SIAM J. SCI. COMPUT.
Vol. 38, No. 1, pp. A614–A630

EXACT ALGORITHMS FOR L^1-TV REGULARIZATION OF REAL-VALUED OR CIRCLE-VALUED SIGNALS*

MARTIN STORATH†, ANDREAS WEINMANN‡, AND MICHAEL UNSER†

Abstract. We consider L^1-TV regularization of univariate signals with values on the real line or on the unit circle. While the real data space leads to a convex optimization problem, the problem is nonconvex for circle-valued data. In this paper, we derive exact algorithms for both data spaces. A key ingredient is the reduction of the infinite search spaces to a finite set of configurations, which can be scanned by the Viterbi algorithm. To reduce the computational complexity of the involved tabulations, we extend the technique of distance transforms to nonuniform grids and to the circular data space. In total, the proposed algorithms have complexity $\mathcal{O}(KN)$, where N is the length of the signal and K is the number of different values in the data set. In particular, the complexity is $\mathcal{O}(N)$ for quantized data. It is the first exact algorithm for total variation regularization with circle-valued data, and it is competitive with the state-of-the-art methods for scalar data, assuming that the latter are quantized.

Key words. total variation regularization, total cyclic variation, circle-valued data, least absolute deviations, dynamic programming, distance transform

AMS subject classifications. 65Kxx, 90C26, 90C39

DOI. 10.1137/15M101796X

1. Introduction.

Total variation (TV) minimization has become a standard method for jump or edge preserving regularization of signals and images. Whereas the classical L^2-TV model (i.e., TV with quadratic data fidelity term [32]) is optimally matched to the Gaussian noise model, L^1 data terms are more robust to noise with more heavy tailed distributions such as Laplacian noise and to the presence of outliers; see, e.g., [30]. Further advantages are the better preservation of the contrast and the invariance to global contrast changes [8]. Since L^1-TV minimization is a convex problem for real- and vector-valued data, it is accessible by convex optimization techniques. In fact, there are several algorithms for L^1-TV minimization with scalar and vectorial data. The minimization methods are typically of an iterative nature, for example, interior point methods [22], iterative thresholding [3], alternating methods of multipliers [37, 24], semismooth Newton methods [10], primal-dual strategies [16, 7], and proximal point methods [29]. There are also other algorithms based on recursive median filtering [1] or graph cuts [13].

For univariate real-valued signals, efficient exact algorithms are available for L^2-TV, for instance, the taut string algorithm which has a linear complexity [28, 14].

*Submitted to the journal's Methods and Algorithms for Scientific Computing section April 21, 2015; accepted for publication (in revised form) November 30, 2015; published electronically February 25, 2016.

http://www.siam.org/journals/sisc/38-1/M101796.html

†Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (martin.storath@epfl.ch, michael.unser@epfl.ch). The reseach of these authors was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement 267439. The research of the first author also was supported by the DFG scientific network Mathematical Methods in Magnetic Particle Imaging.

‡Fast Algorithms for Biomedical Imaging Group, Helmholtz Zentrum München, Ingolstädter Landstraße 1, 85764 Neuherberg, Germany, and Technische Universität München, Germany, and Darmstadt University of Applied Sciences, Germany (andreas.weinmann@tum.de). This author's reseach was supported by the Helmholtz Association within the young investigator group VH-NG-526. This author also was supported by the DFG scientific network Mathematical Methods in Magnetic Particle Imaging.

A recent alternative is the algorithm of Condat [11] which shows a particularly good performance in practice. The L^1-TV problem is computationally more intricate. For data $y \in \mathbb{R}^N$ and a nonnegative weight vector $w \in \mathbb{R}^N$, it is given by

$$\arg\min_{x\in\mathbb{R}^N} \alpha \sum_{n=1}^{N-1} |x_n - x_{n+1}| + \sum_{n=1}^{N} w_n |x_n - y_n|, \tag{1}$$

where $\alpha > 0$ is a model parameter regulating the trade-off between data fidelity and TV prior. In a Bayesian framework, it corresponds to the maximum a posteriori (MAP) estimator of a summation process with Laplace distributed increments under a Laplacian noise model; see, e.g., [35]. Dümbgen and Kovac [17] have derived an exact solver of complexity $\mathcal{O}(N \log N)$ for (1). Recently, Kolmogorov, Pock, and Rolinek [26] have proposed a solver of complexity $\mathcal{O}(N \log\log N)$.

