

THE L^1 -POTTS FUNCTIONAL FOR ROBUST JUMP-SPARSE RECONSTRUCTION*

ANDREAS WEINMANN[†], MARTIN STORATH[‡], AND LAURENT DEMARET[†]

Abstract. We investigate the nonsmooth and nonconvex L^1 -Potts functional in discrete and continuous time. We show Γ -convergence of discrete L^1 -Potts functionals toward their continuous counterpart and obtain a convergence statement for the corresponding minimizers as the discretization gets finer. For the discrete L^1 -Potts problem, we introduce an $O(n^2)$ time and $O(n)$ space algorithm to compute an exact minimizer. We apply L^1 -Potts minimization to the problem of recovering piecewise constant signals from noisy measurements f . It turns out that the L^1 -Potts functional has a quite interesting blind deconvolution property. In fact, we show that mildly blurred jump-sparse signals are reconstructed by minimizing the L^1 -Potts functional. Furthermore, for strongly blurred signals and a known blurring operator, we derive an iterative reconstruction algorithm.

Key words. Potts functional, jump-sparse reconstruction, nonconvex functional, penalized absolute deviation, adaptive estimation

AMS subject classifications. 65K05, 65K10, 65Y20, 49J52, 90C39, 93E14, 62G08

DOI. 10.1137/120896256

1. Introduction. In this article, we study the L^1 -Potts functional

$$(1.1) \quad P_\gamma(u) = \gamma \cdot \|\nabla u\|_0 + \|u - f\|_1 = \gamma \cdot J(u) + \sum_i |u_i - f_i|$$

and its continuous time counterpart. Here, the regularity term $J(u) = \|\nabla u\|_0$ counts the number of jumps of u . (The abusive but appealing norm notation $\|\cdot\|_0$ has been popularized by the sparsity community [14].) A minimizer of the L^1 -Potts functional may be seen as a *jump-sparse* approximation to the data f .

The investigation of L^2 -Potts functionals (with the L^1 norm in (1.1) replaced by $\|\cdot\|_2^2$) can be traced back to Potts [25] in 1952 and even to Ising [20] in 1925 in a physical context. S. Geman and D. Geman [18] were the first to consider such functionals in the context of image processing. For a thorough account on the history, we refer to [28] and references therein. More recent results can be found in [29, 4].

Our investigation of L^1 -Potts functionals is motivated by a signal processing problem, namely, the reconstruction of piecewise constant signals from noisy and blurred spatial measurements f . Recovery of piecewise constant signals from corrupted data is a task needed in various fields of applied sciences. Examples are the reconstruction of brain stimuli [28], the cross-hybridization of DNA [13, 19], and MALDI imaging [26]. The data may be incomplete frequency information as in [7, 22] or noisy and

*Received by the editors October 23, 2012; accepted for publication (in revised form) December 9, 2014; published electronically February 24, 2015. The research of the authors was supported by the German Federal Ministry for Education and Research under SysTec grant 03I5508.

<http://www.siam.org/journals/sinum/53-1/89625.html>

[†]Department of Mathematics, Helmholtz Center Munich, Technische Universität München, Munich 85748, Germany (andreas.weinmann@tum.de, laurent.demaret@helmholtz-muenchen.de). The research of the first author was supported by the Helmholtz Association within the young investigator group VH-NG-526.

[‡]Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (martin.storath@epfl.ch). The research of the second author was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement 267439.

blurred measurements as in this paper. Frequently, the type of noise is unknown or known to be non-Gaussian.

The above reconstruction task is an inverse problem which is typically approached by minimization of an energy functional reflecting a tradeoff between data fidelity and regularity. In this regard, classical Tykhonov approaches favor oversmoothed solutions and are not robust to non-Gaussian noise. To overcome these limitations, various functionals have been proposed; cf. [24]. In particular, L^1 -TV and related minimization problems have gained a lot of interest in recent years, see, e.g., [10, 9, 17, 30] and references therein. Under the prior assumption of jump-sparse signals, the number of jumps of u is a more natural penalty than the total variation $\|\nabla u\|_1$. This directly leads to the minimization of the L^1 -Potts functional (1.1). The numerical experiments in this paper indicate that L^1 -Potts minimization outperforms L^1 -TV minimization whenever the underlying signal is jump-sparse. Furthermore, the L^1 -Potts functional is more robust to non-Gaussian noise than the L^2 -Potts functional and yields comparable results for Gaussian noise.

The main contributions of this article are (i) development of a discretization framework for the continuous L^1 -Potts functional, (ii) development of a fast algorithm to minimize the discretized L^1 -Potts functionals, and (iii) proof that both continuous and discretized L^1 -Potts functionals have blind deconvolution properties.

As for (i), we consider (possibly nonequidistant) samplings f_k of continuous time data f . We show that the discrete L^1 Potts functionals associated with f_k Γ -converge to the continuous L^1 -Potts functional as the sampling density gets finer. Furthermore, we show that each sequence of minimizers of the discrete functionals has an accumulation point and that each such accumulation point is a minimizer of the continuous model. The precise formulation of these statements is Theorem 2.9, which is the main result of section 2. We in particular show that the continuous Potts functional has a minimizer. (This contrasts the situation for general inverse Potts functionals, where the continuous model need not have minimizers, but the discrete models do; cf. [27].) Γ -convergence and convergence of minimizers in relation with the L^2 -Potts functional are topics of [4, 5]. In these papers, Γ -convergence is shown for a modified L^2 -Potts functional. This approach relies on the Hilbert space structure of L^2 and therefore does not directly carry over to the L^1 context.

Concerning (ii), we notice that the nonconvexity of the jump penalty in (1.1) makes standard techniques for convex optimization as used for L^1 -TV minimization inaccessible. Instead, dynamic programming is used [23, 8]. A fast algorithm for L^2 -Potts minimization has been introduced by Friedrich et al. and can be found in [16]. This algorithm computes an exact minimizer in $O(n^2)$ time and $O(n)$ space, where n denotes the length of the discrete data. In their paper, they also consider the discrete L^1 -Potts model. Using a red-black tree approach, they derive an $O(n^2 \log n)$ time algorithm to minimize (1.1), which improves the time complexity of a naive implementation by a factor of n . However, this comes with $O(n^2)$ space consumption, thus losing a factor of n against the naive implementation. Liebscher [21] posed the question whether it is possible to minimize the L^1 -Potts functional as fast as the classical L^2 version. In this paper, we give an affirmative answer. We show that the L^1 -Potts algorithm proposed computes an exact minimizer within $O(n^2)$ time and $O(n)$ space. This statement is formulated as Theorem 3.3, which, more generally, treats the case of not necessarily equal weights. Such weights arise naturally in applications where data are available only on a nonuniform grid [26]. The key to the speed-up is to work with dynamic data structures tailored to the problem. In addition to this asymptotic result, the actual runtime of the L^1 algorithm is only less

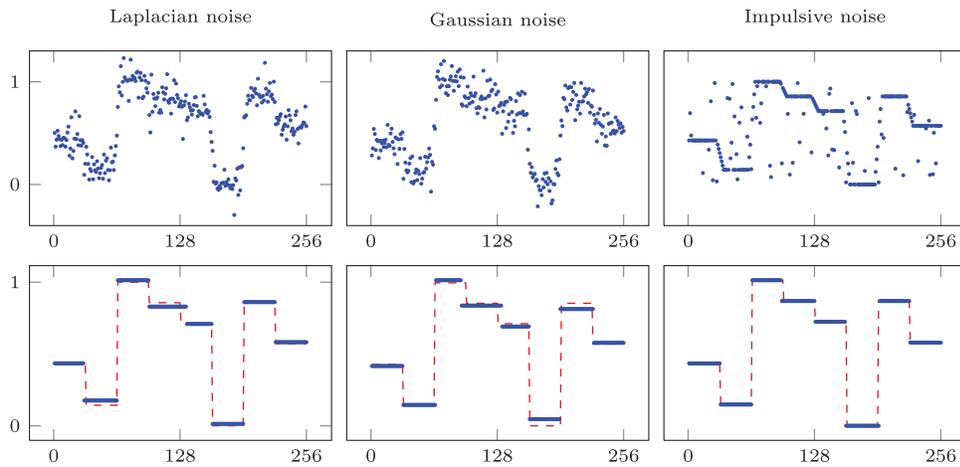


FIG. 1. *Top row: A piecewise constant signal blurred by the moving average of width 7 and corrupted by Laplacian noise ($\sigma = 0.1$), Gaussian noise ($\sigma = 0.1$) or impulsive noise (with 25% of the data destroyed). Bottom row: The original signal (dashed line) is almost perfectly recovered by the respective minimizers of the L^1 -Potts functional (1.1) for all three types of noise.*

than 20% slower than that of the L^2 version for a larger data size. The algorithm and our results as well as numerical experiments comparing the L^1 -Potts algorithm with the L^2 -Potts algorithm of [16] and the L^1 -TV algorithm of [10] under various types of noise are presented in section 3. An efficient implementation is available for download at <http://pottslab.de>.

As for (iii), we observe that in addition to its high robustness to noise, the L^1 -Potts functional has blind deconvolution capabilities which are not shared by the L^2 -Potts and the L^1 -TV functional; cf. Figure 7. In fact, we show that the continuous time L^1 -Potts functional is able to exactly recover piecewise constant signals g from mildly blurred measurements $K * g$ without knowledge of the (narrowly supported) convolution kernel K . Further, this property is inherited by the discretizations of the L^1 -Potts functional. The precise statements are formulated as Theorems 4.4 and 4.5. Experiments indicate that the deconvolution property persists also in the presence of various types of noise; see Figure 1. Theoretical results and experiments in connection with blind deconvolution are the topic of section 4.

Finally, for the reconstruction of severely blurred signals (and known kernel), we consider the related deconvolution problem

$$(1.2) \quad \gamma \cdot \|\nabla u\|_0 + \|K * u - f\|_1 \rightarrow \min.$$

Theoretical properties of the continuous L^2 variant of (1.2) have been investigated in [3]. However, to our knowledge, no algorithms to solve either (1.2) or the corresponding L^2 -variant have been proposed yet. In section 5 we derive a heuristic to solve (1.2), where our fast algorithm to solve (1.1) is employed as a basic building block of an iterative algorithm; for results, see Figure 9.

2. Continuous Potts model and Γ -convergence. In this section, we consider the continuous L^1 -Potts functional defined on $L^1[0, 1]$ by

$$(2.1) \quad P_\gamma(u) = \gamma \cdot J(u) + \|u - f\|_1 \quad \text{if } u \in \text{PC}[0, 1]$$

and by $P_\gamma(u) = \infty$ otherwise. Here, $\text{PC}[0, 1]$ denotes the space of piecewise constant functions on the interval $[0, 1]$. We identify a piecewise constant function with its equivalence class (under identification of absolutely integrable functions which are equal almost everywhere with respect to (w.r.t.) Lebesgue measure) in order to view it as an element of $L^1[0, 1]$. The resulting subspace of piecewise constant functions equals precisely those elements in $L^1[0, 1]$ whose distributional derivatives are (finite) linear combinations of Dirac distributions centered in $(0, 1)$. In view of this equivalent definition, one may also define the symbol $J(u) = \|\nabla u\|_0$ as the number of points in the support of the distributional derivative of the element $u \in L^1[0, 1]$.

We start with some basics on L^1 -Potts functionals and emphasize the nonuniqueness of minimizers. Then we have a look at discretizations of the L^1 Potts functional. Our final goal is to obtain convergence statements of the discretizations to the continuous model. As an intermediate step, we investigate the convergence of semidiscrete Potts functionals. In order to define these functionals, we consider a nested sequence $X^k \subset X^{k+1}$ of finite sets in $[0, 1]$ which we call *discretization sets*. We assume that the union of the nested sets X^k is dense in $[0, 1]$. (The number n_k of elements in the k th level discretization X^k is not required to equal k , but we assume that n_k is strictly increasing.) We use the notation $\text{PC}^k[0, 1]$ to denote the space of those piecewise constant functions which only jump in the k th level discretization set X^k . Then the sequence of semidiscrete Potts functionals (Q_γ^k) (associated with the discretizations X^k) is given by

$$(2.2) \quad Q_\gamma^k(u) = \gamma \cdot J(u) + \|u - f\|_1 \quad \text{if } u \in \text{PC}^k[0, 1]$$

and by $Q_\gamma^k(u) = \infty$ otherwise. The auxiliary functionals (Q_γ^k) are still defined by data f defined on the continuum $[0, 1]$. However, in practice, data are given by discrete samples $f^k = S_k f$ obtained from f by a sampling operator S_k at some level k . One implementation of S_k might be the integral sampling

$$(2.3) \quad f^k(j) = S_k f(j) = \frac{1}{|I_j|} \int_{I_j} f \, d\lambda,$$

where I_j are the intervals given by the discretization sets X^k . If f is continuous, one might also consider point sampling

$$(2.4) \quad f^k(j) = S_k f(j) = f(x_j),$$

where x_j might be taken as the midpoint of the interval I_j . Then, the discrete Potts functionals read

$$(2.5) \quad P_\gamma^k(u) = \gamma \cdot J(u) + \|u - \sum_j S_k f(j) \cdot \mathbf{1}_{I_j}\|_1 \quad \text{if } u \in \text{PC}^k[0, 1],$$

and $P_\gamma^k(u) = \infty$ otherwise.

In subsection 2.2 we obtain convergence statements (Theorem 2.7) for these semidiscrete Potts functionals, which we then apply in subsection 2.3 to obtain the corresponding statements for the (fully) discretized L^1 Potts functionals we are ultimately interested in. The main result is Theorem 2.9.

2.1. Basics on L^1 -Potts functionals. It is well known that a median μ minimizes the L^1 error for a constant approximation, that is,

$$(2.6) \quad \|g - \mu\|_1 \leq \|g - y\|_1 \quad \text{for all } y \in \mathbb{R}.$$

Thus, in order to find a minimizer u of the L^1 -Potts problem, it is sufficient to find its jump locations x_i . Then, a corresponding minimizer is given by $u = \sum_i \mu_i \mathbf{1}_{(x_i, x_{i+1})}$, where μ_i is a median of the data f on the interval (x_i, x_{i+1}) . Recall that a *median* μ of a real-valued function f w.r.t. some finite measure ν on Ω is characterized by the property

$$(2.7) \quad \nu(\{x : f(x) < \mu\}) \leq \frac{1}{2}\nu(\Omega) \quad \text{and} \quad \nu(\{x : f(x) > \mu\}) \leq \frac{1}{2}\nu(\Omega).$$

In our case, we use weighted counting measures in the discrete time case and the Lebesgue measure, on the interval in the continuous time case. If f is a continuous function and ν is the Lebesgue measure, then there is only one unique median, whereas in general, there might be a whole interval of medians.

