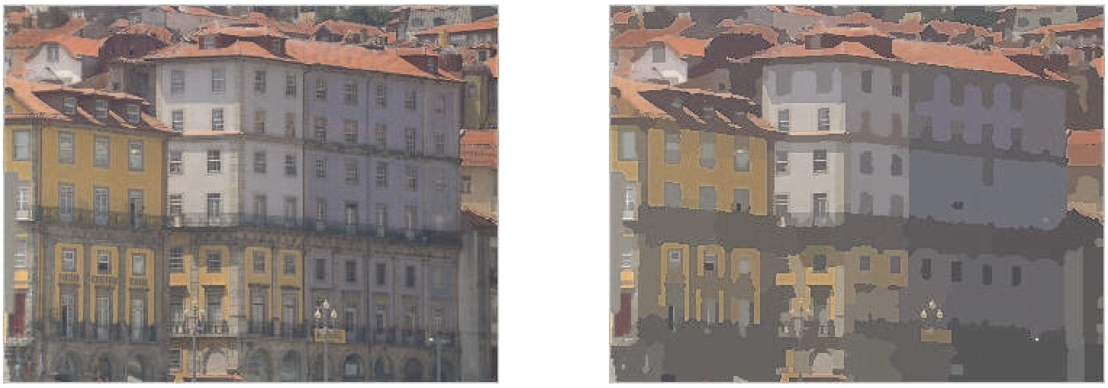


Figure 5. *Our method processes large multispectral images in reasonable time (here* 76.0 *seconds for all* 33 *channels). Left: RGB representation of a multispectral image* [27, 28] *with* $s = 33$ *channels and* 335×255 *pixels. Right: RGB representation of the result of our method using all channels* ($\gamma = 0.125$)*. [This image was originally published in* Visual Neuroscience [28] *and appears here with the permission of Cambridge University Press.]*

4.1. The classical dynamic program for the solution of the univariate Potts problem.

The classical univariate Potts problem is given by

$$P_\gamma(h) = \gamma \cdot \|\nabla h\|_0 + \|h - f\|_2^2 \to \min, \tag{4.1}$$

where $h, f \in \mathbb{R}^n$ and $\|\nabla h\|_0 = \sum_i [h_i \neq h_{i+1}]$ denotes the number of jumps of h. This optimization problem can be solved exactly using dynamic programming [49, 50, 14, 70, 29, 68]. The basic idea is that a minimizer of the Potts functional for data $(f_1, \dots, f_r)$ can be computed in polynomial time provided that minimizers of the partial data $(f_1), (f_1, f_2), \dots, (f_1, \dots, f_{r-1})$ are known. We denote the respective minimizers for the partial data by $h^1,\ h^2, \dots, h^{r-1}$. In order to compute a minimizer for data $(f_1, \dots, f_r)$, we first create a set of r minimizer candidates $g^1, \dots, g^r$, each of length r. These minimizer candidates are given by

$$g^\ell = (h^{\ell-1}, \underbrace{\mu_{[\ell,r]}, \dots, \mu_{[\ell,r]}}_{\text{Length } r-\ell+1}), \tag{4.2}$$

where h^0 is the empty vector and $\mu_{[\ell,r]}$ denotes the mean value of data $f_{[\ell,r]} = (f_\ell, \dots, f_r)$. Among the candidates g^ℓ, one with the least Potts functional value is a minimizer for the data $f_{[1,r]}$.

In [29] Friedrich et al. proposed the following $O(n^2)$ time and $O(n)$ space algorithm. They observed that the functional values of a minimizer $P_\gamma(h^r)$ for data $f_{[1,r]}$ can be computed directly from the functional values $P_\gamma(h^1), \dots, P_\gamma(h^{r-1})$ of the minimizers $h^1, \dots, h^{r-1}$ and the squared mean deviations of the data $f_{[1,r]}, \dots, f_{[r,r]}$. Indeed, using (4.2), the Potts functional value of the minimizer h^r is given (setting $P_\gamma(h^0) = -\gamma$) by

$$P_\gamma(h^r) = \min_{\ell=1,\dots,r} P_\gamma(h^{\ell-1}) + \gamma + d_{[\ell,r]}, \tag{4.3}$$

where $d_{[\ell,r]}$ denotes the squared deviation from the mean value

$$d_{[\ell,r]} = \min_{y\in\mathbb{R}} \|y - f_{[\ell,r]}\|_2^2 = \|\mu_{[\ell,r]} - f_{[\ell,r]}\|_2^2.$$

The evaluation of (4.3) is $O(1)$ if we precompute the first and second moments of data $f_{[\ell,r]}$. If ℓ^* denotes the minimizing argument in (4.3), then ℓ^*-1 indicates the rightmost jump location at step r, which is stored as $J(r)$. The jump locations of a solution h^r are thus $J(r)$, $J(J(r))$, $J(J(J(r))), \ldots$; the values of h^r between two consecutive jumps are given by the mean value of data f on this interval. Note that we have only to compute and store the jump locations $J(r)$ and the minimal Potts functional value $P_\gamma(h^r)$ in each iteration. The reconstruction of the minimizer from the jump locations only has to be done once for h^n at the end; it is thus uncritical.

4.2. Weighted data terms and vector-valued data. In order to apply the above dynamic program to univariate Potts problems with weighted data terms, we need the following modifications. The symbol $\mu_{[\ell,r]}$ is now used to denote the weighted mean given by

$$\mu_{[\ell,r]} = \arg\min_{y\in\mathbb{R}} \|y - f_{[\ell,r]}\|_w^2 = \frac{\sum_{i=\ell}^r w_i f_i}{\sum_{i=\ell}^r w_i}. \tag{4.4}$$

A straightforward computation shows that the weighted squared mean deviation $d_{[\ell,r]}$ is given by

$$d_{[\ell,r]} = \min_{y\in\mathbb{R}} \|y - f_{[\ell,r]}\|_w^2 = \sum_{i=\ell}^r w_i f_i^2 - \frac{1}{\sum_{i=\ell}^r w_i}\left(\sum_{i=\ell}^r w_i f_i\right)^2.$$

(If all weights are zero, then we let $\mu_{[\ell,r]} = 0$ and $d_{[\ell,r]} = 0$.) Toward an efficient evaluation of this expression, we rewrite it as

$$d_{[\ell,r]} = S_r - S_{\ell-1} - \frac{(M_r - M_{\ell-1})^2}{W_r - W_{\ell-1}}, \tag{4.5}$$

where $W_r = \sum_{i=1}^r w_i$, $M_r = \sum_{i=1}^r w_i f_i$, and $S_r = \sum_{i=1}^r w_i f_i^2$. The vectors W, M, and S can be precomputed in linear time. Thus, the evaluation of the weighted squared mean deviation costs $O(1)$.