Recently, TV regularization on nonvectorial data spaces such as, e.g., Riemannian manifolds has received a lot of interest [9, 12, 38, 27, 25, 2]. The nonvectorial setting is a major challenge because the TV problem is, in general, not convex. One of the simplest examples, where the L^1-TV functional is nonconvex, is circle-valued data. Such data appears, for example, as phase signals (which are defined modulo 2π) and as time series of angles. Particular examples for the latter are the data on the orientation of the bacterial flagellar motor [33] and the data on wind directions [15]. The L^1-TV functional for circle-valued data $y \in \mathbb{T}^N$ is given by

$$\arg\min_{x\in\mathbb{T}^N} \alpha \sum_{n=1}^{N-1} d_{\mathbb{T}}(x_n, x_{n+1}) + \sum_{n=1}^{N} w_n \, d_{\mathbb{T}}(x_n, y_n), \tag{2}$$

where $d_{\mathbb{T}}(u, v)$ denotes the arc length distance of $u, v \in \mathbb{T} = \mathbb{S}^1$. Theoretical results on total cyclic variation can be found in the papers of Giaquinta, Modica, and Souček [23] and of Cremers and Strekalovskiy [12]. The authors of the latter have shown that the problem is computationally at least as complex as the Potts problem; this means, in particular, that it is NP-hard in dimensions greater than one. Current minimization strategies for (2) are based on convex relaxations [12], proximal point splittings [38], or iteratively reweighted least squares [25]. However, due to the nonconvexity of (2), these iterative approaches do not guarantee convergence to a global minimizer. Furthermore, they are computationally demanding. To our knowledge, no exact algorithm for (2) has been proposed yet.

In this paper, we propose exact noniterative algorithms for L^1-TV minimization on scalar signals (1) and on circle-valued signals (2). A key ingredient is the reduction of the infinite search space, $\mathbb{R}^N$ or $\mathbb{T}^N$, to a finite search space V^N. This reduction allows us to use the Viterbi algorithm [36, 21] for the minimization of discretized energies as presented in [20]. A time-critical step in the Viterbi algorithm is the computation of a distance transform w.r.t. the nonuniform grid induced by V. For the scalar case, we generalize the efficient two-pass algorithm of Felzenszwalb and Huttenlocher [18, 19] from uniform grids to our nonuniform setup. We further propose a new method for efficiently computing the distance transforms in the circle-valued case. In total, our solvers have complexity $\mathcal{O}(KN)$, where K denotes the number of different values in the data. In particular, if the data is quantized to finitely many levels, the algorithmic complexity is $\mathcal{O}(N)$. It is the first exact algorithm for TV regularization of circle-valued signals, and it is competitive with the state-of-the-art methods for real-valued signals, assuming that the latter are quantized.

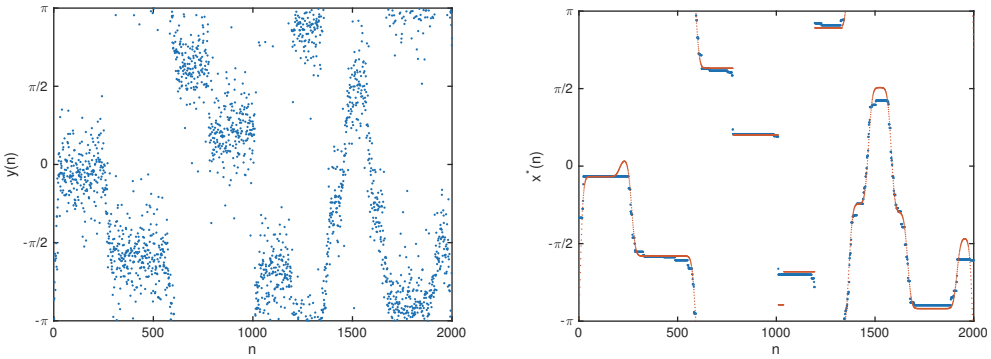

FIG. 1. *Left: Synthetic circle-valued signal corrupted by noise. Right: Global minimizer x^* of the TV functional with $\alpha = 15$. (Ground truth displayed as small red points.) The noise is almost completely removed and the jumps are preserved. The phase jumps of 2π are taken into account properly.*

1.1. Organization of the paper. In section 2, we show that the search space can be reduced to a finite set. In section 3, we present our minimization strategy for the reduced problem. In section 4, we present numerical experiments based on synthetic and real data. Eventually, we discuss the relations to other approaches.