This nonuniqueness of the median is one source of nonuniqueness of minimizers of the (discrete and continuous) L^1 -Potts functional. In addition to this, there are two other sources of nonuniqueness, as the following example shows.

Example 2.1. Consider the data $f = \mathbf{1}_{[0, \frac{1}{4}]} + \mathbf{1}_{[\frac{1}{2}, \frac{3}{4}]}$ on the interval $[0, 1]$. There are three sources for nonuniqueness of the minimizers of the L^1 -Potts functional P_γ associated with f depending on the parameter γ : First, for $\gamma > \frac{1}{2}$, a minimizer of P_γ is constant on $[0, 1]$, and any constant candidate $u_a(x) = a$, $a \in [0, 1]$, has Potts value $P_\gamma = \frac{1}{2}$. Here, the reason for a nonunique minimizer is the nonuniqueness of the median. Second, for $\gamma = \frac{1}{2}$, we easily check that the three-jump solution $u = f$ and the one-jump solution $u = \mathbf{1}_{[0, \frac{1}{4}]}$ attain the same minimal value. Thus, in that case, there is nonuniqueness w.r.t. the number of jumps. Third, if $\frac{1}{8} < \gamma < \frac{1}{4}$, both $u_1 = \mathbf{1}_{[0, \frac{1}{4}]}$ and $u_2 = \mathbf{1}_{[0, \frac{3}{4}]}$ are minimizers of the L^1 -Potts functional. Here, the number of jumps is equal, but the minimizer is not unique w.r.t. the partitions. (For completeness: for $\gamma < \frac{1}{8}$ the minimizer coincides with data f ; for $\gamma = \frac{1}{8}$, possible minimizers are data f as well as the one jump solutions u_1 and u_2 .)

Note that the first type of nonuniqueness is specific to the L^1 -Potts functional. Indeed, since the mean value is unique, this kind of nonuniqueness does not occur for the L^2 -Potts functional. The other two types of nonuniqueness also arise in the L^2 case but have been shown to occur, in the discrete setting, only on a negligible set of data f and parameters γ [29]. We strongly conjecture that analogous statements (modulo the nonuniqueness of medians) can be made for the L^1 -Potts functional. However, a rigorous proof is out of the scope of this article.

2.2. Γ convergence of semidiscrete Potts functionals. We recall that a sequence (F^k) of functionals on $L^1[0, 1]$ taking values in the extended positive real line $[0, \infty]$ is Γ -convergent to a functional F^∞ if

- (i) for each $u \in L^1[0, 1]$ and each sequence $(u^k)_{k \in \mathbb{N}}$ in $L^1[0, 1]$ with $u^k \rightarrow u$ as $k \rightarrow \infty$ holds

$$(2.8) \quad F^\infty(u) \leq \liminf_{k \rightarrow \infty} F^k(u^k),$$

- (ii) and for each $u \in L^1[0, 1]$ there is a *recovery sequence* $(u^k)_{k \in \mathbb{N}}$ in $L^1[0, 1]$ with $u^k \rightarrow u$ as $k \rightarrow \infty$ such that

$$(2.9) \quad F^\infty(u) \geq \limsup_{k \rightarrow \infty} F^k(u^k).$$

The fundamental theorem of Γ -convergence is as follows.

THEOREM 2.1. *Let (F^k) be a sequence of functions on a metric space X with values on the extended real line. Assume that (F^k) is Γ -convergent to F^∞ . If there is a*

compact set $K \subset X$ such that $\inf_X F^k = \inf_K F^k$ for all k , then F^∞ has a minimizer and a minimizing value is given by $\lim_k \inf_X F^k$. Moreover, if (u^k) is a precompact sequence such that $\lim_k F^k(u^k) = \lim_k \inf_X F^k$, then every limit of a subsequence of (u^k) is a minimum point for F^∞ .

For a proof and further information see, e.g., [6]. Our first goal is to show (2.8) for the semidiscrete Potts functionals Q_γ^k . To this end, we need some preparatory lemmas on piecewise constant functions.

LEMMA 2.2. *The set of piecewise constant functions which have at most j jumps and which are bounded by the constant C (w.r.t. the sup-norm)*

$$(2.10) \quad M_{C,j} = \left\{ g = \sum_{l=1}^{j+1} a_l \mathbf{1}_{(x_{l-1}, x_l)} : 0 = x_0 \leq \dots \leq x_{j+1} = 1, |a_l| \leq C \right\}$$

is a compact subset of $L^1[0, 1]$.

Proof. We have that $M_{C,j}$ is the image of the compact set

$$A = \{ (x, a) \in \mathbb{R}^{j+2} \times \mathbb{R}^{j+1} : 0 = x_0 \leq \dots \leq x_{j+1} = 1, |a_l| \leq C \}$$

under the mapping $e(x, a) = \sum_{l=1}^{j+1} a_l \mathbf{1}_{(x_{l-1}, x_l)}$, which is a continuous mapping into $L^1[0, 1]$ for the following reason. We fix $(x, a) \in A$ and consider a small perturbation $(x + \varepsilon_1, a + \varepsilon_2)$ in A . (In the following, we always consider representatives of the L^1 elements $e(x, a)$ which are constant between the x_l .) If $|e(x, a)(t) - e(x + \varepsilon_1, a + \varepsilon_2)(t)| > \|\varepsilon_2\|_\infty$, then t must be near a jump, i.e., there is an index i such that t is contained in the interval $I_i = [\min(x_i, x_i + \varepsilon_{1,i}), \max(x_i, x_i + \varepsilon_{1,i})]$. The Lebesgue measure of the union of these intervals I_i equals $\|\varepsilon_1\|_1$ and since $e(x, a), e(x + \varepsilon_1, a + \varepsilon_2) \in A$, their absolute value does not exceed C . Therefore, we have

$$\begin{aligned} \|e(x + \varepsilon_1, a + \varepsilon_2) - e(x, a)\|_1 &= \|e(x + \varepsilon_1, a + \varepsilon_2) - e(x, a)|_{\cup I_i}\|_1 \\ &\quad + \|e(x + \varepsilon_1, a + \varepsilon_2) - e(x, a)|_{[0,1] \setminus \cup I_i}\|_1 \\ &\leq 2C\|\varepsilon_1\|_1 + \|\varepsilon_2\|_\infty, \end{aligned}$$

which implies that e is continuous. (In principle, we showed that the mappings $(x_i, x_{i+1}) \mapsto \mathbf{1}_{[x_i, x_{i+1}]}$ are continuous and use the continuity of the vector space operations.) Hence, $M_{C,j}$ is compact. \square

LEMMA 2.3. *Let u be a piecewise constant function with minimal jump height h and minimal interval width l . Then, for any piecewise constant function w , we have the implication*

$$(2.11) \quad J(w) < J(u) \quad \implies \quad \|w - u\|_1 \geq h \cdot l.$$

Proof. We first show that it is sufficient to prove (2.11) for functions w whose jump set is included in the jump set of u . To this end, we consider a general w with fewer jumps than u and shift its jump locations to those of u such that the resulting w' fulfills $J(w) \geq J(w')$ and $\|w - u\|_1 \geq \|w' - u\|_1$. In the following, we construct such a function w' . Let $X = \{x_1, \dots, x_n\}$ be the jump set of $u = \sum_i a_i \mathbf{1}_{(x_i, x_{i+1})}$ and let $T = \{t_1, \dots, t_r\}$ be the jump set of w . Denote $S = T \setminus X$ the jump locations of w which are not in the jump locations of u . We reduce S to \emptyset by the following recursive procedure. If $S = \emptyset$, we are done. If $S \neq \emptyset$, we pick a jump point $t_i \in S$. Then we consider the difference between u and $w = \sum_j b_j \mathbf{1}_{(t_j, t_{j+1})}$ to the left of t_i , i.e., $|u(t_i) - b_{i-1}|$ and the difference to the right $|u(t_i) - b_i|$. If $|u(t_i) - b_{i-1}| \geq |u(t_i) - b_i|$,

then we shift t_i to the left until we meet the next jump point $p \in X \cup T$. (If there is no such point, we shift t_i to 0 and let $p = 0$.) If the point p is in $X \setminus T$, say, p equals x_l , then w' reads as

$$w' = \sum_{j \neq i-1, i} b_j \mathbf{1}_{(t_j, t_{j+1})} + b_{i-1} \mathbf{1}_{(t_{i-1}, x_l)} + b_i \mathbf{1}_{(x_l, t_{i+1})}.$$

We have $J(w') = J(w)$, $\|w - u\|_1 \geq \|w' - u\|_1$, and w' has one more jump in the jump set of u than w . If p is in T or $p = 0$, the procedure reduces the number of jumps of w by 1 and still $\|w - u\|_1 \geq \|w' - u\|_1$. In any case, the number of jumps which are not in X is reduced by one. In the opposite case, $|u(t_i) - b_{i-1}| < |u(t_i) - b_i|$, we shift to the right and proceed analogously (where we shift to 1 and let $p = 1$ whenever there is no jump of u or w being further to the right). Iterating this algorithm, we obtain a w' whose jump set is contained in that of u . By construction, this w' still has fewer jumps than u , i.e., $J(w') < J(u)$, and $\|w - u\|_1 \geq \|w' - u\|_1$.

In consequence, we find an interval (t_i, t_{i+1}) where w' does not jump but u does so at $t_i < p < t_{i+1}$. Then

$$(2.12) \quad \|w' - u\|_1 \geq |u(p^+) - u(p^-)| \cdot \min(p - t_i, t_{i+1} - p) \geq hl,$$

which shows (2.11). \square

We now show the first condition for Γ -convergence (2.8) for the semidiscrete Potts functionals (Q_γ^k) .

LEMMA 2.4. *Let $u \in L^1[0, 1]$ and let (u^k) be a sequence in $L^1[0, 1]$ such that $\|u^k - u\|_1 \rightarrow 0$ for $k \rightarrow \infty$. Then $P_\gamma(u) \leq \liminf_{k \rightarrow \infty} Q_\gamma^k(u^k)$.*

Proof. We may restrict ourselves to sequences (u^k) in $\text{PC}[0, 1]$. We first consider $u \in L^1$ which is not essentially bounded, i.e., $u \notin L^\infty$. We show that, for each sequence of piecewise constant functions (u^k) converging to u , the number of jumps tends to infinity, which then yields Lemma 2.4 for essentially unbounded u . To this end, we construct a sequence of disjoint intervals (I_i) and a sequence of positive numbers (ε_i) for which we show the following: for an approximating piecewise constant function \tilde{u} , $\|\tilde{u} - u\|_1 < \varepsilon_i$ implies that $|\tilde{u}|$ takes values in each of the disjoint sets I_1, \dots, I_i . Since u is essentially unbounded, the Lebesgue measure $\lambda(\{|u| \geq n\})$ is positive for any positive integer n . We choose $n_1 = 1$ and let $\varepsilon_1 = \lambda(\{|u| \geq 2n_1\})/2$. Using the Markov inequality, we find $m_1 > 2n_1$ such that $\lambda(\{|u| > m_1\}) \leq \varepsilon_1$. This implies $\lambda(\{2n_1 \leq |u| \leq m_1\}) \geq \varepsilon_1$. If $|\tilde{u}|$ does not take a value in the interval $I_1 = [n_1, 2m_1] \cup [-2m_1, -n_1]$, then we can estimate its approximation error by

$$\|\tilde{u} - u\|_1 \geq \|(\tilde{u} - u)|_{\{2n_1 \leq |u| \leq m_1\}}\|_1 \geq n_1 \cdot \lambda(\{2n_1 \leq |u| \leq m_1\}) \geq \varepsilon_1.$$

We choose $n_2 = 2m_1 + 1$ and let $\varepsilon_2 = \lambda(\{|u| > 2n_2\})/2$. We proceed likewise to obtain intervals I_i and positive numbers ε_i with the claimed properties. Then, if (u^k) converges to u , there is some index k_i such that $\|u^l - u\|_1 < \varepsilon_i$ for all $l > k_i$. Each such piecewise constant function u^l must take values in each of the intervals I_1, \dots, I_i . Thus the number of jumps of the sequence (u^k) tends to infinity.

Next, we consider essentially bounded u . We show that, if (u^k) tends to u and the number of jumps of the u^k is (uniformly) bounded, then u is piecewise constant (i.e., it has a piecewise constant representative). This implies that for nonpiecewise constant u , the number of jumps of a sequence (u^k) converging toward u must go to infinity for the following reason. If there was a subsequence of u^k with a uniformly bounded number of jumps, its limit would be piecewise constant. Furthermore, the limit would

equal u and so u would be piecewise constant, which would be a contradiction. Hence, in order to show Lemma 2.4 for nonpiecewise constant essentially bounded functions, we have to show that the uniform boundedness of the number of jumps of a converging sequence (u^k) implies the piecewise constance of its limit.

In a first step, we construct a sequence (\tilde{u}^k) which converges to u and which is uniformly bounded (w.r.t. the sup-norm) as follows. We define the piecewise constant function \tilde{u}^k by u^k except for the intervals on which $|u^k| > 2\|u\|_\infty$. Here, we let $\tilde{u}^k = 0$. Then (\tilde{u}^k) is uniformly bounded by $2\|u\|_\infty$ and $\|\tilde{u}^k - u\|_1 \leq \|u^k - u\|_1$. This entails that (\tilde{u}^k) converges to u . So we may assume that (u^k) tends to u in L^1 , that the piecewise constant functions (u^k) are uniformly bounded (w.r.t. the sup-norm), and that the number of jumps of the (u^k) is bounded as well. Then the piecewise constance of u is a consequence of the compactness of the sets $M_{C,j}$ defined by (2.10) in Lemma 2.2.

So far, we have shown Lemma 2.4 for nonpiecewise constant u . Now we consider a piecewise constant function u and show the assertion in that case. As a consequence of (2.11) in Lemma 2.3, we find for each sequence of piecewise constant functions (u^k) converging to u in L^1 an index k_0 such that, for all $k \geq k_0$,

$$(2.13) \quad J(u^k) \geq J(u).$$

Equipped with this estimate, we show the assertion of the lemma for piecewise constant functions u . Since $Q_\gamma^k(v) = \infty$ for $v \notin \text{PC}^k[0, 1]$, we may assume without loss of generality that u^k is in $\text{PC}^k[0, 1]$. Then $P_\gamma(u^k) = Q_\gamma^k(u^k)$ and thus, using (2.13),

$$(2.14) \quad \begin{aligned} P_\gamma(u) = \gamma J(u) + \|u - f\|_1 &\leq \liminf_k \gamma J(u^k) + \lim_k \|u^k - f\|_1 \\ &= \liminf_k P_\gamma(u^k) = \liminf_k Q_\gamma^k(u^k), \end{aligned}$$

which completes the proof. \square

Next we show the second condition for Γ -convergence (2.9).