If the data is vector-valued, i.e., if each f_i is in $\mathbb{R}^s$, then the vector-valued (weighted) mean value is just given by componentwise application of (4.4). Then the corresponding squared mean deviation reads

$$d_{[\ell,r]} = \sum_{k=1}^s \left(\sum_{i=\ell}^r w_i f_{i,k}^2 - \frac{1}{\sum_{i=\ell}^r w_i}\left(\sum_{i=\ell}^r w_i f_{i,k}\right)^2 \right). \tag{4.6}$$

Precomputing moments, using (4.5) componentwise, and summing up the results according to (4.6) moments allows us to evaluate this expression in $O(s)$. For vector-valued data, the total complexity of the univariate Potts problem is $O(n^2 s)$ time and $O(ns)$ space.

Algorithm 2: Accelerated dynamic program for the univariate Potts problem.

```
Input: Data vector $f \in \mathbb{R}^n$, model parameter $\gamma > 0$, weights $w \in [0, \infty)^n$
Output: Global minimizer $h$ of the univariate Potts problem (4.1) with weighted data term
Local : Left and right interval bounds $\ell, r \in \mathbb{N}$; Potts values $P \in \mathbb{R}^n$; first and second cumulative
        moments $M, S \in \mathbb{R}^{n+1}$; cumulative weights $W \in \mathbb{R}^{n+1}$; temporary values $p, d \in \mathbb{R}$ (candidate
        Potts value, squared mean deviation); array of rightmost jump locations $J \in \mathbb{N}^n$;
begin
    /* Find the optimal jump locations */
    $M_0 \leftarrow 0$; $S_0 \leftarrow 0$; $W_0 \leftarrow 0$ ;          /* init cumulative moments and weights */
    for $r \leftarrow 1$ to $n$ do
        $M_r \leftarrow M_{r-1} + w_r f_r$;                          /* First moments */
        $S_r \leftarrow S_{r-1} + w_r f_r^2$;                        /* Second moments */
        $W_r \leftarrow W_{r-1} + w_r$;                              /* Cumulative weights */
        $P_r \leftarrow S_r - \frac{1}{W_r} M_r^2$ ;                 /* mean sq. deviation of $f_{[1,r]}$ */
        $J_r \leftarrow 0$ ;                                         /* init rightmost jump location */
        for $\ell \leftarrow r$ to 2 do
            $d \leftarrow S_r - S_{\ell-1} - \frac{1}{W_r - W_{\ell-1}} (M_r - M_{\ell-1})^2$;   /* mean sq. deviation of $f_{[\ell,r]}$ */
            if $P_r < d + \gamma$ then Break;                        /* Acceleration (cf. Theorem 2) */
            ;
            $p \leftarrow P_{\ell-1} + \gamma + d$ ;                 /* compute candidate Potts value $p$ */
            if $p \leq P_r$ then
                $P_r \leftarrow p$ ;                                 /* store new best Potts value */
                $J_r \leftarrow \ell - 1$ ;                          /* update rightmost jump location */
            end
        end
    end
    /* Reconstruct the minimizer $h$ from the optimal jump locations */
    $r \leftarrow n$; $\ell \leftarrow J_r$;
    while $r > 0$ do
        for $i \leftarrow l + 1$ to $r$ do
            $h_i \leftarrow \mu_{[\ell+1,r]}$;                       /* set mean value on partition */
        end
        $r \leftarrow \ell$; $\ell \leftarrow J_r$;                  /* go to next jump */
    end
end
```

4.3. A new acceleration. The dynamic program we have just described runs through all possible pairings of right and left interval bounds $r = 1, \dots, n$ and $\ell = 1, \dots, r$; cf. Algorithm 2. This amounts to $n(n+1)/2$ iterations. We give a condition which allows us to skip certain iterations. In practice, using this condition, many of the iterations can be omitted, as we will see next. In the next theorem we use the notation of Algorithm 2.

Theorem 2. *Let $r \in \{1, \dots, n\}$. If $P_r < d_{[k,r]} + \gamma$ for some $k = 2, \dots, r$, then, for this r, the inner iterations $\ell = 2, \dots, k-1$ of Algorithm 2 can be skipped.*

Proof. Assume that $P_r < d_{[k,r]} + \gamma$ for some $k \in \{2, \dots, r\}$. From Algorithm 2 we see that the condition for introducing a new jump at location ℓ is

$$P_{\ell-1} + \gamma + d_{[\ell,r]} \leq P_r. \tag{4.7}$$

We show that the condition cannot be fulfilled for $\ell \in \{2, \dots, k-1\}$. To this end, we notice that the mapping $\ell \mapsto d_{[\ell,r]}$ is monotonically decreasing, i.e., $d_{[\ell,r]} \geq d_{[i,r]}$ for all $\ell \leq i$. Further, we observe that $P_\ell \geq 0$ for all $\ell = 1, \dots, r$. Hence we obtain

$$P_r < \gamma + d_{[k,r]} \leq \gamma + d_{[\ell,r]} \leq P_{\ell-1} + \gamma + d_{[\ell,r]} \quad \text{for all } \ 2 \leq \ell \leq k.$$

Hence, condition (4.7) cannot be met. Therefore, we can skip the iterations $\ell = 2, \dots, k-1$ in this case. ■

The accelerated dynamic program for the univariate Potts problem is outlined in Algorithm 2. The speed-up we achieve is especially large for small jump penalties γ since the deviation from the mean then exceeds the optimal Potts value relatively early. Regarding the signals and images of this paper, we observed a speed-up by a factor of 4 to 5 using this strategy.

5. Numerical results.

In all experiments we start the iteration with the coupling parameter $\mu = 0.01\,\gamma$, which we increment by the factor $\tau = 2$ in each step. The stopping criterion is $\|u - v\|_2^2 \leq \mathrm{TOL} \cdot \|\mathrm{f}\|_2^2$ with the tolerance $\mathrm{TOL} = 10^{-10}$. The model parameter γ is chosen empirically. In the denoising experiments, we have run the experiment for different values of γ in steps of 0.05 and picked the one with the highest peak signal-to-noise ratio (PSNR). In the image segmentation experiments, we have chosen a relatively large value for γ so that the results consist of relatively few large segments. Except for Figure 4, we use in all experiments the near-isotropic discretization of jump-penalty given by (3.1). The experiments were conducted on a single core of an Intel Xeon with 3.33 GHz and 16 GB RAM. The original images were taken from the Kodak Lossless True Color Image Suite (Figure 1), from 123RF.com (Figure 2), from the Hyperspectral Images of Natural Scenes 2004 [27, 28] (Figure 5), from the Berkeley Segmentation Dataset [47] (Figures 3, 4, 6), and from Wikipedia (Figure 7) with permission of the Board of Trustees of the National Gallery of Art in Washington, DC. The (noise-free) images are scaled to take on values in the cube $[0,1]^s$. Our MATLAB/Java code is provided at http://pottslab.de.