2. Reduction of the search space. A crucial step in our derivation is the reduction of the search space to a finite set. In the following, we denote the L^1-TV functional for data $y \in \mathbb{R}^N$ or $y \in \mathbb{T}^N$ by

$$T_{\alpha;y}(x) = \alpha \sum_{n=1}^{N-1} d(x_n, x_{n+1}) + \sum_{n=1}^{N} w_n \, d(x_n, y_n).$$

Here d denotes the distance that corresponds to the data space, i.e., the Euclidean distance for real-valued data and the arc length distance for circle-valued data. We further use the notation $\mathrm{Val}(y)$ to denote the set of values of the N-tuple y, i.e.,

$$\mathrm{Val}(y) = \{v : \text{ there is } n \text{ with } 1 \le n \le N \text{ s.t. } y_n = v\},$$

or, using set notation, $\mathrm{Val}(y) = \{y_1, \dots, y_n\}$. Also recall that a (weighted) median of y is a minimizer of the functional

$$\mu \mapsto \sum_{n=1}^{N} w_n d(\mu, y_n).$$

We will see that there are always minimizers of the L^1-TV problem whose values are all contained in the values $\mathrm{Val}(y)$ of the data y (united with the antipodal points $\mathrm{Val}(\tilde{y})$ in the circle-valued case).

2.1. Real-valued data. Let us first consider the real-valued case. The following assertion on the minimizers of the L^1-TV functional has been proven by Alliney [1]. His proof is based on results of convex analysis. Here, we develop an alternative technique which does not exploit the convexity of the TV functional. The crucial point is that this technique will allow us to treat the more involved nonconvex circular

case later on.

THEOREM 1. *Let $\alpha > 0$, $y \in \mathbb{R}^N$, and $V = \mathrm{Val}(y)$. Then*

$$\min_{x\in\mathbb{R}^N} T_{\alpha;y}(x) = \min_{x\in V^N} T_{\alpha;y}(x).$$

Proof. The method of proof is as follows: we consider an arbitrary $x \in \mathbb{R}^N$ and construct $x' \in V^N$ such that $T_{\alpha;y}(x') \le T_{\alpha;y}(x)$. If we apply this procedure to a minimizer x^*, we obtain a minimizer with values in V^N which is the assertion of the theorem.

So let $x \in \mathbb{R}^N$ be arbitrary and let us construct $x' \in V^N$ with smaller or equal $T_{\alpha;y}$ value by the following procedure. Let $\mathcal{I}$ be the set of maximal intervals of $\{1,\dots,N\}$, where x is constant on and where x does not attain its value in V. It means that each element I of $\mathcal{I}$ is an "interval" of the form $I = \{l, l+1, \dots, r\}$ such that $a := x_l = \cdots = x_r \notin V$ and such that $x_{l-1} \ne x_l$ (unless $l = 1$) and $x_r \ne x_{r+1}$ (unless $r = N$). If $\mathcal{I}$ is empty, then $x \in V^N$ and we are done. Otherwise, we decrease the number of such intervals $|\mathcal{I}|$ by the following rule: Choose an interval $I = \{l,\dots,r\} \in \mathcal{I}$. We construct $\bar{x}$ which equals x outside I and choose its constant value a' such that the corresponding number of intervals with values which are not in V is strictly smaller than $|\mathcal{I}|$. We distinguish three cases.