LEMMA 2.5. *For each u in $L^1[0, 1]$, there is a recovery sequence (u^k) which converges to u in L^1 and which fulfills $P_\gamma(u) \geq \limsup_{k \rightarrow \infty} Q_\gamma^k(u^k)$.*

Proof. We may restrict u to $\text{PC}[0, 1]$, since otherwise $P_\gamma(u) = \infty$. For $k \in \mathbb{N}$, we define ϵ_k as the largest gap in X^k , i.e., $\epsilon_k = \sup_l |x_l - x_{l+1}|$, where the x_l are a monotone ordering of the points in X^k (including 0 and 1). Since the union of discretization sets X^k is dense in $[0, 1]$, the sequence (ϵ_k) converges to 0 as $k \rightarrow \infty$.

Let us fix k for the moment. For every jump location t_j of u , we find a closest discretization point x_j such that $|x_j - t_j| < \epsilon_k$. Then, for sufficiently large k , replacing each interval (t_i, t_{i+1}) in $u = \sum_i a_i \mathbf{1}_{(t_i, t_{i+1})}$ by the interval (x_i, x_{i+1}) , we obtain a piecewise constant function u^k such that

$$\|u - u^k\|_1 \leq \epsilon_k \cdot H \cdot J(u).$$

Here, H is the maximal jump height of u . For sufficiently large k , u and u^k have the same number of jumps, since (ϵ_k) converges to 0 and so each jump point of u is assigned exactly one point in X^k . Therefore, we have

$$\begin{aligned} P_\gamma(u) = \gamma J(u) + \|u - f\|_1 &\geq \gamma J(u^k) + \|f - u^k\|_1 - \|u^k - u\|_1 \\ &\geq P_\gamma(u^k) - \epsilon_k \cdot H \cdot J(u). \end{aligned}$$

Rewriting this as $P_\gamma(u) + \epsilon_k \cdot H \cdot J(u) \geq P_\gamma(u^k) = Q_\gamma^k(u^k)$ and passing to the limit $k \rightarrow \infty$ yields the assertion. \square

The following lemma is an auxiliary statement needed for the proof of the next theorem, which is our main result on the semidiscrete Potts functionals (Q_γ^k) . It is also needed in the proof of Theorem 2.9.

LEMMA 2.6. *Let $f \in L^1[0, 1]$ and let f^k , $k = 1, 2, \dots$ be the functions obtained from f via the integral sampling (2.3). For $\varepsilon > 0$, there is a constant $C > 0$ such that, for any $g \in \{f, f^1, f^2, \dots\}$,*

$$(2.15) \quad |\mu_I(g)| \geq C \quad \implies \quad \int_I |g| \leq \varepsilon.$$

Here, $\mu_I(g)$ denotes a median of g on the interval I . The interval I is arbitrary for $g = f$, and for $g = f^k$, we assume that the endpoints of I are in the discretization set X^k .

Proof. For $h \in L^1$, we have the following connection between the absolute value of a median $\mu_I(h)$ on some interval I and the length of the interval $|I|$

$$(2.16) \quad |\mu_I(h)| \geq c \quad \implies \quad |I| \leq \frac{2\|h\|_1}{c} \quad \text{for any } c > 0.$$

Indeed, $|\mu_I(h)| \geq c$ implies that $\lambda(\{|h| \geq c\}) \geq |I|/2$, and the Markov inequality yields $\lambda(\{|h| \geq c\}) \leq \|h\|_1/c$. We apply (2.16) to f^k and note that since $f^k = \sum_l S_k f(l) \cdot \mathbf{1}_{I_l}$, we have $\|f^k\|_1 \leq \|f\|_1$. We get that, for any interval I ,

$$(2.17) \quad |\mu_I(f^k)| \geq c \quad \implies \quad |I| \leq \frac{2\|f\|_1}{c} \quad \text{for any } c > 0.$$

Furthermore, if the interval length $|I|$ is small, the integral of $|f|$ on I is small, i.e., for $\varepsilon > 0$ there is $\delta > 0$ such that

$$(2.18) \quad |I| \leq \delta \quad \implies \quad \int_I |f| \leq \varepsilon.$$

This implication can be seen by considering the finite measure $|f|d\lambda$ with density $|f|$ w.r.t. the Lebesgue measure λ . The corresponding distribution function is absolutely continuous and thus uniformly continuous, which is precisely the statement of (2.18). Since $\int_I |f^k| \leq \int_I |f|$, for all intervals I whose endpoints are in the discretization set X^k , we get that (2.18) implies

$$(2.19) \quad |I| \leq \delta \quad \implies \quad \int_I |f^k| \leq \varepsilon$$

for all such intervals I . Combining (2.17) and (2.19) yields the assertion of the lemma for $g = f^k$, and combining (2.16) for $h = f$ and (2.18) yields the assertion for $g = f$. \square

We formulate our main result on the semidiscrete Potts functionals (Q_γ^k) .

THEOREM 2.7. *Let f be an integrable function on $[0, 1]$ and let P_γ be the corresponding continuous L^1 -Potts functional. Then the semidiscrete Potts functionals (Q_γ^k) w.r.t. a nested sequence of discretization sets (X^k) Γ -converge to P_γ . Furthermore, each functional Q_γ^k has a minimizer. Each sequence (u^k) , where u^k is a minimizer of Q_γ^k , has at least one accumulation point. Each such accumulation point u^* is a minimizer of P_γ , i.e., $P_\gamma(u^*) = \inf_{v \in L^1} P_\gamma(v)$.*

As an immediate consequence, we have the following statement.

THEOREM 2.8. *The continuous L^1 -Potts functional P_γ has a minimizer.*

Proof of Theorem 2.7. Our first aim is to show that all the minimizers of the functionals (Q_γ^k) are contained in a compact subset M of $L^1[0, 1]$, which is independent

of k . We let $M = M_{C,j}$ be the compact set from (2.10), where we define the maximal number of jumps j by the smallest integer larger than $\|f\|_1/\gamma$. The constant C is defined in the course of the proof. To show that all the minimizers are contained in M , we first prove that

$$(2.20) \quad \inf_{u \in M} Q_\gamma^k(u) = \inf_{u \in \text{PC}[0,1]} Q_\gamma^k(u) < Q_\gamma^k(v) \text{ for all } v \notin M.$$

To this end, we consider an arbitrary piecewise constant function $u \notin M$ and show that there is a piecewise constant function \tilde{u} contained in M with lower Potts value. Because of the minimizing property (2.6) of the median, we may assume that $u = \sum_i \alpha_i \mathbf{1}_{(x_i, x_{i+1})}$ is given by a median μ_I on each interval $I = (x_i, x_{i+1})$. If the number of jumps of u obeys $J(u) > j > \|f\|_1/\gamma$, then $\tilde{u} = 0$ has a smaller Potts functional value since

$$(2.21) \quad P_\gamma(u) \geq \gamma J(u) > \|f\|_1 \geq P_\gamma(0) \geq \inf_{v \in \text{PC}[0,1]} P_\gamma(v).$$

Therefore, we may assume that u has at most j jumps. Now we define the constant C in $M_{C,j}$ as the resulting number C in (2.15) for $g = f$ and the choice $\varepsilon = \gamma/6$. This means

$$(2.22) \quad |\mu_I(f)| \geq C \quad \implies \quad \int_I |f| \leq \gamma/6 \quad \text{for any interval } I.$$

If $C < 3\|f\|_1$, we enlarge it such that $C \geq 3\|f\|_1$. We observe that there is at least one interval $I = (x_i, x_{i+1})$ where $|u| \leq C$. If this was not the case, we would have $\lambda(\{|f| > C\}) \geq 1/2$. This contradicts the assumption $C \geq 3\|f\|_1$ since $\lambda(\{|f| > C\}) \leq \|f\|_1/C \leq 1/3$. We start with an interval I where $|u| \leq C$ and define $\tilde{u} = u$ on I . Then we consider a (left or right) neighbor interval J . (Such a neighboring interval always exists since we assumed that u is not in $M_{C,j}$.) If $|u| \leq C$ on J , we define $\tilde{u} = u$ on J and proceed with a neighbor of I or J . If $|u| > C$ on J , we define $\tilde{u}|_J = u|_I$, i.e., we remove a jump. We have $J(\tilde{u}) = J(u) - 1$. Furthermore, $\|\tilde{u}|_J\|_1 \leq \|u|_J\|_1$, since $|u|_I \leq C < |u|_J$. Also, $\|u|_J\|_1 \leq 2\|f|_J\|_1$ because the value of u on J is a median of f on J and thus $\|u|_J\|_1 - \|f|_J\|_1 \leq \|(f - u)|_J\|_1 \leq \|f|_J\|_1$. By (2.22), $\|(\tilde{u} - f)|_J\|_1 \leq \|\tilde{u}|_J\|_1 + \|f|_J\|_1 \leq 3\|f|_J\|_1 \leq \gamma/2$. Consequently,

$$(2.23) \quad P_\gamma(\tilde{u}) = \gamma J(\tilde{u}) + \|f - \tilde{u}\|_1 \leq \gamma J(u) - \gamma + \|f - u\|_1 + \gamma/2 < P_\gamma(u).$$

Proceeding likewise for the other intervals, we obtain a piecewise constant function \tilde{u} which is smaller than C in absolute value and thus is contained in $M_{C,j}$. Furthermore, $P_\gamma(\tilde{u}) < P_\gamma(u)$. Since $P_\gamma = Q_\gamma^k$ for piecewise constant functions which only jump in the discretization set X^k and since the jump set of \tilde{u} is contained in the jump set of u , (2.23) remains true when we replace P_γ by Q_γ^k . This shows (2.20).

We are now able to prove the assertions of the theorem. Lemmas 2.4 and 2.5 imply the Γ -convergence of (Q_γ^k) to P_γ . Since (2.20) holds, we deduce from the lower semicontinuity of Q_γ^k (which is a consequence of (2.14)) and the compactness of M that each Q_γ^k has a minimizer which then is in M . Furthermore, the compactness of M implies the existence of a cluster point for any sequence of minimizers (u^k) . Then the last assertion of the theorem is a consequence of the fundamental theorem of Γ -convergence; cf. Theorem 2.1 or [6, Theorem 1.21, p. 29]. \square

2.3. Γ convergence of discrete Potts functionals. We use the results on the semidiscrete Potts functional to obtain Γ -convergence w.r.t. the L^1 -norm for the discrete Potts functionals defined by (2.5).

THEOREM 2.9. *Let f be an integrable function and let P_γ be the corresponding continuous L^1 -Potts functional. Then the discrete L^1 -Potts functionals (P_γ^k) Γ -converge to P_γ . Each sequence (u^k) , where u^k is a minimizer of P_γ^k , has at least one accumulation point. Each such accumulation point u^* is a minimizer of P_γ , i.e., $P_\gamma(u^*) = \inf_{v \in L^1[0,1]} P_\gamma(v)$.*

Proof. We start showing the statement on Γ -convergence. We first establish a quantitative relation between P_γ^k and Q_γ^k . We may estimate, for any $u \in \text{PC}^k[0, 1]$,

$$\begin{aligned} P_\gamma^k(u) &= \gamma \cdot J(u) + \|u - f\|_1 \\ &\leq \gamma \cdot J(u) + \left\| u - \sum_j S_k f(j) \cdot \mathbf{1}_{I_j} \right\|_1 + \left\| f - \sum_j S_k f(j) \cdot \mathbf{1}_{I_j} \right\|_1 \\ (2.24) \quad &= Q_\gamma^k(u) + \left\| f - \sum_j S_k f(j) \cdot \mathbf{1}_{I_j} \right\|_1. \end{aligned}$$

If S_k is given by the integral sampling (2.3), we have the following inequality:

$$(2.25) \quad \left\| f - \sum_j S_k f(j) \cdot \mathbf{1}_{I_j} \right\|_1 \leq C \cdot \omega(f, \sup_j |I_j|),$$

where C is a positive constant and ω is the L^1 modulus of continuity given by $\omega(f, t) = \sup_{|h| \leq t} \|f(\bullet + h) - f\|_1$, cf. [11]. Since the translation is continuous in L^1 and the union of the nested sampling sets X^k is dense, it follows by (2.25) that $\|f - \sum_j S_k f(j) \cdot \mathbf{1}_{I_j}\|_1 \rightarrow 0$, as $k \rightarrow \infty$. Now exchanging the roles of P_γ^k and Q_γ^k in (2.24), we conclude that

$$(2.26) \quad |P_\gamma^k(u) - Q_\gamma^k(u)| \leq \left\| f - \sum_j S_k f(j) \cdot \mathbf{1}_{I_j} \right\|_1 \rightarrow 0, \quad \text{as } k \rightarrow \infty.$$

We emphasize that the above convergence is uniform in u . If we consider point sampling (2.4) and a continuous function f , (2.26) remains true. This is because

$$(2.27) \quad \left\| f - \sum_j S_k f(j) \mathbf{1}_{I_j} \right\|_1 \leq \sup_j \sup_{x, y \in I_j} |f(x) - f(y)|$$

tends to zero as k increases since f is uniformly continuous on $[0, 1]$ and the maximum interval length $|I_j|$ goes to zero since the union of the nested X^k is dense.

In order to show Γ -convergence, we show (2.8) and (2.9) for the discrete Potts functionals. Let us consider a sequence (u^k) which converges to u in L^1 . Then $Q_\gamma^k(u^k) \leq P_\gamma^k(u^k) + |P_\gamma^k(u^k) - Q_\gamma^k(u^k)|$. This, together with Lemma 2.4, in turn implies that

$$(2.28) \quad \begin{aligned} P_\gamma(u) &\leq \liminf_{k \rightarrow \infty} Q_\gamma^k(u^k) \\ &\leq \liminf_{k \rightarrow \infty} P_\gamma^k(u^k) + \limsup_{k \rightarrow \infty} \sup_{n \geq k} |P_\gamma^n(u^n) - Q_\gamma^n(u^n)|. \end{aligned}$$

Since the convergence in (2.26) is uniform in u , the second summand on the right-hand side is 0, which shows (2.8). In order to obtain a recovery sequence for the discrete Potts functionals (P_γ^k) , we consider a recovery sequence (u^k) for the semidiscrete Potts functionals (Q_γ^k) , which exists according to Lemma 2.5. Then, also by Lemma 2.5,

$$\begin{aligned} P_\gamma(u) &\geq \limsup_{k \rightarrow \infty} Q_\gamma^k(u^k) \\ &\geq \limsup_{k \rightarrow \infty} P_\gamma^k(u^k) - \limsup_{k \rightarrow \infty} \sup_{n \geq k} |P_\gamma^n(u^n) - Q_\gamma^n(u^n)|. \end{aligned}$$

As above, the second summand on the right-hand side is 0 and so (u^k) is a recovery sequence for (P_γ^k) , too. This shows (2.9), and we have shown Γ -convergence.