5.1. Complexity and runtime.

The time-critical parts of the proposed Potts ADMM iteration (Algorithm 1) are the solutions of the univariate Potts problem, which appear in the first and the second steps of (2.6). There, we have to solve n and m univariate Potts problems of length m and n, respectively, each of which amounts to quadratic runtime and linear space. More precisely, each step of the algorithm needs $O(n^2 ms)$ and $O(nm^2 s)$ time and $O(nms)$ space. Note that the required memory is linear in the storage cost of the image $f \in \mathbb{R}^{m \times n \times s}$. For $n \sim m$, we have $O(n^3 s)$ time and $O(n^2 s)$ space complexity in each step of the iteration. Expressing the complexity in dependence of the number of pixels $N = n^2$, we have $O(N^{\frac{3}{2}} s)$ time and $O(Ns)$ space complexity. Typically, we require 20–40 iterations in our experiments. In practice, the runtimes of the proposed method are on the order of one minute for medium-size color images (of about 512 × 512 pixels).

There is room for acceleration, which is outside the scope of this work. Parallelization seems very promising, which we expect to bring a significant speed-up. Theoretically, due to the separable structure of our ADMM strategy, we can achieve $O(Ns)$ time complexity in each iteration using n parallel processors.

5.2. Comparison to related approaches.

We compare our method with two state-of-the-art approaches to the Potts problem whose implementations are publicly available.

The first one is the α-expansion graph cut algorithm based on max-flow/min-cut of the library `GCOptimization 3.0` of Veksler and Delong [9, 8, 41]. Here we used the neighborhood weights of (3.1) and $8 \times 8 \times 8$ discrete labels. The second state-of-the-art method is the convex relaxation method of Pock et al. [57]. Here we used 1000 iterations and only $4 \times 4 \times 4$ labels in order to achieve reasonable runtimes. We further compare our method with total variation minimization using the split Bregman method [34] realized in the toolbox `tvreg` by Getreuer [32]. We used a stopping tolerance of 10^{-6} and a maximum of 1000 iterations.

In Figure 6 we segment a noisy natural image. Total variation minimization often does not produce sharp segment boundaries, whereas the results of the Potts minimization strategies do. The three Potts strategies give reasonable segmentation results, although the graph cut and convex relaxation methods are affected by the label space discretization.

In Figure 7, we reconstruct a cartoon image which was corrupted by Gaussian noise. To measure the reconstruction quality, we use the PSNR. The PSNR of a reconstruction u with respect to the noise-free image $\bar{f}$ is given by

$$\mathrm{PSNR}(u) = 10 \log_{10} \left(\frac{m \cdot n \cdot s \cdot (\max_{i,j,k} |\bar{f}_{ijk}|)^2}{\sum_{i,j,k} |\bar{f}_{ijk} - u_{ijk}|^2} \right).$$

Our first observation is that total variation minimization does not completely remove the noise, although we tried to find an optimal parameter. To that end, we sampled the parameter λ in steps of 0.05 and computed the corresponding minimizer. Then we chose that parameter where the corresponding result had the highest PSNR. The results of all three methods for the Potts problem are almost free of noise. Our method gives the highest reconstruction quality among the three Potts strategies.

We give a detailed quantitative comparison of our method with the graph cut method and the convex relaxation method in Table 1. There, we compare the final energy states of the respective solutions u^*, that is, the functional values of (1.2). We observe that our method attains lower energy states than the other methods.

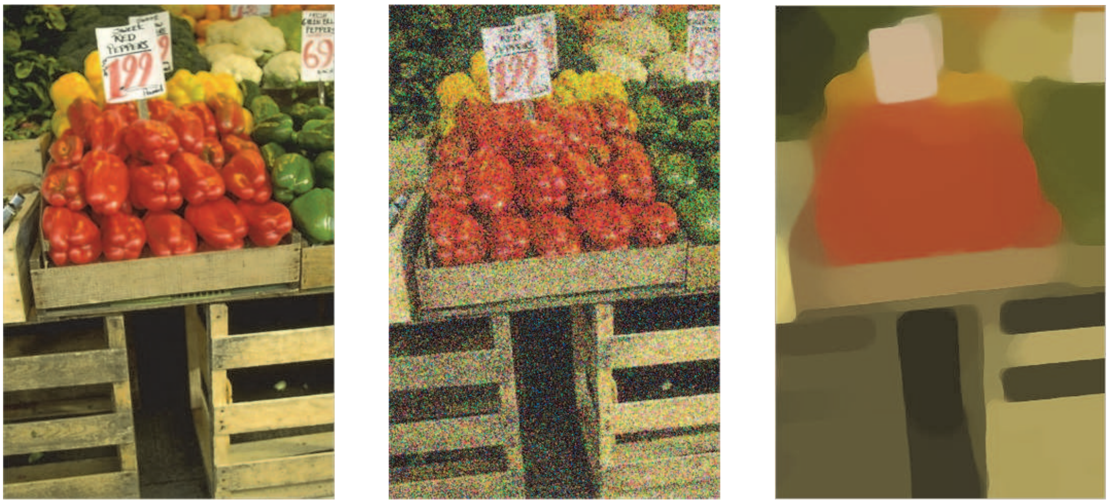
The experiments show that the proposed method is significantly faster than the graph cuts and the convex relaxation approaches.