First assume that I is not a boundary interval (i.e., $l \ne 1$ and $r \ne N$) and that the values of the two neighboring intervals, x_{l-1} and x_{r+1}, are both smaller than the value on I which is equal to a. (We call an interval a neighbor of I if it contains the index $l-1$ or $r+1$.) We denote the nearest smaller and the nearest greater neighbors of a in V by b^- and b^+, respectively. Let $b' = \max(x_{l-1}, x_{r+1}, b^-)$. By replacing a by some $a' \in [b', b^+]$ we change the TV penalty by $2\alpha(a'-a)$ and the data penalty by $(W^- - W^+)(a'-a)$. Here $W^+ = \sum_{\{i\in I: y_i > a\}} w_i$ and $W^- = \sum_{\{i\in I: y_i \le a\}} w_i$ are the weights of elements in the interval I that are greater or smaller than a, respectively. If $W^- + 2\alpha < W^+$, we let a' equal its greater neighbor b^+. Otherwise, we let $a' = b'$, where, by the definition above, $b' = \max(x_{l-1}, x_{r+1}, b^-)$. If $a' = b^-$, then the value of $\bar{x}$ on I belongs to V. If $a' \in \{x_{l-1}, x_{r+1}\}$ the interval merges with one of its neighbors. In both cases, the number of intervals with "undesired" values, $|\mathcal{I}|$, decreases by one. By symmetry, the same argumentation is valid for the case that the values of the neighboring intervals x_{l-1} and x_{r+1} are both greater than a.

As a second case we consider the situation where I is not a boundary interval and where x_{l-1} is smaller and x_{r+1} is greater than a. (Again, the case $x_{l-1} > a > x_{r+1}$ is dealt with by symmetry.) Since replacing a by any value in $[x_{l-1}, x_{r+1}]$ does not change the TV penalty, we only need to look at the approximation error. This amounts to setting a' equal to a (weighted) median of $y_l, \dots, y_r$. Note that there exists a (weighted) median that it is contained in $\{y_l,\dots,y_r\} \subset V$. We use such a median in V to define a'. Hence, also in this case, $|\mathcal{I}|$ decreases by one.

Finally, we consider the third case where the interval is located at the boundary. If either $1 \in I$ or $N \in I$, then we proceed analogously to the first case. The relevant difference is that we let a' equal its greater nearest neighbor b^+ if $W^- + \alpha < W^+$ (instead of $W^- + 2\alpha < W^+$). If the interval touches both boundaries, i.e., if $I = \{1,\dots,N\}$, we proceed as in the second case, which is setting a' to be a (weighted) median y which is contained in V.

We repeat the above procedure until $|\mathcal{I}| = 0$, which implies that the final result x' is contained in V^N. By construction, the functional value $T_{\alpha;y}(x')$ is not exceeding the functional value of x, since all intermediately constructed $\bar{x}$ do so. This completes the proof. ∎

Note that the assertion of Theorem 1 is not true for quadratic data fidelities. As the following simple example shows, it is not uncommon that $\mathrm{Val}(\hat{x}) \cap \mathrm{Val}(y) = \emptyset$ for all L^2-TV minimizers $\hat{x}$. We consider toy data $y = (0, 1) \in \mathbb{R}^2$ and the corresponding L^2-TV functional given by $x \mapsto \alpha|x_1 - x_2| + x_1^2 + (x_2 - 1)^2$. It is easy to check that the unique minimizer of this L^2-TV problem is given by $\hat{x} = (\alpha/2, 1-\alpha/2)$ if $\alpha < 1$ and by $\hat{x} = (1/2, 1/2)$ otherwise. We note that this is an example where $\mathrm{Val}(\hat{x}) \cap \mathrm{Val}(y) = \emptyset$ even for all $\alpha > 0$. This shows that one cannot even expect an analogous result when one chooses a suitable parameter. For a more detailed discussion of this aspect we refer to the paper of Nikolova [30]. It is interesting to note that an assertion analogous to that of Theorem 1 can be shown for the Potts model, although the model and the corresponding proof are quite different; see [40, 34].

2.2. Circle-valued data. Now we use the techniques developed for the real-valued case in our proof of Theorem 1 in the more involved situation of circle-valued data to prove the following theorem allowing for the reduction of the search space for minimizers of the L^1-TV functional for $\mathbb{S}^1$-valued data as well.

THEOREM 2. *Let $\alpha > 0$, $y \in \mathbb{T}^N$, and $V = \mathrm{Val}(y) \cup \mathrm{Val}(\tilde{y})$, where $\tilde{y}$ denotes the tuple of antipodal points of y. Then*

$$\min_{x \in \mathbb{T}^N} T_{\alpha;y}(x) = \min_{x \in V^N} T_{\alpha;y}(x).$$

Proof. As in the proof of Theorem 1, we consider an arbitrary $x \in \mathbb{T}^N$ and construct $x' \in V^N$ such that $T_{\alpha;y}(x') \leq T_{\alpha;y}(x)$. Note that, in contrast to the proof of Theorem 1, $V = \mathrm{Val}(y) \cup \mathrm{Val}(\tilde{y})$ here. Similarly, we let $\mathcal{I}$ be the set of the maximal intervals I of $\{1, \dots, N\}$, where x is constant on and where the attained value a of x on I is not contained in V. We decrease the number of such intervals $|\mathcal{I}|$ by the procedure explained below.