Our next goal is to locate all minimizers of (P_γ^k) in a common compact set M . To this end, we show the analogous statement to (2.20) replacing the semidiscrete Potts functional by their discrete counterparts, i.e.,

$$(2.29) \quad \inf_{u \in M} P_\gamma^k(u) = \inf_{u \in \text{PC}[0,1]} P_\gamma^k(u) < P_\gamma^k(v) \text{ for all } v \notin M.$$

The proof is based on the corresponding proof of (2.20), which can be found in the proof of Theorem 2.7.

We first consider the case where S_k is the integral sampling operator (2.3). We choose the compact set $M = M_{C,j}$ with C and j as in the proof of Theorem 2.7. In particular, this yields the implication (2.15) for $g = f^k$ and $\varepsilon = \gamma/6$. Now we can consider a piecewise constant function $u \notin M$ whose jump set is contained in X^k and proceed as in the proof of Theorem 2.7 to obtain a piecewise constant function \tilde{u} whose jump set is contained in X^k , which has smaller P_γ^k value, and which is contained in $M_{C,j}$. This in turn implies (2.29).

If S_k is the point sampling operator (2.4) and if f is continuous, we have $\|f^k\|_1 \leq \|f^k\|_\infty \leq \|f\|_\infty$. Then (2.16) applied to f^k yields the implication (2.17) with $\|f\|_1$ replaced by $\|f\|_\infty$. In this case, (2.19) is trivial since $\int_I |f^k| \leq \|f\|_\infty \cdot |I|$. As in the case of integral sampling, we use these two implications to establish (2.15) for $g = f^k$, which allows us to proceed as in the proof of Theorem 2.7 to obtain (2.29).

Each P_γ^k has a minimizer since it is equal to a semidiscrete Potts functional for data $\sum_j S_k f(j) \mathbf{1}_{I_j}$, which has a minimizer by Theorem 2.7. Equipped with (2.29) and knowing that each P_γ^k has a minimizer, we conclude that all minimizers are contained in $M = M_{C,j}$. The compactness of M implies the existence of a cluster point for any sequence of minimizers (u^k) . Then the last assertion of the theorem is a consequence of the fundamental theorem of Γ -convergence; cf. Theorem 2.1 or [6, Theorem 1.21, p. 29]. \square

3. A fast algorithm for the exact minimization of L^1 -Potts functionals.

In this section we propose a fast algorithm to minimize weighted Potts functionals of the form

$$(3.1) \quad \|u - f\|_{\ell_w^1} + \gamma J(u) = \sum_i w_i |u_i - f_i| + \gamma \cdot \#\{i : u_i \neq u_{i+1}\},$$

where the w_i are positive weights. Weighted ℓ^1 -data terms appear naturally when data are not sampled equidistantly. Clearly, if all weights w_i are identical, this reduces to the nonweighted L^1 -Potts functional (1.1).

In subsection 3.1 we recall the Potts core algorithm as proposed in [16]. In subsection 3.2 we introduce a suitable dynamic data structure. In subsection 3.3

we explain our fast algorithm and obtain results on its complexity in subsection 3.4. In subsection 3.5 we experimentally compare L^1 -Potts, L^2 -Potts, and L^1 -TV minimization under various types of noise.

3.1. The Potts core algorithm. 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 reduced data $f_{[1,1]}, f_{[1,2]}, \dots, f_{[1,r-1]}$ are known [23, 8]. Here, we use the notation $f_{[\ell,r]}$ for the partial data $(f_\ell, f_{\ell+1}, \dots, f_r)$. The procedure works as follows. Let us denote by u^1, u^2, \dots, u^{r-1} arbitrary (not necessarily unique) minimizers of the Potts functional for the partial data $f_{[1,1]}, f_{[1,2]}, \dots, f_{[1,r-1]}$, respectively. In order to compute a minimizer for data $f_{[1,r]}$, we first create a set of r candidates v^1, \dots, v^r , each of length r , which are given by

$$(3.2) \quad v^\ell = (u^{\ell-1}, \underbrace{\mu_{[\ell,r]}, \dots, \mu_{[\ell,r]}}_{\text{Length } r-\ell+1}).$$

Here, u^0 is the empty vector and the symbol $\mu_{[\ell,r]}$ denotes an (arbitrary) weighted median for data $f_{[\ell,r]}$. Recall that a weighted median is a minimizer of the functional $y \mapsto \|(y, \dots, y) - f_{[\ell,r]}\|_{\ell_w^1}$ and that the minimal functional value is given by the median deviation

$$(3.3) \quad d_{[\ell,r]} = \|f_{[\ell,r]} - (\mu_{[\ell,r]}, \dots, \mu_{[\ell,r]})\|_{\ell_w^1} = \sum_{i=\ell}^r w_i |f_i - \mu_{[\ell,r]}|.$$

Among the candidates v^ℓ , there is a minimizer of the Potts functional, as the following lemma shows.

LEMMA 3.1. *For the Potts functional associated with data $f_{[1,r]}$, we have that*

$$\min_{\ell=1, \dots, r} P_\gamma(v^\ell) = \min_{u \in \mathbb{R}^r} P_\gamma(u).$$

In particular, the Potts functional value of a minimizer u^* is given by

$$(3.4) \quad P_\gamma(u^*) = \min_{\ell=1, \dots, r} P_\gamma(u^{\ell-1}) + \gamma + d_{[\ell,r]},$$

where we set $P_\gamma(u^0) = -\gamma$.

Proof. Let u^* be a minimizer of the Potts functional for data $f_{[1,r]}$. If u^* is constant, then $P_\gamma(u^*) = P_\gamma(v^1)$ because both u^* and v^1 are constant vectors containing a median of data $f_{[1,r]}$; hence, v^1 is a minimizer of the Potts functional. In particular, we have $P_\gamma(u^*) = d_{[1,r]} = P_\gamma(u^0) + \gamma + d_{[1,r]}$, which shows (3.4) in the constant case.

If u^* is not constant, let $\ell^* \in \{1, \dots, r-1\}$ be its rightmost jump location, i.e., $u_{\ell^*}^* \neq u_{\ell^*+1}^*$ and u^* is constant on ℓ^*+1, \dots, r . Since u^{ℓ^*} minimizes the Potts functional associated with data $f_{[1,\ell^*]}$ and since u^* has a jump at ℓ^* , we have for the candidate v^{ℓ^*+1} , which jumps at ℓ^* , that

$$(3.5) \quad P_\gamma(v^{\ell^*+1}) \leq P_\gamma(u^{\ell^*}) + \gamma + d_{[\ell^*+1,r]} \leq P_\gamma(u_{[1,\ell^*]}^*) + \gamma + d_{[\ell^*+1,r]} = P_\gamma(u^*).$$

Here, $u_{[1,\ell^*]}^*$ denotes the vector of the first ℓ^* components of u^* . Hence, v^{ℓ^*+1} is a minimizer of the Potts functional. Further, since $P_\gamma(v^\ell) \leq P_\gamma(u^{\ell-1}) + \gamma + d_{[\ell,r]}$ for all $\ell = 1, \dots, r$, we have that

$$\min_{\ell=1, \dots, r} P_\gamma(v^\ell) \leq \min_{\ell=1, \dots, r} P_\gamma(u^{\ell-1}) + \gamma + d_{[\ell,r]} \leq P_\gamma(u^{\ell^*}) + \gamma + d_{[\ell^*+1,r]} \leq P_\gamma(u^*).$$

Using $P_\gamma(u^*) \leq \min_{\ell=1, \dots, r} P_\gamma(v^\ell)$, we obtain (3.4) in the nonconstant case. \square

Recall from Example 2.1 that minimizers of the L^1 -Potts functional are not unique in general, so it is possible that several candidates v^ℓ fulfill $P_\gamma(v^\ell) = P_\gamma(u^*)$. Note that, for such a minimizer v^ℓ , the values $v_{\ell-1}^\ell$ and v_ℓ^ℓ are different, i.e., v^ℓ in fact has a jump at $\ell - 1$ (despite the possible nonuniqueness of medians). This is a consequence of the proof of Lemma 3.1 above since, otherwise, we could replace the first “ \leq ” symbol in (3.5) by “ $<$,” which would contradict u^* being a minimizer.

Formula (3.4) already yields a procedure to minimize the Potts functional. A speedup is given by a crucial observation of [16]: we can compute a minimizer from knowing the rightmost jump locations of (arbitrary) minimizers of certain data.

LEMMA 3.2. *For $\ell = 1, \dots, r$, let $Z(\ell)$ be the rightmost jump location of an (arbitrary) minimizer u^ℓ of the Potts functional associated with data $f_{[1,\ell]}$. ($Z(\ell) = 0$ if u^ℓ is constant.) We consider the strictly decreasing finite sequence $r, Z(r), Z(Z(r)), Z(Z(Z(r))), \dots, 0$. We define \bar{u} on $Z^{k+1}(r) + 1, \dots, Z^k(r)$ as a median $\mu_{[Z^{k+1}(r)+1, Z^k(r)]}$. Then \bar{u} is a minimizer of the Potts functional for data $f_{[1,r]}$.*

Proof. We use induction on r . For $r = 1$, $Z(r) = 0$ and $u = f_1 = \mu_{[1,1]}$ is a minimizer of P_γ for data f_1 . As induction hypothesis, we assume that the statement is true for all $\ell = 1, \dots, r - 1$. We consider data $f_{[1,r]}$ and the minimizer u^r whose rightmost jump location is $Z(r) < r$. If $Z(r) = 0$, then u^r is constant. By the minimizing property of a median, $\|\bar{u} - f_{[1,r]}\|_{\ell_w^1} \leq \|u^r - f_{[1,r]}\|_{\ell_w^1}$ and thus $P_\gamma(\bar{u}) \leq P_\gamma(u^r)$. Hence, \bar{u} is a minimizer. If $Z(r) \neq 0$, we denote the first $Z(r)$ components of \bar{u} by \bar{u}' . \bar{u}' is a minimizer of P_γ for data $f_{[1,Z(r)]}$ by the induction hypotheses. Thus,

$$P_\gamma(\bar{u}) \leq P_\gamma(\bar{u}') + \gamma + d_{[Z(r)+1,r]} \leq P_\gamma(u^r|_{\{1,\dots,Z(r)\}}) + \gamma + d_{[Z(r)+1,r]} \leq P_\gamma(u^r).$$

This implies that \bar{u} minimizes P_γ for data $f_{[1,r]}$, which completes the induction. \square

Using Lemma 3.2 together with (3.4), we get the following *Potts core algorithm*, which is the core part of Algorithm 1. Assume that, for $\ell = 1, \dots, r - 1$, $Z(\ell)$ and $P_\gamma(u^\ell)$ have been calculated and stored. Calculate $P_\gamma(u^r)$ and $Z(r)$ of an arbitrary minimizer u^r by (3.4). We store these values and proceed with $r + 1$ until we reach n . Then we use Lemma 3.2 to construct a minimizer for full data $f_{[1,n]}$, from the rightmost jump locations. To get a deterministic algorithm, we use the following selection rules: we take a candidate in (3.4) with smallest l , and we always choose smallest medians (when not unique). The Potts core algorithm needs at least $O(n^2)$ steps, since we iterate over $r = 1, \dots, n$ in an outer loop and over $\ell = 1, \dots, r$ in an inner loop. Optimal time complexity can only be achieved if the operations in the inner ℓ -loop take constant time. The critical computation in the inner loop is the evaluation of the right-hand side of (3.4) for fixed ℓ , $\ell = 1, \dots, r$. The stored Potts functional value $P_\gamma(u^{\ell-1})$ can be simply looked up in constant time. Thus, the remaining time critical computations are the evaluations of the median deviations $d_{[\ell,r]}$. So the challenge to get a fast L^1 Potts algorithm is to provide fast computation of the $O(n^2)$ median deviations $d_{[\ell,r]}$ within $O(n^2)$ time and to perform calculations in situ in order not to consume extra memory.

3.2. The indexed linked histogram. For the fast computation of the median deviations (3.3), we introduce the dynamic data structure *indexed linked histogram*.

Basic structure. A histogram associated with data $f = (f_1, \dots, f_m)$ and weights w_i is the image measure $f(w)$ of the measure w given by the w_i under the mapping f . Thus the domain of the histogram is equal to the set of data values of f , and it assigns to each data value f_j the sum $\sum_{i:f_i=f_j} w_i$ of the weights w_i of all data points with value f_j . An indexed linked histogram basically stores a histogram; for an example, see Figure 2. It has four crucial characteristics:

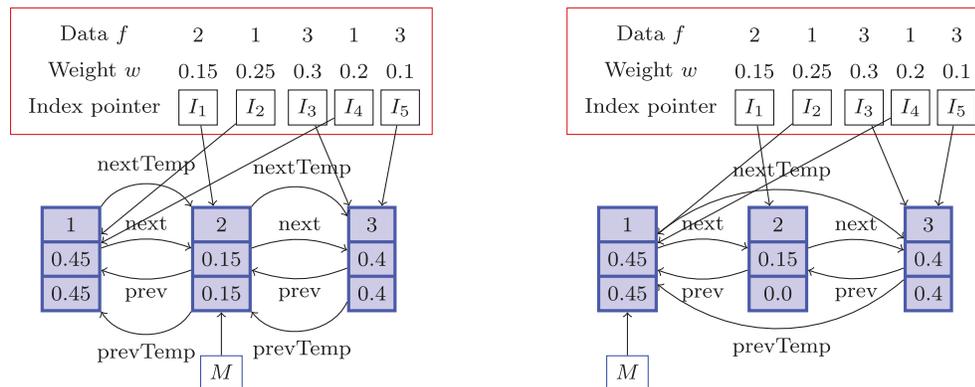


FIG. 2. *Data structure indexed linked histogram.* The shaded rectangles depict the histogram nodes with the entries value, weight, and temporary weight (from top to bottom). Left: The temporary pointers at the beginning of the (inner) ℓ -loop coincide with the nontemporary ones. Right: In the first ℓ -iteration, the left-most weight w_1 is subtracted from the temporary weight of the node corresponding to $f_1 = 2$. Since the temporary weight of that node then equals zero, it is removed from the temporary list. The temporary median pointer M is shifted accordingly and the deviation is updated.