6. Conclusion.

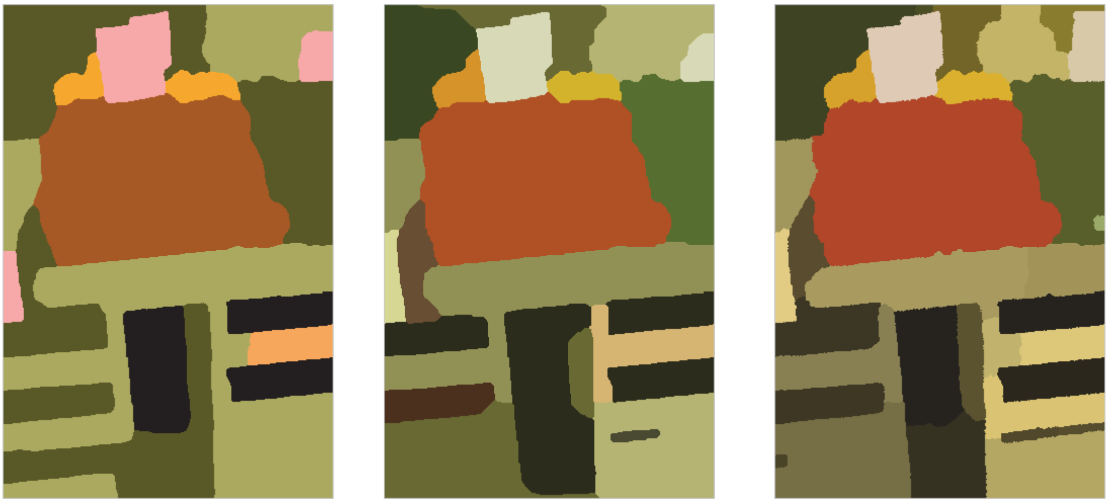
We have proposed a new splitting approach to the Potts problem. We applied our method for image segmentation as well as for the reconstruction of cartoon-like images. We compared it with graph cuts and with a method based on a convex relaxation of the partitioning problem, which are both state-of-the-art methods.

The main benefit of our strategy is its efficiency. Especially for vector-valued images, our method is significantly faster than the graph cuts and the convex relaxation approaches. At the same time, our approach has reconstruction quality as good as these state-of-the-art methods. Furthermore, there is no need for an a priori selection of discrete labels.

The key idea of our strategy was to split the Potts problem into coupled, computationally tractable subproblems using the alternating direction method of multipliers (ADMM). We solved these subproblems using an acceleration of a classical dynamic program. In our experiments, the acceleration gave a gain of factor five compared to the classical program.



(a) Original (481 × 321 pixel) (b) Gaussian noise ($\sigma = 0.3$) (c) Total variation [32] (42.5 sec)

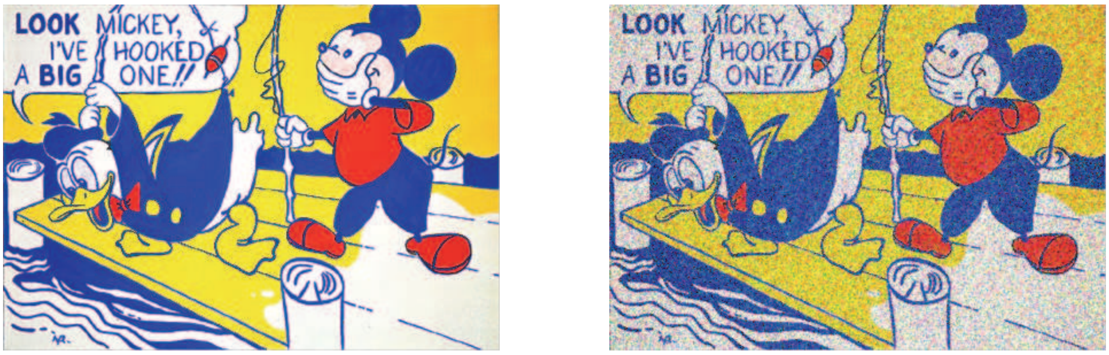


(d) Convex relaxation [57], 4^3 discrete labels (9910.7 sec) (e) Graph cut [9, 8, 41], 8^3 discrete labels (173.1 sec) (f) Our method (31.0 sec)

Figure 6. *Comparison of segmentations of a natural image ($\gamma = 2.0$). Total variation minimization (model parameter $\lambda = 4.0$) tends to smooth out some of the segment boundaries* (c)*. Among the Potts segmentation approaches* (d)–(f)*, the proposed method is the fastest. [Original image credit:* http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/*.]*

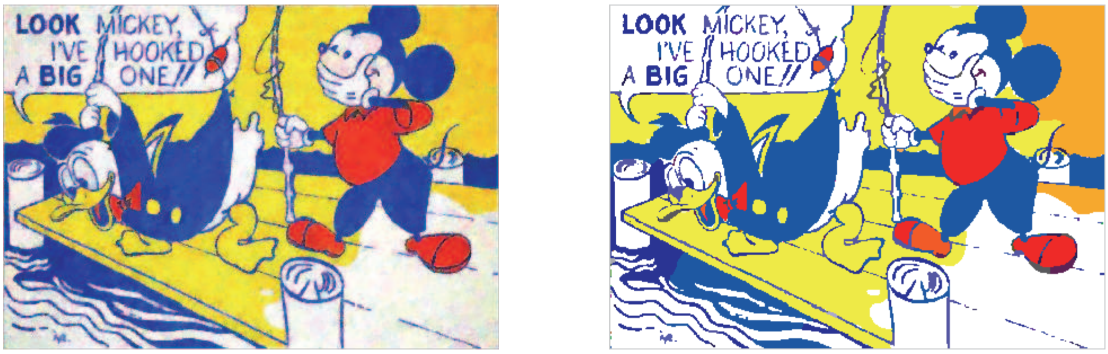
While the linear memory complexity of our algorithm is already optimal, there is still a large potential for runtime reductions by parallelization, to which our approach is particularly suited. Furthermore, the extension to Blake–Zisserman/Mumford–Shah penalties as well as the inclusion of linear measurements are work in progress.

Appendix A. Proof of Theorem 1. We show the statement of Theorem 1 under the more


LOOK MICKEY, I'VE HOOKED A BIG ONE!!
LOOK MICKEY, I'VE HOOKED A BIG ONE!!

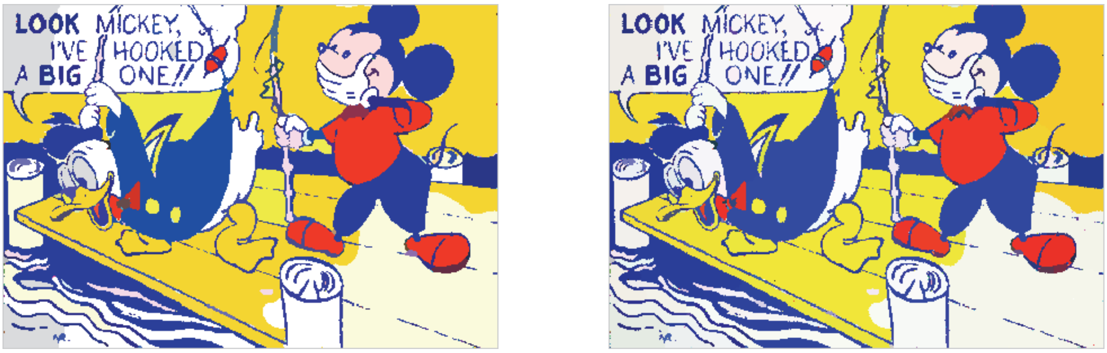
(a) Original (432 × 300 pixel).