Before being able to give the explanation we need some notions related to $\mathbb{S}^1$ data. Let us consider a point a on the sphere and its antipodal point $\tilde{a}$. Then there are two hemisphere/half-circles connecting a and $\tilde{a}$. We use the convention that $\tilde{a}$ is contained in both hemispheres, whereas a is contained in none of them. These two hemispheres can be distinguished into the hemisphere $H_1 = H_1(a)$ determined by walking from a in a clockwise direction and the hemisphere $H_2 = H_2(a)$ obtained from walking in a counterclockwise direction.

Equipped with these preparations, we explain the procedure to reduce the number of intervals $|\mathcal{I}|$. We pick an arbitrary interval $I = \{l, \cdots, r\} \in \mathcal{I}$ and let $a = x_l = \cdots = x_r$ be the value of x on I. We let b_1 and b_2 be the nearest neighbors of a in $H_1 \cap V$ and in $H_2 \cap V$, which are the values of the data (or their antipodal points) on the clockwise and the counterclockwise hemisphere, respectively. We note that b_1, b_2 exist and both are not equal to the antipodal point $\tilde{a}$ of a. This is because, together with a point p, its antipodal point $\tilde{p}$ is also contained in V, which implies that either p or $\tilde{p}$ is a member of H_1 and either $\tilde{p}$ or p is a member of H_2. Since a is not contained in the set V of values of y and its antipodal points, the distance to either p or $\tilde{p}$ is strictly smaller that π. We construct $\bar{x}$ which equals x outside I and with constant value a' on I such that $|\mathcal{I}|$ decreases. We have to differentiate three cases.

First we assume that I is no boundary interval and that the left and right neighboring candidate items x_{l-1} and x_{r+1} are both located on the clockwise hemisphere H_1 and none of them agrees with $\tilde{a}$. Let $W_1 = \sum_{i:y_i \in H_1} w_i$ be the weight of y on H_1 and let $W_2 = \sum_{i:y_i \in H_2} w_i$ be the weight of y on H_2. (Note that $\tilde{a}$ which is the only point in both H_1 and H_2 is not a member of y.) If $W_1 > W_2 + 2\alpha$, which means that

the clockwise hemisphere H_1 is "heavier" than the counterclockwise hemisphere H_2 plus the variation penalty, we set a' to be the nearest neighbor of a in $\{x_l, x_r, b_1\}$. This may be visualized as shifting the value on I in a clockwise direction until we hit the first value in $\{x_l, x_r, b_1\}$. Since $W_1 > W_2 + 2\alpha$, we have that $T_{\alpha;y}(\bar{x}) \leq T_{\alpha;y}(x)$. Otherwise, we set $a' = b_2$, which means that we shift to the other direction. Since then $W_1 \leq W_2 + 2\alpha$, we get $T_{\alpha;y}(\bar{x}) \leq T_{\alpha;y}(x)$ also in this situation. By symmetry, the same argument applies when both x_{l-1} and x_{r+1} are located on the counterclockwise hemisphere.

In the second case we assume that I is no boundary interval and that x_{l-1} and x_{r+1} are located on different hemispheres. Here we also include the case where one of or both x_{l-1} and x_{r+1} are antipodal to a. If only one neighbor is antipodal, we interpret it to lie on the opposite hemisphere of the nonantipodal member. If both neighbors are antipodal, we interpret them to lie on different hemispheres. We let $\mathcal{C}$ be the arc connecting x_{l-1} and x_{r+1} which has a as a member. Letting a' equal any value on the arc $\mathcal{C}$ leads to $TV(\bar{x}) \leq TV(x)$, meaning that it does not increase the variation penalty $TV(x) = \sum_n \alpha d_{\mathbb{T}}(x_n, x_{n+1})$. By definition, the data term is minimized by letting a' be a (weighted) median of $y_l, \ldots, y_r$. A (weighted) median of the circle-valued data can be chosen as an element of the unique values $\{y_l, \ldots, y_r\}$ unified with the antipodal points $\{\tilde{y}_l, \ldots, \tilde{y}_r\}$. We choose a' as such a median. This implies $T_{\alpha;y}(\bar{x}) \leq T_{\alpha;y}(x)$.