- (i) The histogram is realized as linked list where the pairings “value \mapsto weight” are realized as nodes called *histogram nodes*.
- (ii) The linked list is sorted by the values of the histogram nodes.
- (iii) The histogram nodes have a twofold structure, that is, they have permanent pointers and weights, and extra temporary pointers and temporary weights. This allows one to store a temporary histogram for partial data ($f_{[\ell,r]}$) without losing the histogram for the full data.
- (iv) An array I of *index pointers* links the position i of each data item f_i with its corresponding histogram node.

Building the histogram. Assume that we have a histogram for the data $f_{[1,r]}$. Then we obtain a histogram for the data $f_{[1,r+1]}$ by adding the weight w_{r+1} to the histogram node N with value f_{r+1} (which has to be created, if not existent.) This involves searching (and possibly inserting) in a linked list which is a linear time task. Furthermore, the index pointer I_{r+1} is set to point to N . Then we find the median histogram node and store it in the *median pointer* M . The median pointer can be computed from scratch by a simple iteration over the sorted histogram in linear time or, more efficiently, by restoring and updating the median pointer of the r th iteration. We compute the deviation $d_{[1,r]}$ of the median from the data $f_{[1,r]}$ by simple summation. At last, we initialize the temporary structure by the permanent structure. These are all linear time tasks.

Fast computation of median deviations. Assume that, for data $f_{[\ell,r]}$, we are given an indexed linked histogram, a pointer to its median node, and the median deviation $d_{[\ell,r]}$. Then we compute the median deviation $d_{[\ell+1,r]}$ for the reduced data $f_{[\ell+1,r]}$ as follows:

- (i) We remove the weight w_ℓ from the histogram node corresponding to the data f_ℓ . The crucial point is that due to the indexing by the array I , we find this histogram node in constant time, which is the time-critical task.
- (ii) The weights below or above the (old) median node may have more than half of the total weight. Therefore, we shift the median pointer downward or upward until we reach the (new) median node of the reduced histogram.

Hereby, the number of required shifting operations depends on the fraction between maximum and minimum weight as follows:

$$(3.6) \quad \text{Updating the median requires at most } \left\lceil \frac{\max_i w_i}{\min_i w_i} \right\rceil \text{ pointer shifts.}$$

- (iii) Along with each shift operation, we update the median deviation, which is of constant runtime for each shift operation. Hence,

$$(3.7) \quad \text{Computing the deviation } d_{[\ell+1,r]} \text{ needs at most } \left\lceil \frac{\max_i w_i}{\min_i w_i} \right\rceil \text{ steps.}$$

3.3. The fast L^1 -Potts algorithm. We connect our data structure with the Potts core algorithm. Our L^1 -Potts algorithm is outlined in Algorithm 1.

ALGORITHM 1. MINIMIZATION OF THE L^1 -POTTS FUNCTIONAL.

input : Data vector $f \in \mathbb{R}^n$, weight vector $w \in \mathbb{R}_+^n$.
output: Minimzer u of the (weighted) L^1 -Potts functional
local : Left and right interval bounds $\ell, r \in \mathbb{N}$; Potts values $P \in \mathbb{R}^{n+1}$; temporary values $p, \mu, d \in \mathbb{R}$ (candidate Potts value, median, and median deviation); indexed linked histogram H ; array of right-most jumps $Z \in \mathbb{N}^n$; pointer to histogram node M ;

begin
 $P_0 \leftarrow -\gamma$;
 $H \leftarrow$ Empty indexed linked histogram;
for $r \leftarrow 1$ **to** n **do**
 $P_r \leftarrow \infty$; /* init candidate Potts value */
 Insert element f_r with weight w_r to histogram H ;
 Set temporary values and pointers of the histogram H to the permanent ones;
 $M \leftarrow$ Pointer to median node of histogram H ;
 $\mu \leftarrow$ value of median node M ; /* init the current median μ */
 $d \leftarrow \sum_{i=1}^r w_i |\mu - f_i|$; /* init the current median deviation d */
 for $\ell \leftarrow 1$ **to** r **do**
 $p \leftarrow P_{\ell-1} + \gamma + d$; /* compute candidate Potts value p */
 /* if p is lower than previous best Potts value P_r */
 if $p \leq P_r$ **then**
 $P_r \leftarrow p$; /* store new best Potts value */
 $Z_r \leftarrow \ell - 1$; /* update right-most jump */
 end
 Temporarily remove the weight w_ℓ from the node which the I_ℓ (cf. subsection 3.2) points to (if the weight of that node then equals zero, remove it from H);
 Shift the median pointer M toward the new median node of H and update the current median deviation d ;
 end
end
Reconstruct the minimizer u from the array of right-most jumps Z ;

end

Using the indexed linked histogram, we compute the deviations $d_{[\ell,r-1]}$ as follows. First, starting from a histogram for data $f_{[1,r-1]}$, we add data f_r as explained in the paragraph on “*Building the histogram*” in section 3.2. We obtain the deviation $d_{[1,r]}$. To obtain the deviations $d_{\ell,r}$, we successively remove the left-hand data items f_ℓ as described in the paragraph on *fast computation of median deviations* in section 3.2. This removal is done on the temporary structure of the histogram in order to save the histogram for data $f_{[1,r]}$ for the next update $r \rightarrow r + 1$.

Moreover, our data structure allows for in situ computation of the $d_{[\ell,r]}$ right at the place where they are needed such that no additional memory is required.

More precisely, each deviation $d_{[1,r]}, d_{[2,r]}, \dots, d_{[r,r]}$ produced in the inner ℓ -loop can immediately be used to compute the values $P_\gamma(u^{\ell-1}) + \gamma + \|f_{[\ell,r]} - (\mu_{[\ell,r]}, \dots, \mu_{[\ell,r]})\|_{\ell_w^1}$, ($\ell = 1, \dots, r$), cf. (3.4). After passing to the infimum over ℓ , we obtain a minimizer for the data $f_{[1,r]}$ according to (3.4), and the above deviations are not required any more.

For further details and implementational aspects, we refer to our MATLAB implementation, available for download at <http://pottslab.de>.

3.4. Theorem on time and space complexity. We consider the continuous Potts functional P_γ defined by (2.1) and its discretizations (P_γ^k) defined by (2.5). The piecewise constant functions associated with P_γ^k only jump in the finite sets X^k (cf. section 2), which allows us to identify such a piecewise constant function with a vector in euclidean space. Letting $w_i^{(k)} = \lambda(I_j^{(k)})$ be the length of the interval I_j (cf. (2.5)), the discretized Potts problem (2.5) reads

$$(3.8) \quad \gamma J(u) + \sum_i w_i |u_i - (S_k f)_i| = \gamma J(u) + \|u - S_k f\|_{\ell_w^1} \rightarrow \min,$$

which is a weighted discrete L^1 -Potts problem of type (3.1). If the weights behave well, we have the following statement on the complexity of the algorithm if the mesh size goes to 0.

THEOREM 3.3. *Consider a sequence of weight vectors $w^{(1)}, w^{(2)}, \dots$ where each weight vector $w^{(k)} = (w_1^{(k)}, \dots, w_{n_k}^{(k)})$ is of length n_k and the sequence n_k increases. We assume that*

$$(3.9) \quad \frac{\max_i w_i^{(k)}}{\min_i w_i^{(k)}} \leq C \quad \text{with } C \text{ independent of } k.$$

Then the proposed L^1 -Potts algorithm computes an exact minimizer of the discrete weighted L^1 -Potts functional within $O(n_k^2)$ time and $O(n_k)$ space as k increases.

In the case of equal weights, we obtain the following corollary.

COROLLARY 3.4. *The proposed L^1 -Potts algorithm computes an exact minimizer of the discrete L^1 -Potts functional (1.1) within $O(n^2)$ time and $O(n)$ space.*

Proof of Theorem 3.3. We begin with the time complexity. In the paragraph on the Potts core algorithm, we have seen that the Potts core algorithm needs $O(n^2)$ operations. Assumption (3.9) ensures that the medians $\mu_{[\ell,r]}$ in the inner ℓ -loop of Algorithm 1 can be updated within $O(1)$; cf. (3.7). The same is true for the deviations $d_{[\ell,r]}$ given by (3.3); cf. (3.6). This guarantees that the operations in the inner ℓ -loop are of constant complexity $O(1)$. All operations in the outer r -loop are of complexity $O(n)$ as explained in the paragraph ‘‘Building the histogram’’ in section 3.2. The final reconstruction step is of order $O(n \log n)$. (This can be further reduced to $O(n)$ if the medians computed in the main iteration are stored.) Hence, the overall time complexity is $O(n^2)$.

We come to space complexity. The only items depending on the length of input data in the data structure indexed linked histogram are the number of histogram nodes and the indexing array I . Both have at most length n . Furthermore, the arrays P and Z storing the Potts values $P_\gamma(u^r)$ and the right-most jump locations (see subsection 3.1) have length of order n . Summing up, Algorithm 1 has space complexity $O(n)$.

It remains to show the correctness of the proposed algorithm. For all partial data $f_{[1,r]}$, the algorithm computes a rightmost jump location $Z(r) = \ell^* - 1$ where ℓ^* is

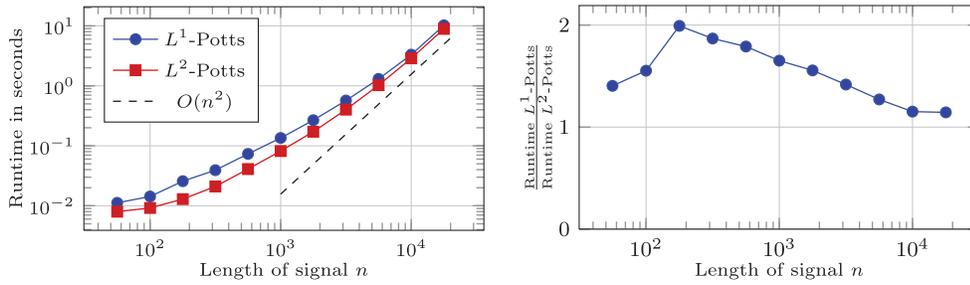


FIG. 3. Runtimes of the L^1 and L^2 -Potts algorithms on a logarithmic scale (left) and relative runtimes (right). For larger signals, the L^1 algorithm is almost as fast as the L^2 algorithm. The slower runtime for smaller signals is mainly due to overhead in the histogram structure. The experiments were conducted on an Apple MacBook Pro, with Intel Core 2 Duo 2.66 GHz and 8 GB RAM, using our implementation (<http://pottslab.de>).

a minimizing argument in (3.4). It follows from Lemma 3.1 that this $Z(r)$ is the jump location of a minimizer for data $f_{[1,r]}$. From these data $Z(r)$, $r = 1, \dots, n$, it produces a function which according to Lemma 3.2 is a minimizer of the L^1 Potts problem. \square

We give some comments. We note that our data structure allows in situ computation of the $d_{[\ell,r]}$, which makes our algorithm an $O(n)$ space algorithm. The asymptotic time and space complexity of our L^1 -Potts algorithm equals that of the L^2 -Potts algorithm proposed by Friedrich et al. [16]. Remarkably, our algorithm is not only asymptotically of comparable runtime. The actual runtime of the L^1 -Potts algorithm is only about 20% higher than that of the L^2 -Potts algorithm, for reasonably large signals; we refer to Figure 3 for a comparison of runtimes.

We next show that the condition (3.9) in Theorem 3.3 cannot be dropped. To this end, we consider data f_i and weights w_i given by the following table:

| | | | | | | |
|-------|-----|-----|-----|---------|-----|---------------------------------|
| i | 1 | 2 | 3 | 4 | ... | n |
| f_i | 1 | n | 2 | $n - 1$ | ... | $n/2$ |
| w_i | 1/2 | 1/4 | 1/8 | 1/16 | ... | 1/2 ^{$n+1$} |

We see that, for updating the median in the inner l -loop, our algorithm shifts the median pointer from the outer left side to the outer right side and vice versa, respectively. Thus the complexity of the median updating is of order n , and the algorithm is no longer of complexity $O(n^2)$.

3.5. Numerical experiments on the L^1 -Potts functional. We compare the L^1 -Potts functional, the L^2 -Potts functional and the L^1 -TV functional w.r.t. to their capabilities to reconstruct jump-sparse signals under different types of noise. For L^1 -TV minimization, we used the implementation of [10] with a maximum of 100 outer and 1000 inner iterations. In all experiments, we adjusted the model parameter γ so that the solutions were visually closest to the ground truth.

Our first experiment deals with additive Laplacian noise; see Figure 4. Every data point is corrupted by a random variable which is distributed according to the Laplacian probability density of variance σ^2

$$p(x) = \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}}{\sigma}|x|} \quad \text{with } \sigma > 0.$$

Our next experiment considers salt and pepper type noise; see Figure 5. Here, a

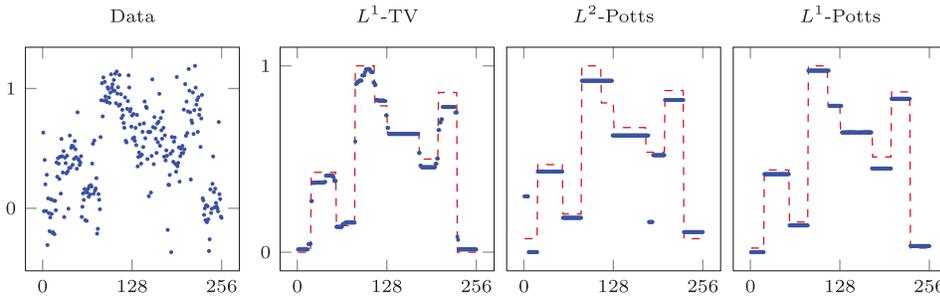


FIG. 4. Laplacian noise with $\sigma = 0.2$ (ground truth: dashed line). The L^1 -TV reconstruction exhibits staircasing effects. The L^2 -Potts reconstruction is not robust to strong outliers. The L^1 -Potts procedure recovers the original piecewise constant signal almost perfectly.