(b) Gaussian noise, $\sigma = 0.3$ (PSNR: 10.5).


LOOK MICKEY, I'VE HOOKED A BIG ONE!!
LOOK MICKEY, I'VE HOOKED A BIG ONE!!

(c) Total variation [32] (PSNR: 18.9; 4.0 sec).

(d) Convex relax. [57], 4^3 discrete labels (PSNR: 17.9; 9240.0 sec).


LOOK MICKEY, I'VE HOOKED A BIG ONE!!
LOOK MICKEY, I'VE HOOKED A BIG ONE!!

(e) Graph cut [9, 8, 41], 8^3 discrete labels (PSNR: 19.2; 131.5 sec).

(f) Our method (PSNR: 19.6; 19.4 sec).

Figure 7. *Denoising a cartoon image (*$\gamma = 0.75$*). Total variation minimization (*$\lambda = 0.35$*) does not optimally recover the piecewise constant regions of the image. The label space discretization required by graph cuts and convex relaxation negatively affects the reconstructions. The proposed method gives the best recovery result both with respect to PSNR and visual impression. [“Look Mickey,” Roy Lichtenstein,* 1961. *© Board of Trustees, National Gallery of Art, Washington, DC]*

Table 1

Our method achieves lower energies than the convex relaxation method (CR) of [57] *using* 4^3 *labels and the graph cut method (GC) of* [9, 8, 41] *using* 8^3 *discrete labels. At the same time it is significantly faster. (The application of the graph cut method and the convex relaxation method to the multispectral image exceeded the available memory.)*

		Energy of solution			**Runtime in seconds**		
		CR	**GC**	**Ours**	**CR**	**GC**	**Ours**
$\gamma = 0.5$	Caps	17143.3	8805.2	**7210.5**	22412.9	321.7	**97.3**
	Genie	6578.3	5551.4	**5280.3**	7825.0	74.5	**23.9**
	Desert	9363.2	5540.4	**5044.2**	8842.8	91.3	**25.8**
	Church	7448.9	4042.8	**3722.7**	8806.6	63.6	**24.0**
	Peppers	9856.7	7945.5	**7305.8**	8930.4	88.3	**25.6**
	Mickey	12963.6	11486.1	**11305.1**	8198.5	85.6	**16.2**
	Multispectral	–	–	**6232.0**	–	–	**98.7**
$\gamma = 2.0$	Caps	23965.8	17491.5	**15796.1**	22129.5	538.8	**89.2**
	Genie	17508.2	16152.9	**16112.7**	8513.6	153.6	**25.8**
	Desert	12594.8	9172.7	**8803.5**	8915.7	161.2	**20.5**
	Church	11762.7	8665.6	**8225.9**	8973.2	126.8	**20.3**
	Peppers	16553.7	15106.3	**14305.8**	9031.3	167.3	**27.1**
	Mickey	31602.5	29025.5	**28580.3**	9008.6	95.7	**17.6**
	Multispectral	–	–	**11292.1**	–	–	**120.4**

general assumption that $\{\mu_k\}_k$ is a monotonically increasing sequence fulfilling $\sum_{k=0}^{\infty} \mu_k^{-1/2} < \infty$, which is clearly fulfilled by the actually used geometric progression $\mu_k = \mu_0 \tau^k$.

Proof. Our first goal is to show that the sequence $\{v^k\}_k$ is bounded. Since the sequence member u^{k+1} is a minimizer of the first line in (2.6), we get the estimate

$$\frac{2\gamma}{1+\mu_k}\|\nabla_1 u^{k+1}\|_0 + \left\| u^{k+1} - \frac{f+\mu_k v^k - \lambda^k}{1+\mu_k} \right\|_2^2 \\ \leq \frac{2\gamma}{1+\mu_k}\left\| \nabla_1 \frac{f+\mu_k v^k - \lambda^k}{1+\mu_k} \right\|_0 \leq \frac{2\gamma mn}{1+\mu_k}.$$

As a consequence we obtain that

$$\left\| u^{k+1} - \frac{f+\mu_k v^k - \lambda^k}{1+\mu_k} \right\|_2 \leq \frac{C}{\sqrt{1+\mu_k}}. \tag{A.1}$$

Here the positive constant C is given by $C = \sqrt{2\gamma mn}$, where γ is the regularization parameter and n, m are determined by the size of the image. Likewise, the second line of (2.6) yields

$$\left\| v^{k+1} - \frac{f+\mu_k u^{k+1} + \lambda^k}{1+\mu_k} \right\|_2 \leq \frac{C}{\sqrt{1+\mu_k}}. \tag{A.2}$$

Summing up the terms within the norm symbols in (A.1) and (A.2), and afterward using the triangle inequality, we get that

$$\left\| u^{k+1} + v^{k+1} - \frac{2f + \mu_k v^k + \mu_k u^{k+1}}{1+\mu_k} \right\|_2 \le \frac{2C}{\sqrt{1+\mu_k}}.$$

Rewriting this inequality yields

$$\left\| \frac{u^{k+1} + v^{k+1} - 2f}{1+\mu_k} - \frac{\mu_k}{1+\mu_k}(v^k - v^{k+1}) \right\|_2 \le \frac{2C}{\sqrt{1+\mu_k}}. \tag{A.3}$$

Likewise, we subtract the terms within the norm symbols in (A.1) from that in (A.2) and use the triangle inequality in the form $\|a-b\|_2 \le \|a\|_2 + \|b\|_2$ to obtain

$$\left\| u^{k+1} - v^{k+1} - \frac{\mu_k}{1+\mu_k} v^k + \frac{\mu_k}{1+\mu_k} u^{k+1} + \frac{2\lambda^k}{1+\mu_k} \right\|_2 \le \frac{2C}{\sqrt{1+\mu_k}}.$$

We substitute $\lambda^k + \mu_k(u^{k+1} - v^{k+1}) = \lambda^{k+1}$, which is the third line of (2.6), to get

$$\left\| \frac{2\lambda^{k+1}}{1+\mu_k} + \frac{u^{k+1} - v^{k+1}}{1+\mu_k} - \frac{\mu_k}{1+\mu_k}(v^k - v^{k+1}) \right\|_2 \le \frac{2C}{\sqrt{1+\mu_k}}. \tag{A.4}$$