It remains to consider the boundary intervals. If $I = \{1, \ldots, N\}$, we proceed as in the second case and set $\bar{u}_i = a'$ for all i, where a' is a (weighted) median of y which is contained in V. Else, if either $1 \in I$ or $N \in I$ we proceed analogously to the first case with the difference that we replace the decision criterion $W_1 > W_2 + 2\alpha$ employed there by $W_1 > W_2 + \alpha$.

We repeat the above procedure until $|\mathcal{I}| = 0$, which implies that the values of the final result x' all lie in V^N. Then plugging in a minimizer $x = x^*$ results in a minimizer $x' \in V^N$ which shows the theorem. ∎

As for scalar data, the assertion of Theorem 2 is not true for quadratic data terms. This can be seen using the previous example interpreting the data $y = (0, 1)$ as angles.

In order to illustrate the difference to the real-valued data case, let us point out a degenerate situation which is due to the circular nature of the data. Assume that the data only consists of a point $z \in \mathbb{T}$ and its antipodal point $\tilde{z}$, i.e., $y = (z, \tilde{z})$. For sufficiently large α, any minimizer $\hat{x}$ of (2) is constant, say, $\hat{x} = (a, a)$. Since the TV penalty gets equal to zero, a must be equal to a median of y. It is not hard to check that every point on the sphere is a median of y. This behavior appears curious at first glance. However, the data shows no clear tendency toward a distinguished orientation. Thus, every estimate can be considered as equally good. The result seems even more natural than that of L^2-TV regularization. An L^2-TV minimizer would consist of one of the two "mean orientations" which are given by rotating z by $\pi/2$ in a clockwise or counterclockwise direction. Both minimizers seem rather arbitrary, and, moreover, the two options point to opposing directions.

3. Efficient algorithms for the reduced problems. Theorems 1 and 2 allow us to reduce the infinite search spaces $\mathbb{R}^N$ and $\mathbb{T}^N$ in (1) and (2), respectively, to the finite sets V^N, which are specified in these theorems. Thus, it remains to solve the problems: find

$$x^* \in \arg\min_{x \in V^N} T_{\alpha;y}(x).$$

This can be achieved with dynamic programming whose basic idea is to decompose the problem into a series of similar, simpler, and tractable subproblems. For an early account on dynamic programming, we refer to [4].

3.1. The Viterbi algorithm for energy minimization on finite search spaces. We utilize a dynamic programming scheme developed by Viterbi [36]; see also [21]. Related algorithms have been proposed in [5, 6]. In this section, we review a special instance of the Viterbi algorithm following the presentation of the survey [20].

We aim at minimizing an energy functional of the form

$$E(x_1, \dots, x_N) = \alpha \sum_{n=1}^{N-1} d(x_n, x_{n+1}) + \sum_{n=1}^{N} w_n d(x_n, y_n), \tag{3}$$

where the arguments $x_1, \dots, x_N$ can take values in a finite set $V = \{v_1, \dots, v_K\}$. The Viterbi algorithm solves this problem in two steps: tabulation of energies and reconstruction by backtracking.

For the tabulation step, the starting point is the table $B^1 \in \mathbb{R}^K$ given by

$$B_k^1 = w_1 d(v_k, y_1) \quad \text{for } k = 1, \dots, K.$$

From now on, the symbol K denotes the cardinality of V. For $n = 2, \dots, N$ we successively compute the tables $B^n \in \mathbb{R}^K$ which are given by

$$B_k^n = w_n d(v_k, y_n) + \min_l \{B_l^{n-1} + \alpha\, d(v_k, v_l)\} \tag{4}$$

for $k = 1, \dots, K$. The entry B_k^n represents the energy of a minimizer on data $(y_1, \dots, y_n)$ whose endpoint is equal to v_k.