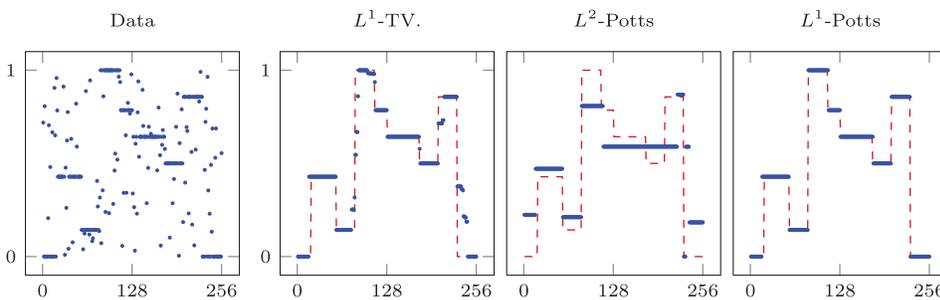


FIG. 5. A signal corrupted by salt and pepper noise: 40% of the data points are destroyed. The L^2 -Potts solution has few of the jumps of the original signal. The L^1 -TV reconstruction gets some plateaus, but several outliers remain. The L^1 -Potts solution yields an almost perfect reconstruction.

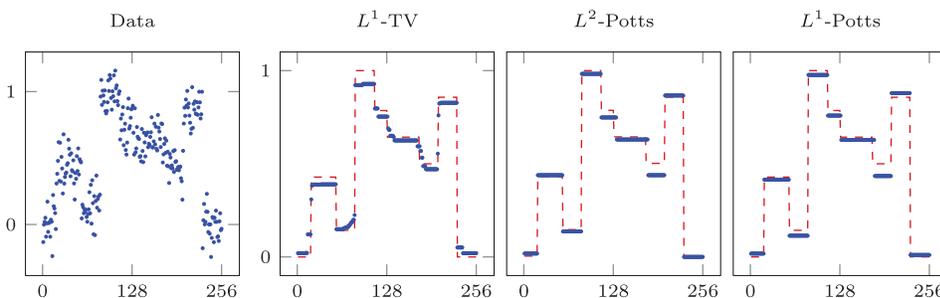


FIG. 6. In presence of Gaussian noise with $\sigma = 0.1$, both the L^1 - and the L^2 -Potts algorithm give almost perfect reconstructions. The L^1 -TV reconstruction does not find all plateaus.

fraction p of a signal g is set to a random variable η , i.e., the data are given by

$$f_i = g_i, \text{ with probability } 1 - p \quad \text{and} \quad f_i = \eta \text{ with probability } p.$$

In our experiments, η is distributed uniformly in the interval $[0, 1]$. Our last experiment compares the reconstruction methods under Gaussian noise of variance σ^2 , see Figure 6. Summing up, we observe that, in all three experiments, the minimizers of the L^1 -Potts functional approximate the ground truth very closely. For Gaussian noise, the L^2 and the L^1 -Potts perform equally well, whereas, for Laplacian and salt and pepper noise, L^1 Potts minimization turns out to be the method of choice.

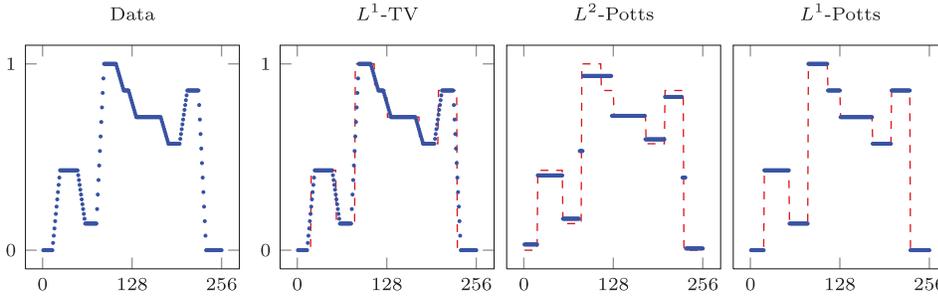


FIG. 7. Blurred data $f = K * g$, where K is the moving average of size 11. The L^1 -TV method reconstructs in the blurred signal itself. The L^2 -Potts method does not only not reconstruct all original plateaus, but even adds nonexistent ones. The minimizer of the L^1 -Potts recovers the piecewise constant signal g perfectly.

4. Robustness of the L^1 -Potts functional to mildly blurred data. In this section, we consider the L^1 -Potts functional (2.1) associated with blurred data $f = K * g$. Here, g denotes a piecewise constant function on $[0, 1]$ and K is a convolution kernel. Concerning K , we assume the following throughout this section.

Assumption 4.1. We assume that K is symmetric, compactly supported, non-negative with $K > 0$ in a neighborhood of the origin, and that it has total mass 1.

In order to properly define the convolution product $K * g$, we extend the piecewise constant function g to the left of 0 and to the right of 1 by its value on the leftmost and rightmost interval, respectively. (With minor technical modifications, our results are also valid for periodical extensions, zero padding, and extensions by reflection.)

From Figure 7 we see that neither L^1 -TV nor L^2 -Potts minimization are able to reconstruct such blurred data $f = K * g$. For L^1 -Potts minimization, however, Figure 7 suggests that it has some built-in blind deconvolution property: Without knowing the kernel K , the signal g is reconstructed from data $f = K * g$. We can even prove this blind deconvolution property: There are parameters $\gamma > 0$ and a maximal kernel size $\kappa > 0$ such that for all convolution kernels K with smaller support size

$$(4.1) \quad g = \operatorname{argmin}_{u \in \text{PC}[0,1]} \gamma J(u) + \|u - f\|_1, \quad \text{where } f = K * g.$$

This statement and the corresponding discretized version are the main results of this section; the precise formulations are Theorems 4.4 and 4.5. We note that for our results, the specific type of kernel is irrelevant, e.g., truncated Gaussian or indicator functions may be the underlying unknown kernel.

Furthermore, the experiments in Figure 1 suggest that the blind deconvolution property of the L^1 -Potts solutions even persists under noise. Here, the quality of the reconstructions barely depends on the type of noise.

In order to show (4.1), we first need a series of lemmas. In these lemmas, we throughout consider nonconstant piecewise constant functions g since, for constant signals g , (4.1) is obviously fulfilled. We denote the jump locations of the piecewise constant function g by $0 < j_1 < \dots < j_n < 1$ and let l_{\min} be the minimal interval length given by this partition; cf. Figure 8.

For positive $\kappa \leq \frac{l_{\min}}{6}$, we consider intervals J_i around the jump location j_i given by

$$(4.2) \quad J_i = [j_i - \kappa, j_i + \kappa].$$

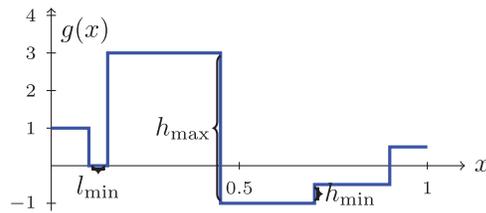


FIG. 8. Notation used in section 4. l_{\min} denotes the minimum interval length, h_{\max} the maximum jump height, and h_{\min} the minimum jump height of a piecewise constant function g .

We denote the complementary intervals N_i lying in between the jump locations by

$$(4.3) \quad N_i = (j_{i-1} + \kappa, j_i - \kappa).$$

On the left boundary, we set $N_1 = [0, j_1)$ and on the right boundary $N_{n+1} = (j_n + \kappa, 1]$.

LEMMA 4.1. *For any $u_0 \in \text{PC}[0, 1]$, there is $u_1 \in \text{PC}[0, 1]$ such that u_1 has no jump in the union $\bigcup_i N_i$ and such that $P_\gamma(u_1) \leq P_\gamma(u_0)$.*

Proof. Consider an arbitrary index i and assume that u_0 jumps in N_i . We first consider the case where u_0 has more than one jump in N_i . In this case, we define $u_1 = f$ on N_i and $u_1 = u_0$ elsewhere. Since $f = g$ is constant on N_i , we have $J(u_1) \leq J(u_0)$ and $\|f - u_1\|_1 \leq \|f - u_0\|_1$. Hence, $P_\gamma(u_1) \leq P_\gamma(u_0)$, which shows the lemma in this case.

It remains to prove the case where u_0 has exactly one jump in N_i , say, at $x \in N_i = (l, r)$. To this end, we compare the value $d_l = |f - u_0| = |g - u_0|$ on (l, x) with the value $d_r = |f - u_0| = |g - u_0|$ on (x, r) . If $d_l \geq d_r$, we shift the jump position to the left; that means we let $u_1 = u_0(x^+)$ on N_i and $u_1 = u_0$ elsewhere. The number of jumps does not increase, $|f - u_1| \leq |f - u_0|$ on N_i , and thus

$$\|f - u_1\|_1 = \|(f - u_0)|_{N_i^c}\|_1 + \|(f - u_1)|_{N_i}\|_1 \leq \|f - u_0\|_1.$$

If $d_l < d_r$, we define $u_1 = u_0(x^-)$ on N_i and $u_1 = u_0$ elsewhere. Then, an analogous argument completes the proof. \square

We use the notation h_{\max} and h_{\min} for the maximal and minimal jump height of g , respectively.

LEMMA 4.2. *Let $u_1 \in \text{PC}[0, 1]$ with no jumps in $\bigcup_i N_i$. If*

$$(4.4) \quad 6h_{\max}\kappa \leq \frac{1}{2}h_{\min}(l_{\min} - 2\kappa) - \gamma,$$

then there is $u_2 \in \text{PC}[0, 1]$ such that $P_\gamma(u_2) \leq P_\gamma(u_1)$ and such that u_2 has at least one jump in each J_i .

Proof. Assume that u_1 has no jump in J_{i_0} for some i_0 . Then we find integers $k \geq 0$ and $s \geq 1$ such that u_1 is constant on $N_{i_0-k} \cup J_{i_0-k} \cup \dots \cup J_{i_0+s-1} \cup N_{i_0+s}$, such that u_1 jumps in J_{i_0-k-1} or N_{i_0-k} is the leftmost interval and such that u_1 jumps in J_{i_0+s} or N_{i_0+s} is the rightmost interval. We denote by x_l the rightmost jump point of u_1 in J_{i_0-k-1} (or by 0 if N_{i_0-k} is the leftmost interval) and by x_r the leftmost jump point of u_1 in J_{i_0+s} (or by 1 if N_{i_0+s} is the rightmost interval). We define u_2 by $u_2 = g$ on (x_l, x_r) and elsewhere by $u_2 = u_1$. Obviously, u_2 has at least one jump in each J_i for $i = i_0 - k, \dots, i_0 + s - 1$. Furthermore, we have

$$(4.5) \quad J(u_2) \leq J(u_1) + k + s.$$

Now we estimate the deviation of u_2 to $f = K * g$ on (x_l, x_r) to obtain

$$(4.6) \quad \begin{aligned} \|(u_2 - K * g)|_{(x_l, x_r)}\|_1 &= \|(g - K * g)|_{(x_l, x_r)}\|_1 \\ &\leq (k + s + 2)h_{\max}2\kappa \leq 6(k + s)h_{\max}\kappa. \end{aligned}$$

This is because g equals $K * g$ on each N_i and $\|(g - K * g)|_{J_i}\|_1$ is bounded by the jump height of g in J_i times the length of J_i , which in turn equals 2κ by definition.

In order to estimate the deviation of u_1 to $K * g$ on (x_l, x_r) , we need the following preparation. Let us consider a data sequence g_0, \dots, g_n and an arbitrary $y \in \mathbb{R}$. Using the triangle inequality, we have, for two consecutive members g_j and g_{j+1} , that $|g_j - y| + |g_{j+1} - y| \geq \min_i |g_i - g_{i+1}|$. This implies

$$(4.7) \quad \sum_{i=0}^n |g_i - y| \geq \frac{n}{2} \min_i |g_i - g_{i+1}|.$$

Now we come back to the estimate of $\|(u_1 - K * g)|_{(x_l, x_r)}\|_1$. Recalling that u_1 is constant on (x_l, x_r) (with possibly $x_l = 0$ and $x_r = 1$ as explained above) and that $f = K * g$ equals g on each N_i , we see that

$$(4.8) \quad \|(u_1 - K * g)|_{(x_l, x_r)}\|_1 \geq \sum_{i=i_0-k}^{i_0+s} \|(u_1 - K * g)|_{N_i}\|_1 \geq \min_i |N_i| \sum_{i=i_0-k}^{i_0+s} |y - g_i|,$$

where y is the function value of u_1 on (x_l, x_r) and g_i is the function value of $g = K * g$ on N_i , where g is constant on. We employ (4.7) to get

$$(4.9) \quad \sum_{i=i_0-k}^{i_0+s} |y - g_i| \geq \frac{1}{2}(k + s) \min_i |g_i - g_{i+1}| \geq \frac{1}{2}(k + s) h_{\min}.$$

Plugging (4.9) into (4.8) and using $|N_i| \geq l_{\min} - 2\kappa$, we get

$$(4.10) \quad \|(u_1 - K * g)|_{(x_l, x_r)}\|_1 \geq (l_{\min} - 2\kappa) \frac{k+s}{2} h_{\min}.$$

Next, we set up a lower bound for $\|u_2 - K * f\|_1$ in terms of $\|u_1 - K * f\|_1$. To this end, we first restrict to the interval (x_l, x_r) and obtain

$$(4.11) \quad \begin{aligned} \|(u_2 - K * g)|_{(x_l, x_r)}\|_1 &\leq (k + s)6h_{\max}\kappa \\ &\leq (k + s) \left(\frac{1}{2}h_{\min}(l_{\min} - 2\kappa) - \gamma\right) \\ &\leq \|(u_1 - K * g)|_{(x_l, x_r)}\|_1 - (k + s)\gamma. \end{aligned}$$

For the first inequality we used (4.6), for the second inequality (4.4), and for the last inequality (4.10). Since u_2 equals u_1 outside (x_l, x_r) , inequality (4.11) implies that

$$\begin{aligned} \|u_2 - K * g\|_1 &= \|(u_1 - K * g)|_{[0,1] \setminus (x_l, x_r)}\|_1 + \|(u_2 - K * g)|_{(x_l, x_r)}\|_1 \\ &\leq \|(u_1 - K * g)|_{[0,1] \setminus (x_l, x_r)}\|_1 + \|(u_1 - K * g)|_{(x_l, x_r)}\|_1 - (k + s)\gamma \\ &= \|(u_1 - K * g)\|_1 - (k + s)\gamma. \end{aligned}$$

Therefore, using (4.5), we have

$$\begin{aligned} P_\gamma(u_2) &= J(u_2) + \|(u_2 - K * g)\|_1 \\ &\leq J(u_1) + \gamma(k + s) + \|(u_1 - K * g)\|_1 - \gamma(k + s) = P_\gamma(u_1). \end{aligned}$$

So far, we have shown how to modify u_1 such that it jumps in each of the intervals $J_{i_0-k}, \dots, J_{i_0+s-1}$ without increasing P_γ . The procedure may cancel existing jumps in J_{i_0-k-1} and J_{i_0+s} (which only happens if $u_1(x_l^-) = g(x_l^+)$ or $u_1(x_r^+) = g(x_r^-)$). If there is only one jump in J_{i_0-k-1} or J_{i_0+s} and this jump is canceled, i.e., u_2 has no jump there, we apply the procedure of this lemma a second time, this time to u_2 for the initial interval J_{i_0-k-1} or J_{i_0+s} . Then the resulting u'_2 has no jump in J_{i_0-k-1} or J_{i_0+s} . Cancellation in J_{i_0-k} or J_{i_0+s-1} cannot occur for u'_2 since $u_2(x_l^-) = g(x_l^-) \neq g(x_l^+)$ or $u_2(x_r^+) = g(x_r^+) \neq g(x_r^-)$ this time. Summing up, we found a way to introduce jumps on $J_{i_0-k}, \dots, J_{i_0+s-1}$ keeping at least one jump in the adjacent intervals J_{i_0-k-1} and J_{i_0+s} while not increasing P_γ . \square

LEMMA 4.3. *Let $u_2 \in \text{PC}[0, 1]$ and assume (4.4) and $\kappa \leq \frac{1}{6}l_{\min}$. Then there is $u_3 \in \text{PC}[0, 1]$ such that u_3 has no jumps in $\bigcup_i N_i$ and precisely one jump in each J_i , and*

$$P_\gamma(u_3) \leq P_\gamma(u_2) \quad \text{if } 2\kappa h_{\max} \leq \gamma.$$

Proof. By Lemma 4.1, we may assume that u_2 has no jump in $\bigcup_i N_i$ and, by Lemma 4.2, that u_2 has at least one jump in each J_i .