The inequalities (A.3) and (A.4) are of the form $\|x-y\| \le a$ and $\|z-y\| \le a$, respectively. Hence, $\|x-z\| \le 2a$, which in our case means

$$\left\| \frac{2\lambda^{k+1}}{1+\mu_k} + \frac{u^{k+1} - v^{k+1}}{1+\mu_k} - \frac{u^{k+1} + v^{k+1} - 2f}{1+\mu_k} \right\|_2 \le \frac{4C}{\sqrt{1+\mu_k}}.$$

Simplifying the left-hand side of this inequality yields

$$\left\| \frac{\lambda^{k+1}}{1+\mu_k} - \frac{v^{k+1} - f}{1+\mu_k} \right\|_2 \le \frac{2C}{\sqrt{1+\mu_k}}. \tag{A.5}$$

Equipped with these estimates, we are now going to bound the distance between u^{k+2} and a convex combination of v^{k+1} and f. We first use (A.1) with k replaced by $k+1$ and get

$$\begin{aligned} &\left\| u^{k+2} - \frac{(\mu_{k+1} - 1)v^{k+1} + 2f}{1+\mu_{k+1}} \right\|_2 \\ &= \left\| u^{k+2} - \frac{f + \mu_{k+1} v^{k+1}}{1+\mu_{k+1}} + \frac{\lambda^{k+1}}{1+\mu_{k+1}} - \frac{\lambda^{k+1}}{1+\mu_{k+1}} + \frac{v^{k+1} - f}{1+\mu_{k+1}} \right\|_2 \\ &\le \frac{C}{\sqrt{1+\mu_{k+1}}} + \left\| \frac{v^{k+1} - f}{1+\mu_{k+1}} - \frac{\lambda^{k+1}}{1+\mu_{k+1}} \right\|_2 . \end{aligned}$$

Now we apply (A.5) to get that

$$\begin{aligned} &\left\| u^{k+2} - \frac{(\mu_{k+1} - 1)v^{k+1} + 2f}{1+\mu_{k+1}} \right\|_2 \\ &\le \frac{C}{\sqrt{1+\mu_{k+1}}} + \frac{1+\mu_k}{1+\mu_{k+1}} \left\| \frac{v^{k+1} - f}{1+\mu_k} - \frac{\lambda^{k+1}}{1+\mu_k} \right\|_2 \\ &\le \left(1 + 2\frac{\sqrt{1+\mu_k}}{\sqrt{1+\mu_{k+1}}} \right) \frac{C}{\sqrt{1+\mu_{k+1}}} \le \frac{3C}{\sqrt{1+\mu_{k+1}}}. \end{aligned} \tag{A.6}$$

The last inequality is a consequence of the monotonicity of the sequence μ_k. Likewise, we estimate using (A.2) with k replaced by $k+1$ and (A.5)

$$\begin{aligned}
&\left\| v^{k+2} - \frac{\mu_{k+1}u^{k+2} + v^{k+1}}{1+\mu_{k+1}} \right\|_2 \\
&\leq \left\| v^{k+2} - \frac{f + \mu_{k+1}u^{k+2} + \lambda^{k+1}}{1+\mu_{k+1}} \right\|_2 + \left\| \frac{v^{k+1} - f - \lambda^{k+1}}{1+\mu_{k+1}} \right\|_2 \\
&\leq \frac{C}{\sqrt{1+\mu_{k+1}}} + \frac{1+\mu_k}{1+\mu_{k+1}} \left\| \frac{v^{k+1} - f - \lambda^{k+1}}{1+\mu_k} \right\|_2 \leq \frac{3C}{\sqrt{1+\mu_{k+1}}}.
\end{aligned} \tag{A.7}$$

Next, we combine (A.6) and (A.7) to get

$$\begin{aligned}
&\left\| v^{k+2} - \frac{\mu_{k+1}^2 + 1}{(\mu_{k+1}+1)^2} v^{k+1} - \frac{2\mu_{k+1}}{(1+\mu_{k+1})^2} f \right\|_2 \\
&= \left\| v^{k+2} - \frac{\mu_{k+1}(\mu_{k+1}-1) + \mu_{k+1} + 1}{(\mu_{k+1}+1)^2} v^{k+1} - \frac{2\mu_{k+1}}{(1+\mu_{k+1})^2} f \right\|_2 \\
&= \left\| v^{k+2} - \frac{\mu_{k+1}}{1+\mu_{k+1}} \frac{(\mu_{k+1}-1)v^{k+1} + 2f}{1+\mu_{k+1}} - \frac{v^{k+1}}{1+\mu_{k+1}} + \frac{\mu_{k+1}u^{k+2}}{1+\mu_{k+1}} - \frac{\mu_{k+1}u^{k+2}}{1+\mu_{k+1}} \right\|_2 \\
&\leq \frac{\mu_{k+1}}{1+\mu_{k+1}} \left\| u^{k+2} - \frac{(\mu_{k+1}-1)v^{k+1} + 2f}{1+\mu_{k+1}} \right\|_2 + \left\| v^{k+2} - \frac{\mu_{k+1}u^{k+2}}{1+\mu_{k+1}} - \frac{v^{k+1}}{1+\mu_{k+1}} \right\|_2 \\
&\leq \frac{6C}{\sqrt{1+\mu_{k+1}}}.
\end{aligned} \tag{A.8}$$

Using the last estimate, we obtain that

$$\begin{aligned}
\|v^{k+2}\|_2 &\leq \|v^{k+2} - d_k v^{k+1} - (1-d_k)f\|_2 + \|d_k v^{k+1} + (1-d_k)f\|_2 \\
&\leq \frac{6C}{\sqrt{1+\mu_{k+1}}} + \|v^{k+1}\|_2 + \frac{2\mu_{k+1}}{(1+\mu_{k+1})^2}\|f\|_2,
\end{aligned}$$

where $d_k = \frac{\mu_{k+1}^2+1}{(\mu_{k+1}+1)^2}$. Resolving the recursion in v^k, we have

$$\|v^{k+2}\|_2 \leq 6C \sum_{i=0}^{k} \frac{1}{\sqrt{1+\mu_{i+1}}} + 2\sum_{i=0}^{k} \frac{1}{1+\mu_{i+1}} \|f\|_2 + \|v^1\|_2.$$

By assumption $\sum_{i=0}^{\infty} \mu_i^{-1/2} < \infty$, which implies that both sums above are bounded. Hence, $\{v^k\}_k$ is bounded.