For the backtracking step, it is convenient to introduce an auxiliary tuple $l \in \mathbb{N}^N$ which stores minimizing indices. We initialize the last entry of l by $l_N = \arg\min_k B_k^N$. Then we successively compute the entries of l for $n = N-1, N-2, \dots, 1$ by

$$l_n = \arg\min_k B_k^n + \alpha\, d(v_k, v_{l_{n+1}}). \tag{5}$$

Eventually, we reconstruct a minimizer $\hat{x}$ from the indices in l by

$$\hat{x}_n = v_{l_n} \quad \text{for } n = 1, \dots, N.$$

The result $\hat{x}$ is a global minimizer of the energy (3); see [20]. For a general functional, filling the table B^n in (4) costs $\mathcal{O}(K^2)$. This implies that the described procedure is in $\mathcal{O}(K^2 N)$. In the next subsections, we will derive procedures to reduce the complexity for filling the tables B^n for our concrete problem to $\mathcal{O}(K)$.

3.2. Distance transform on a nonuniform real-valued grid. We first consider the case of real-valued data. The time critical part of the Viterbi algorithm is the computation of the minima

$$D_k = \min_l B_l + \alpha |v_k - v_l| \quad \text{for all } k = 1, \dots, K. \tag{6}$$

This problem is known as distance transform with respect to the ℓ^1 distance (weighted by α). Felzenszwalb and Huttenlocher [18, 19] describe an efficient algorithm for (6) when V forms an integer grid, i.e., $V = \{0, \dots, K-1\}$. In our setup, V forms a nonuniform grid in general. Therefore, we generalize their method accordingly.

In the following, we identify the elements of V with a K-dimensional vector v which is ordered in ascendingly, i.e., $v_1 < v_2 < \cdots < v_K$. The sorting causes no problems since we can sort v in $\mathcal{O}(K \log K)$, and since the logarithm of the number of values K is smaller than the data length N, we have $\mathcal{O}(K \log K) \subset \mathcal{O}(KN)$.

As we will show below, the two-pass procedure in Algorithm 1 computes the real-valued distance transform D.

Algorithm 1. Real-valued distance transform distTransReal(B, v, α).

Input: $B \in \mathbb{R}^K$; $v \in \mathbb{R}^K$ sorted in ascending order; $\alpha > 0$;
Output: Distance transform D
begin
  $D \leftarrow B$;
  for $k \leftarrow 2, 3, \ldots, K$ **do**
    $D_k \leftarrow \min(D_{k-1} + \alpha(v_k - v_{k-1}); D_k)$;
  end
  for $k \leftarrow K-1, K-2, \ldots, 1$ **do**
    $D_k \leftarrow \min(D_{k+1} + \alpha(v_{k+1} - v_k); D_k)$;
  end
  return D;
end

In order to show the correctness of the method, we build on the structurally related proof given in [18] for uniform grids. The major new idea is to pass from discrete to continuous infimal convolutions in order to deal with the nonequidistant grid. The (continuously defined) infimal convolution of two functions F and G on $\mathbb{R}$ with values on the extended real line $[-\infty, \infty]$ is given by

$$F \square G(r) = \inf_{u \in \mathbb{R}} \{F(u) + G(r-u)\};$$

see [31, section 5]. In the following, the infimum will always be attained, so that it actually is a minimum; we use this fact in the notation we employ.

For real-valued data, we get the following result accelerating the bottleneck operation in the general Viterbi algorithm from section 3.1.

THEOREM 3. *Algorithm* 1 *computes* (6) *in* $O(K)$.

Proof. We define the function F on $\mathbb{R}$ by $F(v_l) = B_l$ for $v_l \in V$ and by $F(r) = \infty$ for $r \in \mathbb{R} \setminus V$. We also define $G(u) = \alpha|u|$. Then, D_k can be formulated in terms of the infimal convolution of F and G evaluated at v_k, that is,

$$D_k = F \square G(v_k).$$

In order to decompose G, we define

$$G_+(r) = \begin{cases} \alpha r & \text{for } r \geq 0, \\ \infty & \text{otherwise,} \end{cases} \quad \text{and} \quad G_-(r) = \begin{cases} -\alpha r & \text{for } r \leq 0, \\ \infty & \text{otherwise.} \end{cases}$$

We see that G is the infimal convolution of G_+ and G_- by using that

$$G_+ \square G_-(r) = \min_{t \in \mathbb{R}} G_+(t) + G_-(r-t) = \alpha|r| = G(r).$$

By the associativity of the infimal convolution (see [31, section 5]), we obtain

$$F \square G = F \square (G_+ \square G_-) = (F \square G_+) \square G_-. \tag{7}$$