We want to find u' such that u' agrees with g on N_i for all i and such that $P_\gamma(u') \leq P_\gamma(u_2)$. To this end, we denote by l_i the rightmost jump point of u_2 in J_{i-1} and by r_i the leftmost jump point of u_2 in J_i . We define u' by $u' = g_i$ on (l_i, r_i) and by $u' = u_2$ elsewhere. Here, g_i is the value of g on N_i . The number of jumps of u' does not exceed that of u_2 . We first observe that, since $|N_i| \geq l_{\min} - 2\kappa$,

$$\|(g - u_2)|_{N_i}\|_1 \geq (l_{\min} - 2\kappa)|g_i - (u_2)_i|,$$

where $(u_2)_i$ is the value of u_2 on N_i . Using this estimate and noting that $\|(g - u')|_{N_i}\|_1 = 0$, we obtain

$$\begin{aligned} \|(g - u')|_{(l_i, r_i)}\|_1 &= \|(g - u')|_{(l_i, r_i) \setminus N_i}\|_1 \\ (4.12) \quad &\leq \|(g - u_2)|_{(l_i, r_i) \setminus N_i}\|_1 + \|(u_2 - u')|_{(l_i, r_i) \setminus N_i}\|_1 \\ &\quad + \|(g - u_2)|_{N_i}\|_1 - (l_{\min} - 2\kappa)|g_i - (u_2)_i|. \end{aligned}$$

Since $u' = g_i$ on (l_i, r_i) and $|(l_i, r_i) \setminus N_i| \leq 4\kappa$, we have

$$\|(u_2 - u')|_{(l_i, r_i) \setminus N_i}\|_1 \leq 4\kappa|g_i - (u_2)_i|.$$

Plugging this into (4.12), we get

$$\|(g - u')|_{(l_i, r_i)}\|_1 \leq \|(g - u_2)|_{(l_i, r_i)}\|_1 + (6\kappa - l_{\min})|g_i - (u_2)_i| \leq \|(g - u_2)|_{(l_i, r_i)}\|_1,$$

since an assumption of the lemma was that $\kappa \leq \frac{l_{\min}}{6}$. Therefore, $P_\gamma(u') \leq P_\gamma(u_2)$. Proceeding likewise for all intervals N_i (where we let $l_i = 0$ and $r_i = 1$ for the boundary intervals), we get u' with the desired property.

Now we define the function u_3 and show that its Potts-value $P_\gamma(u_3) \leq P_\gamma(u')$. To that end, we consider the k intervals $J_i = [l_i, r_i]$ where the u' has more than one jump and define u_3 as follows: we set $u_3 = g(l_i^-)$ on $(l_i, \frac{r_i+l_i}{2})$, $u_3 = g(r_i^+)$ on $(\frac{r_i+l_i}{2}, r_i)$, and $u_3 = u'$ elsewhere. Then, $J(u_3) \leq J(u') - k$. Furthermore, on each such interval $J = [l, r]$, where u' has more than one jump, we have that

$$(4.13) \quad \|(K * g - u_3)|_J\|_1 \leq \|(K * g - u')|_J\|_1 + 2\kappa h_{\max}.$$

This is because $|K * g - u_3| \leq h_{\max}$ on J since both functions $K * g$ and u_3 take their values in the interval $[g(l_i^-), g(r_i^+)]$ whose length does not exceed h_{\max} . Summing over all the k intervals where u' jumps more than once, we get using (4.13)

$$\begin{aligned} P_\gamma(u_3) &= \gamma J(u_3) + \|(K * g - u_3)\|_1 \\ &\leq \gamma(J(u') - k) + \|(K * g - u')\|_1 + k \cdot 2\kappa h_{\max} \\ &= P_\gamma(u') + k(2\kappa h_{\max} - \gamma) \leq P_\gamma(u'), \end{aligned}$$

since by hypothesis $2\kappa h_{\max} \leq \gamma$. Summing up, we have found a function u_3 with $P_\gamma(u_3) \leq P_\gamma(u') \leq P_\gamma(u_2)$, which has exactly one jump in each interval J_i . \square

Recall that the minimal and maximal jump height of g are denoted by h_{\min} and h_{\max} , respectively. Furthermore, l_{\min} is the minimal interval length between two jumps of g ; see Figure 8.

THEOREM 4.4. *Let g be a piecewise constant function on $[0, 1]$ and $f = K * g$ with K as in Assumption 4.1. If the support size κ of the convolution kernel K satisfies*

$$(4.14) \quad \kappa \leq \frac{h_{\min} l_{\min}}{2(8h_{\max} + h_{\min})},$$

then g is the unique minimizer of the Potts functional P_γ associated with f , i.e.,

$$(4.15) \quad P_\gamma(g) = \inf_{u \in \text{PC}[0,1]} P_\gamma(u) = \inf_{u \in \text{PC}[0,1]} \gamma J(u) + \|u - f\|_1$$

for any Potts parameter γ satisfying

$$(4.16) \quad 2\kappa h_{\max} \leq \gamma \leq \frac{1}{2} h_{\min} l_{\min} - (h_{\min} + 6h_{\max}) \kappa.$$

Proof. If g is constant, $f = g$ for any kernel of any size and the assertion of the theorem is obviously true. So we may assume that g has at least one jump.

We first notice that (4.14) implies $\kappa \leq \frac{1}{6} l_{\min}$, which we assumed at the beginning of this section. By (4.14), we get

$$2\kappa h_{\max} \leq \frac{1}{2} h_{\min} l_{\min} - (h_{\min} + 6h_{\max}) \kappa,$$

which allows us to choose γ according to (4.16). Manipulating the right-hand inequality of (4.16) yields that

$$6h_{\max} \kappa \leq \frac{1}{2} h_{\min} (l_{\min} - 2\kappa) - \gamma.$$

This is precisely the assumption on κ and γ in Lemma 4.2. Furthermore, the left-hand inequality of (4.16) ensures that Lemma 4.3 is applicable. Now Lemma 4.1 allows us to search the minimizer of P_γ , cf. (4.15), in the class of functions of piecewise constant functions without jumps in $\bigcup_i N_i$. Then, by Lemmas 4.2 and 4.3, we may restrict this class further to those functions having precisely one jump in each interval J_i , i.e.,

$$(4.17) \quad \inf_{u \in \text{PC}} P_\gamma(u) = \inf \{P_\gamma(u) : u \text{ jumps in no } N_i \text{ and precisely once in each } J_i\}.$$

Now we consider such a function u which is a member of the set defined by the right-hand side of (4.17) and assume that u does not equal g identically. Let us denote the jump locations of u by x_1, \dots, x_n with $x_i \in J_i$. Without loss of generality, we may assume that u coincides with g on each N_i , since otherwise, redefining $u = g_i$ on

(x_i, x_{i+1}) (and probably also on the boundary intervals) yields a smaller P_γ value (cf. beginning of the proof of Lemma 4.3).

Since u does not equal g , we find at least one interval J_i where the jump locations j_i of g and the corresponding jump location x_i of u do not coincide. We show that

$$(4.18) \quad \|(u - K * g)|_{J_i}\|_1 > \|(g - K * g)|_{J_i}\|_1$$

on each J_i where the jumps of g and u do not coincide. This implies, as g and u have the same number of jumps, that $P_\gamma(u) > P_\gamma(g)$.

To complete the proof, it remains to show (4.18). To this end, we denote the left-hand and right-hand boundary of J_i by l and r , respectively. We note that $g(l) = u(l^-)$, $g(r) = u(r^+)$, and g jumps at $\frac{1}{2}(l+r) = j_i$. Recall that, by assumption, the jump location x_i of u fulfills $x_i \neq j_i$. The symmetry of K implies that the mass of K on the negative half axis equals that on the positive half axis, which in turn yields

$$(4.19) \quad K * g(\tfrac{1}{2}(l+r)) = \tfrac{1}{2}(g(r) + g(l)).$$

Since K is positive, $K * g$ is monotone on J_i . Together with (4.19), this implies

$$(4.20) \quad \|(g - K * g)|_J\|_1 \leq \tfrac{1}{2}|g(r) - g(l)| \cdot |J| \quad \text{for any interval } J \subseteq J_i.$$

Furthermore, as we assume that $K > 0$ in a neighborhood of the origin (cf. Assumption 4.1), $K * g$ is strictly monotone in a neighborhood of j_i . This shows that inequality (4.20) is strict, i.e., we may replace the symbol “ \leq ” in (4.20) by “ $<$.” On the interval I' which is given by the endpoints x_i and j_i , it holds that $|u(x) - K * g(x)| > \frac{1}{2}|g(r) - g(l)|$ for almost all x . This follows from inspection of all possible cases: if $u(l) = g(l) < g(r) = u(r)$ and $x_i < j_i$, then $K * g < \frac{1}{2}(g(r) + g(l))$ on I' and $u = g(r)$ on I' ; hence, $u - K * g > \frac{1}{2}(g(r) - g(l))$ on I' . The other three cases are analogous. Summing up,

$$(4.21) \quad \|(K * g - u)|_{I'}\|_1 > \tfrac{1}{2}|g(r) - g(l)| \cdot |I'| > \|(K * g - g)|_{I'}\|_1.$$

As assumed above, u equals g on all N_i , and both u and g jump precisely once in each J_i . Then (4.21) implies that u equals g on $[0, 1]$ for the following reason. Otherwise, there were J_i and a (proper) subinterval $I' \subset J_i$ (with positive Lebesgue measure) such that (4.21) holds true. This would imply (4.18), which, in turn, would contradict u being a minimizer. Hence, any minimizer equals g which completes the proof. \square

The next statement is the discrete version of Theorem 4.4.

THEOREM 4.5. *Let g be a piecewise constant function on $[0, 1]$ and let κ be the support size of the convolution kernel K fulfilling Assumption 4.1. We recall that P_γ^k is the discrete Potts functional given by (2.5) and is associated with the discretization set $X^k = \{x_1, \dots, x_n\}$ of mesh size $\eta = \max_i |x_i - x_{i-1}|$. We assume that*

$$(4.22) \quad \kappa + \eta \leq \frac{h_{\min} l_{\min}}{2(8h_{\max} + h_{\min})}.$$

*Let $g^k = \sum_i \alpha_i \mathbf{1}_{(s_i, s_{i+1})}$ be a “ X^k -discretization” of $g = \sum_i \alpha_i \mathbf{1}_{(t_i, t_{i+1})}$, where $s_i \in X^k$ is a nearest neighbor of the jump location t_i of g . Furthermore, we consider data $f^k = \sum_j S_k f(j) \cdot \mathbf{1}_{I_j}$ obtained from sampling $f = K * g$ on the discretization set X^k using*

(2.3) or (2.4) (with midpoint sampling). Then each function g^k is a minimizer of the discrete Potts functional P_γ^k associated with data f^k , i.e.,

$$(4.23) \quad P_\gamma^k(g^k) = \inf_{u \in \text{PC}^k[0,1]} P_\gamma^k(u) = \inf_{u \in \text{PC}^k[0,1]} \gamma J(u) + \|u - f^k\|_1$$

for any Potts parameter γ satisfying

$$(4.24) \quad 2(\kappa + \eta)h_{\max} \leq \gamma \leq \frac{1}{2}h_{\min}l_{\min} - (h_{\min} + 6h_{\max})(\kappa + \eta).$$

The X^k -discretizations g^k of the above type are the only minimizers; hence the minimizer is unique whenever each jump location t_i of g has a unique nearest neighbor in X^k . Furthermore, if (X^k) is a nested sequence of discretization sets being eventually dense, each sequence of minimizers (g^k) converges to g .

Proof. The proof of Theorem 4.5 uses arguments analogous to those in the proof of Theorem 4.4. In particular, the statements and proofs of Lemmas 4.1, 4.2, and 4.3 remain true in the setup of Theorem 4.5 if we replace each occurrence of κ by $\kappa + \eta$ in these lemmas, their proofs, and the definition of N_i and J_i in (4.2) and (4.3), respectively. For the rest of this proof, we define N_i and J_i accordingly with $\kappa + \eta$ replacing κ . Therefore, it is sufficient to show (4.23) for the functions u in $\text{PC}^k[0, 1]$, which jump in no N_i and precisely once in each J_i . Thus, let us consider a function u , which we may assume, as in the proof of Theorem 4.4, to coincide with g on each N_i . We denote the jump location of u in J_i by z_i and the jump location of g in J_i by t_i . We use the notation S_i for the set of nearest neighbors of t_i in the k th level discretization set X^k (which lie in J_i since we used $\kappa + \eta$ in the definition of J_i at the beginning of this proof). We consider intervals J_i , where $z_i \notin S_i$, and show that

$$(4.25) \quad \|(u - f^k)|_{J_i}\|_1 > \|(g^k - f^k)|_{J_i}\|_1.$$

Furthermore, we consider an additional X^k -discretization \tilde{g}^k (with the same properties as g^k but possibly different jump locations in S_i) and show that, on each J_i ,

$$(4.26) \quad \|(g^k - f^k)|_{J_i}\|_1 = \|(\tilde{g}^k - f^k)|_{J_i}\|_1.$$

Since the number of jumps of u , g^k , and \tilde{g}^k are equal, (4.25) and (4.26) yield (4.23).