Using (A.6), we get that the sequence $\{u^k\}_k$ is bounded as well. It follows from (A.3) that

$$\|v^{k+1} - v^k\|_2 \to 0.$$

Using the monotonicity of the sequence μ_k (A.4) implies

$$\frac{\lambda^{k+1}}{1+\mu_{k+1}} \to 0.$$

From (A.1) and (A.2) we conclude that

$$\|u^{k+1} - v^k\|_2 \to 0 \quad \text{and} \quad \|u^{k+1} - v^{k+1}\|_2 \to 0.$$

Therefore, it is sufficient to show that $\{v^k\}_k$ is convergent. To this end, we estimate using (A.8)

$$\begin{aligned}\|v^{k+\ell} - v^k\|_2 &\leq \sum_{s=k}^{\ell-1} \|v^{s+1} - v^s\|_2 \\ &= \sum_{s=k}^{\ell-1} \|v^{s+1} - d_{s-1}v^s - (1-d_{s-1})f + (1-d_{s-1})(f - v^s)\|_2 \\ &\leq \sum_{s=k}^{\ell-1} \|v^{s+1} - d_{s-1}v^s - (1-d_{s-1})f\|_2 + \sum_{s=k}^{\ell-1}(1-d_{s-1})\|f - v^s\|_2 \\ &\leq C' \cdot \sum_{s=k}^{\ell-1} \frac{1}{\sqrt{1+\mu_s}} + C'' \sum_{s=k}^{\ell-1} \frac{2\mu_s}{(\mu_s+1)^2}\end{aligned}$$

for some constants $C', C'' > 0$ which are independent of k and l. Thus, $\{v^k\}_k$ is a Cauchy sequence and therefore convergent. This completes the proof. ■

Acknowledgment. The first author would like to thank Thomas Pock for a valuable discussion at the workshop Advances in Mathematical Image Processing 2012.

REFERENCES

[1] M. Artina, M. Fornasier, and F. Solombrino, *Linearly constrained nonsmooth and nonconvex minimization*, SIAM J. Optim., 23 (2013), pp. 1904–1937.

[2] E. Bae, J. Yuan, and X.-C. Tai, *Global minimization for continuous multiphase partitioning problems using a dual approach*, Internat. J. Comput. Vision, 92 (2011), pp. 112–129.

[3] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[4] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA, 1987.

[5] T. Blumensath and M. Davies, *Iterative thresholding for sparse approximations*, J. Fourier Anal. Appl., 14 (2008), pp. 629–654.

[6] T. Blumensath and M. Davies, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmon. Anal., 27 (2009), pp. 265–274.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learning, 3 (2011), pp. 1–122.

[8] Y. Boykov and V. Kolmogorov, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*, IEEE Trans. Pattern Anal. Machine Intell., 26 (2004), pp. 1124–1137.

[9] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Trans. Pattern Anal. Machine Intell., 23 (2001), pp. 1222–1239.

[10] E.S. Brown, T.F. Chan, and X. Bresson, *Completely convex formulation of the Chan-Vese image segmentation model*, Internat. J. Comput. Vision, 98 (2012), pp. 103–121.

[11] X. Cai, R. Chan, and T. Zeng, *A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding*, SIAM J. Imaging Sci., 6 (2013), pp. 368–390.

[12] X. Cai and G. Steidl, *Multiclass segmentation by iterated ROF thresholding*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, New York, 2013, pp. 237–250.
[13] E.J. Candès, M.B. Wakin, and S.P. Boyd, *Enhancing sparsity by reweighted ℓ^1 minimization*, J. Fourier Anal. Appl., 14 (2008), pp. 877–905.
[14] A. Chambolle, *Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations*, SIAM J. Appl. Math., 55 (1995), pp. 827–863.
[15] A. Chambolle, *Finite-differences discretizations of the Mumford-Shah functional*, ESAIM Math. Model. Numer. Anal., 33 (1999), pp. 261–288.
[16] A. Chambolle, D. Cremers, and T. Pock, *A convex approach to minimal partitions*, SIAM J. Imaging Sci., 5 (2012), pp. 1113–1158.
[17] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
[18] T.F. Chan, S. Esedoḡlu, and M. Nikolova, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math., 66 (2006), pp. 1632–1648.
[19] T.F. Chan, B.Y. Sandberg, and L.A. Vese, *Active contours without edges for vector-valued images*, J. Visual Commun. Image Represent., 11 (2000), pp. 130–141.
[20] T.F. Chan and L.A. Vese, *Active contours without edges*, IEEE Trans. Image Process., 10 (2001), pp. 266–277.
[21] H.-D. Cheng, X.H. Jiang, Y. Sun, and J. Wang, *Color image segmentation: Advances and prospects*, Pattern Recognition, 34 (2001), pp. 2259–2281.
[22] L. Condat, *A direct algorithm for 1-D total variation denoising*, IEEE Signal Process. Lett., 20 (2013), pp. 1054–1057.
[23] D. Cremers, M. Rousson, and R. Deriche, *A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape*, Internat. J. Comput. Vision, 72 (2007), pp. 195–215.
[24] P. Davies and A. Kovac, *Local extremes, runs, strings and multiresolution*, Ann. Statist., 29 (2001), pp. 1–65.
[25] Y. Deng and B.S. Manjunath, *Unsupervised segmentation of color-texture regions in images and video*, IEEE Trans. Pattern Anal. Machine Intell., 23 (2001), pp. 800–810.
[26] M. Fornasier and R. Ward, *Iterative thresholding meets free-discontinuity problems*, Found. Comput. Math., 10 (2010), pp. 527–567.
[27] D.H. Foster, K. Amano, S.M.C. Nascimento, and M.J. Foster, *Frequency of metamerism in natural scenes*, J. Optical Soc. Amer. A, 23 (2006), pp. 2359–2372; also available online from http://personalpages.manchester.ac.uk/staff/david.foster/Hyperspectral_images_of_natural_scenes_04.html.
[28] D. Foster, A. Nascimento, and K. Amano, *Information limits on neural identification of colored surfaces in natural scenes*, Visual Neurosci., 21 (2004), pp. 331–336.
[29] F. Friedrich, A. Kempe, V. Liebscher, and G. Winkler, *Complexity penalized M-estimation*, J. Comput. Graph. Statist., 17 (2008), pp. 201–224.
[30] K.-S. Fu and J.K. Mui, *A survey on image segmentation*, Pattern Recognition, 13 (1981), pp. 3–16.
[31] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Machine Intell., 6 (1984), pp. 721–741.
[32] P. Getreuer, *Rudin-Osher-Fatemi total variation denoising using split Bregman*, Image Processing On Line, 10 (2012); http://www.getreuer.info/home/tvreg.
[33] B. Goldluecke, E. Strekalovskiy, and D. Cremers, *Tight convex relaxations for vector-valued labeling*, SIAM J. Imaging Sci., 6 (2013), pp. 1626–1664.
[34] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.
[35] D.M. Greig, B.T. Porteous, and A.H. Seheult, *Exact maximum a posteriori estimation for binary images*, J. Roy. Statist. Soc. Ser. B (Methodol.), 51 (1989), pp. 271–279.
[36] P.C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580.
[37] R.M. Haralick and L.G. Shapiro, *Image segmentation techniques*, Comput. Vision Graph. Image Process., 29 (1985), pp. 100–132.
[38] H. Hirschmüller, *Accurate and efficient stereo processing by semi-global matching and mutual information*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2,

IEEE, Washington, DC, 2005, pp. 807–814.
[39] H. Hirschmüller, *Stereo vision in structured environments by consistent semi-global matching*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, IEEE, Washington, DC, 2006, pp. 2386–2393.
[40] H. Hirschmüller, *Stereo processing by semiglobal matching and mutual information*, IEEE Trans. Pattern Anal. Machine Intell., 30 (2008), pp. 328–341.
[41] V. Kolmogorov and R. Zabih, *What energy functions can be minimized via graph cuts?*, IEEE Trans. Pattern Anal. Machine Intell., 26 (2004), pp. 147–159.
[42] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr, *Convex multi-class image labeling by simplex-constrained total variation*, in Scale Space and Variational Methods in Computer Vision, Springer, Berlin, 2009, pp. 150–162.
[43] J. Lellmann, B. Lellmann, F. Widmann, and C. Schnörr, *Discrete and continuous models for partitioning problems*, Internat. J. Comput. Vision, 104 (2013), pp. 241–269.
[44] J. Lellmann and C. Schnörr, *Continuous multiclass labeling approaches and algorithms*, SIAM J. Imaging Sci., 4 (2011), pp. 1049–1096.
[45] Z. Lu, *Iterative hard thresholding methods for l_0 regularized convex cone programming*, Math. Program., 147 (2014), pp. 125–154.
[46] Z. Lu and Y. Zhang, *Sparse approximation via penalty decomposition methods*, SIAM J. Optim., 23 (2013), pp. 2448–2478.
[47] D. Martin, C. Fowlkes, D. Tal, and J. Malik, *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, in Proceedings of the 8th International Conference on Computer Vision, 2001, pp. 416–423; also available online from http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.
[48] V. Morozov, *Methods for Solving Incorrectly Posed Problems*, Springer, New York, 1984.
[49] D. Mumford and J. Shah, *Boundary detection by minimizing functionals*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 17, IEEE, Washington, DC, 1985, pp. 137–154.
[50] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
[51] J.C. Nascimento and J.S. Marques, *Performance evaluation of object detection algorithms for video surveillance*, IEEE Trans. Multimedia, 8 (2006), pp. 761–774.
[52] M.K. Ng, P. Weiss, and X. Yuan, *Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods*, SIAM J. Sci. Comput., 32 (2010), pp. 2710–2736.
[53] M. Nikolova, M.K. Ng, and C.-P. Tam, *Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction*, IEEE Trans. Image Process., 19 (2010), pp. 3073–3088.
[54] M. Nikolova, M.K. Ng, S. Zhang, and W.-K. Ching, *Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization*, SIAM J. Imaging Sci., 1 (2008), pp. 2–25.
[55] N. Parikh and S. Boyd, *Proximal algorithms*, Found. Trends Optim., 1 (2013), pp. 123–231.
[56] D. Pham, C. Xu, and J. Prince, *Current methods in medical image segmentation*, Ann. Rev. Biomed. Engrg., 2 (2000), pp. 315–337.
[57] T. Pock, A. Chambolle, D. Cremers, and H. Bischof, *A convex relaxation approach for computing minimal partitions*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Washington, DC, 2009, pp. 810–817; also available online from http://gpu4vision.icg.tugraz.at.
[58] R.B. Potts, *Some generalized order-disorder transformations*, Math. Proc. Cambridge Philos. Soc., 48 (1952), pp. 106–109.
[59] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[60] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Machine Intell., 22 (2000), pp. 888–905.
[61] G. Steidl and T. Teuber, *Removing multiplicative noise by Douglas-Rachford splitting methods*, J. Math. Imaging Vision, 36 (2010), pp. 168–184.
[62] M. Storath, A. Weinmann, and L. Demaret, *Jump-sparse and sparse recovery using Potts functionals*, IEEE Trans. Signal Process., 62 (2014), pp. 3654–3666.

[63] E. Strekalovskiy, A. Chambolle, and D. Cremers, *A convex representation for the vectorial Mumford-Shah functional*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2012, pp. 1712–1719.

[64] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, *A comparative study of energy minimization methods for Markov random fields with smoothness-based priors*, IEEE Trans. Pattern Anal. Machine Intell., 30 (2008), pp. 1068–1080.

[65] M. Unser, *Texture classification and segmentation using wavelet frames*, IEEE Trans. Image Process., 4 (1995), pp. 1549–1560.

[66] L.A. Vese and T.F. Chan, *A multiphase level set framework for image segmentation using the Mumford and Shah model*, Internat. J. Comput. Vision, 50 (2002), pp. 271–293.

[67] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.

[68] A. Weinmann, M. Storath, and L. Demaret, *The L^1-Potts functional for robust jump-sparse reconstruction*, submitted, preprint, arXiv:1207.4642, 2012.

[69] G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*, Appl. Math. 27, Springer-Verlag, Berlin, 2003.

[70] G. Winkler and V. Liebscher, *Smoothers for discontinuous signals*, J. Nonparametric Statist., 14 (2002), pp. 203–222.

[71] G. Winkler, O. Wittich, V. Liebscher, and A. Kempe, *Don't shed tears over breaks*, Jahresber. Deutsch. Math.-Verein., 107 (2005), pp. 57–87.

[72] L. Xu, C. Lu, Y. Xu, and J. Jia, *Image smoothing via l_0 gradient minimization*, ACM Trans. Graphics, 30 (2011), p. 174.

[73] J. Yang and Y. Zhang, *Alternating direction algorithms for ℓ_1-problems in compressive sensing*, SIAM J. Sci. Comput., 33 (2011), pp. 250–278.

[74] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer, *Fast global labeling for real-time stereo using multiple plane sweeps*, in Vision, Modeling, and Visualization Workshop, Konstanz, Germany, 2008, pp. 243–252.

[75] Y. Zhang, B. Dong, and Z. Lu, *l_0 minimization for wavelet frame based image restoration*, Math. Comp., 82 (2013), pp. 995–1015.