In order to show (4.25) and (4.26), we first note that, by the proof of Theorem 4.4, $f(t_i) = K * g(t_i) = \frac{1}{2}(g(t_i^-) + g(t_i^+)) =: a$ and f is monotone and strictly monotone in a neighborhood of t_i . So, if t_i has two nearest neighbors l, r in S_i , then the symmetry of K implies that f^k equals $a = \frac{1}{2}(g(t_i^-) + g(t_i^+))$ on the interval (l, r) . This is because $K * g(t_i - x) + K * g(t_i + x) = 2a$ on (l, r) and thus both the integral sampling (2.3) and the sampling (2.4) using midpoints yield that f^k equals a on (l, r) . Furthermore, f^k is monotone on J_i since so is f . (We recall that, in the beginning of the proof, we defined J_i with $\kappa + \eta$ replacing κ in (4.2).) Hence, $|g^k - f^k| \leq b := \frac{1}{2}|g(t_i^-) - g(t_i^+)|$. Also, f^k jumps at l and r since f is strictly monotone in a neighborhood of t_i . Hence, if the jump point z_i of u is not in $S_i = \{l, r\}$, then $|u - f^k| > b$ (a.e.) on the interval B connecting z_i and n , where $n \in \text{argmin}_{\nu \in \{l, r\}} |\nu - z_i|$. This follows from an argument analogous to the differentiation of cases preceding (4.21). Since u equals g^k on $J_i \setminus B$, we have shown (4.25) when t_i has two nearest neighbors. Since g^k and \tilde{g}^k jump at l or r , we have that $|g^k - f^k| = b = |\tilde{g}^k - f^k|$ on (l, r) . Since g^k and \tilde{g}^k are equal on $J_i \setminus (l, r)$, we get (4.26).

If t_i has only one nearest neighbor s_i in X^k , then on the one side (left or right) of s_i , we have $f^k > a$, and on the other side (right or left) of s_i , we have $f^k < a$. To see this, denote the other neighbor (which is not a nearest neighbor) of t_i in X^k by o_i . As seen in the proof of Theorem 4.4, f is monotone and strictly monotone in a neighborhood of t_i . Then, the midpoint m_i of s_i and o_i does not equal t_i and hence $f^k(t_i) \neq a$ by the strict monotonicity of f near t_i for both kind of sampling operators (2.3) and (2.4) with midpoint sampling. Furthermore, $f^k(t_i) = f^k(m_i) > a$ if $f(s_i) < a$ since s_i is closer to t_i than o_i . Analogously, $f^k(t_i) < a$ if $f(s_i) > a$. (Note that $f(s_i) = a$ does not happen, since we consider the case where s_i is unique.) Then, if $f(s_i) < a$, we have $f^k < a$ on that side of s_i where t_i does not lie, and analogously (with inverse inequality) for the case $f(s_i) > a$. Now, we proceed as in the case of nonunique nearest neighbors above. We assume that the jump point z_i on J_i of u does not equal s_i . Then we denote the interval connecting z_i and s_i by B . On B , we have $|u - f^k| > b$ (a.e.) on the interval B connecting z_i and s_i . Since $g^k = u$ on $J_i \setminus B$ and since $|g^k - f^k| \leq b$ on J_i , we have shown (4.25) in the case where t_i has only one nearest neighbor. The last assertion of the theorem is a consequence of the fact that the mesh sizes of (X^k) tend to 0 as $k \rightarrow \infty$. \square

We note that, although the statement of the theorem holds for both kinds of sampling operators introduced in (2.3) and (2.4) with midpoint sampling, it is more natural to consider point sampling here since averaging might be already modeled by the kernel K .

5. Reconstruction of highly blurred jump-sparse signals. In the previous section we have seen that L^1 -Potts minimization is robust to mildly blurred data. In order to deal with severely blurred data, we propose a heuristic splitting approach to the deconvolution problem (1.2). In contrast to the previous section, we need to know the kernel K . We rewrite (1.2) as the constrained optimization problem in two variables

$$(5.1) \quad \min_{u,v} \gamma J(u) + \|K * v - f\|_1, \quad \text{subject to } u - v = 0.$$

Then we add the constraint $u - v$ as the extra penalty $\mu \|u - v\|_1$ to the target functional to obtain the unconstrained optimization problem

$$(5.2) \quad \gamma J(u) + \mu \|u - v\|_1 + \|K * v - f\|_1 \rightarrow \min.$$

The extra parameter $\mu > 0$ controls how strongly u and v are tied to each other. We approach the bivariate problem (5.2) by the following heuristic. We first fix v and minimize w.r.t. the variable u ; then we fix u and optimize w.r.t. v . This yields the two univariate problems

$$(5.3) \quad \min_u \frac{\gamma}{\mu} J(u) + \|u - v\|_1 \quad \text{and} \quad \min_w \mu \|w\|_1 + \|K * w - (K * u - f)\|_1,$$

where $w = u - v$. The first problem is a L^1 -Potts minimization problem for which we apply the fast algorithm developed in this paper. The second problem is a convex optimization problem which we solve by using the corresponding algorithm in [1]; alternatives are [17, 9]. Summing up, we alternately minimize the two problems in (5.3), where we successively increase the coupling parameter μ (by a factor of 1.5) during the iteration. In contrast to the rest of the paper, no convergence analysis for this method is carried out here.

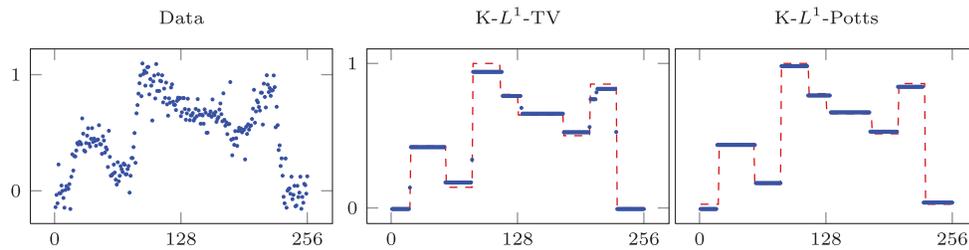


FIG. 9. Signal blurred by a (normalized) Gaussian kernel and corrupted by Laplacian noise with $\sigma = 0.1$. The minimizer of the L^1 -TV deconvolution functional has additional jumps and less contrast at the large jumps. The proposed L^1 -Potts deconvolution strategy reconstructs the original signal almost perfectly.

Numerical experiments comparing the results of the proposed splitting algorithm for the L^1 -Potts deconvolution problem (1.2) with the solutions of the L^1 -TV problem

$$(5.4) \quad \gamma \|\nabla u\|_1 + \|K * u - f\|_1 \rightarrow \min$$

can be found in Figure 9. To solve (5.4), we used the algorithm proposed in [10]. We observe that the computed minimizer of the L^1 -TV functional still contains noise, whereas the solution of our approach recovers the piecewise constant signal almost perfectly.

6. Conclusion and future research. We have seen that the L^1 -Potts functional is particularly well suited to reconstruct piecewise constant functions under non-Gaussian noise. We established a continuous L^1 -Potts model with the feature that the limits of minimizers of its (possibly nonequidistant) discretizations are minimizers of the continuous model. For the solution of these discrete L^1 -Potts functionals, we have introduced an exact and fast algorithm.

We have shown that the L^1 -Potts functional is robust to mild blurring. For strongly blurred signals and known source of blurring, we have proposed an algorithm to solve the associated L^1 -Potts deconvolution problem. One direction of future research is to obtain analytic results for this heuristic approach.

Another direction of future research is the application of our computational strategies to higher dimensional problems. Minimizing the direct analogue of the Potts functional in higher dimensions is an NP-hard problem [2]. However, the computational load may be reduced by restricting the set of admissible partitions. One approach in that direction are wedgelets [12]. In the L^2 setting, a fast wedgelet algorithm has been obtained in [15]. Building on the ideas for the fast computations of medians in this paper, the authors were able to derive a fast algorithm for the computation of L^1 -wedgelets. Its computational complexity lies in the order of $O(n \log n)$, where n is the number of pixels. This result is the matter of a forthcoming paper. Here, a related open problem is to find fast algorithms for wider classes of admissible partitions.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which helped to improve the presentation of the paper.

REFERENCES

- [1] J.C. BARITAUX AND M. UNSER, *A primal-dual reconstruction algorithm for fluorescence and bioluminescence tomography*, in Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011, pp. 960–963.
- [2] Y. BOYKOV, O. VEKSLER, AND R. ZABIH, *Fast approximate energy minimization via graph cuts*, IEEE Trans. Pattern Anal. Mach. Intell., 23 (2001), pp. 1222–1239.
- [3] L. BOYSEN, S. BRUNS, AND A. MUNK, *Jump estimation in inverse regression*, Electron. J. Stat., 3 (2009), pp. 1322–1359.
- [4] L. BOYSEN, A. KEMPE, V. LIEBSCHER, A. MUNK, AND O. WITTICH, *Consistencies and rates of convergence of jump-penalized least squares estimators*, Ann. Statist., 37 (2009), pp. 157–183.
- [5] L. BOYSEN, V. LIEBSCHER, A. MUNK, AND O. WITTICH, *Scale space consistency of piecewise constant least squares estimators: Another look at the regressogram*, IMS Lecture Notes Monogr. Ser. 55, 2007, pp. 65–84.
- [6] A. BRAIDES, *Gamma-Convergence for Beginners*, Oxford University Press, Oxford, 2002.
- [7] E.J. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.
- [8] A. CHAMBOLLE, *Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations*, SIAM J. Appl. Math., 55 (1995), pp. 827–863.
- [9] 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.
- [10] C. CLASON, B. JIN, AND K. KUNISCH, *A duality-based splitting method for ℓ^1 -TV image restoration with automatic regularization parameter choice*, SIAM J. Sci. Comput., 32 (2009), pp. 1484–1505.
- [11] R.A. DEVORE AND G.G. LORENTZ, *Constructive Approximation*, Springer, New York, 1993.
- [12] D.L. DONOHO, *Wedgelets: Nearly minimax estimation of edges*, Ann. Statist., 27 (1999), pp. 859–897.
- [13] A.L. DROBYSHEV, C. MACHKA, M. HORSCH, M. SELTMANN, V. LIEBSCHER, M.H. DE ANGELIS, AND J. BECKERS, *Specificity assessment from fractionation experiments (safe): A novel method to evaluate microarray probe specificity based on hybridisation stringencies*, Nucleic Acids Research, 31 (2003), pp. 1–10.
- [14] M. ELAD, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, 2010.
- [15] F. FRIEDRICH, L. DEMARET, H. FÜHR, AND K. WICKER, *Efficient moment computation over polygonal domains with an application to rapid wedgelet approximation*, SIAM J. Sci. Comput., 29 (2008), pp. 842–863.
- [16] F. FRIEDRICH, A. KEMPE, V. LIEBSCHER, AND G. WINKLER, *Complexity penalized M -Estimation*, J. Comput. Graph. Statist., 17 (2008), pp. 201–224.
- [17] H. FU, M. NG, M. NIKOLOVA, AND J. BARLOW, *Efficient minimization methods of mixed ℓ^1 - ℓ^1 and ℓ^2 - ℓ^1 norms for image restoration*, SIAM J. Sci. Comput., 27 (2006), pp. 89–97.
- [18] S. GEMAN AND D. GEMAN, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Mach. Intell., 6 (1984), pp. 721–741.
- [19] P. HUPÉ, N. STRANSKY, J.P. THIERY, F. RADVANYI, AND E. BARILLOT, *Analysis of array CGH data: From signal ratio to gain and loss of DNA regions*, Bioinformatics, 20 (2004), pp. 3413–3422.
- [20] E. ISING, *Beitrag zur Theorie des Ferromagnetismus*, Zeitschrift für Physik A: Hadrons and Nuclei, 31 (1925), pp. 253–258.
- [21] V. LIEBSCHER, *Geiz in der Statistik*, Lecture at the “Festkolloquium” in Honour of G. Winkler’s 65th Birthday, 2011.
- [22] R. MARCH AND P. BARONE, *Reconstruction of a piecewise constant function from noisy Fourier coefficients by Padé method*, SIAM J. Appl. Math., 60 (2000), pp. 1137–1156.
- [23] D. MUMFORD AND J. SHAH, *Boundary detection by minimizing functionals*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 17, 1985, pp. 137–154.
- [24] M. NIKOLOVA, *Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers*, SIAM J. Numer. Anal., 40 (2003), pp. 965–994.
- [25] R.B. POTTS, *Some generalized order-disorder transformations*, Math. Proc. Cambridge Philos. Soc., 48 (1952), pp. 106–109.
- [26] R. SCHÖNMEYER, G. SCHMIDT, S. MEDING, A. WALCH, AND G. BINNIG, *Automated co-analysis of MALDI- and H&E-images of retinal tissue for an improved spatial MALDI resolution*, in Proceedings of the 53rd Symposium of the Society for Histochemistry, 2011, pp. 54–55.

- [27] M. STORATH, A. WEINMANN, AND L. DEMARET, *Jump-sparse and sparse recovery using Potts functionals*, IEEE Trans. Signal Process., 62 (2014), pp. 3654–3666.
- [28] G. WINKLER, O. WITTICH, V. LIEBSCHER, AND A. KEMPE, *Don't shed tears over breaks*, Jahrbuch. Dtsch. Math.-Ver., 107 (2005), pp. 57–87.
- [29] O. WITTICH, A. KEMPE, G. WINKLER, AND V. LIEBSCHER, *Complexity penalized least squares estimators: Analytical results*, Math. Nach., 281 (2008), pp. 582–595.
- [30] J. YANG, Y. ZHANG, AND W. YIN, *An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise*, SIAM J. Sci. Comput., 31 (2009), pp. 2842–2865.