We use the right-hand representation; for the right-hand term in brackets, we get, for $v_k \in V$,

$$\begin{aligned} F\square G_+(v_k) &= \min_j F(v_j) + G_+(v_k - v_j) \\ &= \min_{j \le k} F(v_j) + \alpha(v_k - v_j) \\ &= \min\{\min_{j \le k-1} F(v_j) + \alpha(v_k - v_j); F(v_k)\} \\ &= \min\{\min_{j \le k-1} F(v_j) + \alpha(v_{k-1} - v_j + v_k - v_{k-1}); F(v_k)\} \\ &= \min\{F\square G_+(v_{k-1}) + \alpha(v_k - v_{k-1}); F(v_k)\}. \end{aligned}$$

Now, we denote the result by $F' = F\square G_+$ and continue to manipulate the right-hand term of (7) noticing that for all $r \notin V$, we have $F'(r) = \infty$. We obtain

$$\begin{aligned} F'\square G_-(v_k) &= \min_j F'(v_j) + G_-(v_k - v_j) \\ &= \min_{j \ge k} F'(v_j) - \alpha(v_k - v_j) \\ &= \min\{\min_{j \ge k+1} F'(v_j) - \alpha(v_k - v_j); F'(v_k)\} \\ &= \min\{\min_{j \ge k+1} F'(v_j) - \alpha(v_{k+1} - v_j + v_k - v_{k+1}); F'(v_k)\} \\ &= \min\{F\square G_-(v_{k+1}) + \alpha(v_{k-1} - v_k); F'(v_k)\}. \end{aligned}$$

The above recursive equations show that the forward pass and the backward pass of Algorithm 1 compute the desired infimal convolutions. ∎

3.3. Distance transform on a nonuniform circle-valued grid. Now we look at the circular case. In this case, the corresponding ℓ^1 distance transform is given by

$$D_k = \min_l B_l + \alpha d_{\mathbb{T}}(v_k, v_l) \quad \text{for all } k = 1, \dots, K. \tag{8}$$

Our task is to compute the distance transform in the circle case as well. To this end, we employ the angular representation of values on the circle in the interval $(-\pi, \pi]$. As in the real-valued case, we identify the elements of V with a K-tuple v which is sorted in ascending order. In order to compute (8), we use Algorithm 2.

Algorithm 2. Circle-valued distance transform distTransCirc(B, v, α).

Input: $B \in \mathbb{R}^K$; $v \in (-\pi, \pi]^K$ sorted in ascending order; $\alpha > 0$;
Output: Distance transform D
begin
  $B' \leftarrow (B_1, \dots, B_K, B_1, \dots, B_K, B_1, \dots, B_K)$;
  $v' \leftarrow (v_1 - 2\pi, \dots, v_K - 2\pi, v_1, \dots, v_K, v_1 + 2\pi, \dots, v_K + 2\pi)$;
  $D' \leftarrow$ distTransReal(B', v', α);
  $D \leftarrow (D'_{K+1}, \dots, D'_{2K})$;
  return D;
end

We point out that this algorithm employs the real-valued distance transform of section 3.2. The next result in particular shows that Algorithm 2 actually computes a minimizer of the distance transform (8). The proof uses infimal convolutions on the real line and employs the corresponding statement Theorem 3 for real-valued data.

THEOREM 4. *Algorithm* 2 *computes* (8) *in* $O(K)$.

Proof. First we observe that the arc length distance on $\mathbb{S}^1 = \mathbb{T}$ can be written using the absolute value on $(-\pi, \pi]$ by

$$d_{\mathbb{T}}(u, w) = \min\{|u - 2\pi - w|; |u - w|; |u + 2\pi - w|\}$$

for $u, w \in (-\pi, \pi]$. We define the extended real-valued functions F, F' defined on $\mathbb{R}$ as follows: we let $F(v_k) = B_k$ on the points v_k and $F(r) = \infty$ for $r \in \mathbb{R} \setminus V$; to define F', we let

$$F'(t) = \min\left\{F(t - 2\pi), F(t), F(t + 2\pi)\right\}.$$

Our goal is to show that D_k is the infimal convolution of F' and G with G given by $G(v) = \alpha|v|$. We get that

$$\begin{aligned}
D_k &= \min_{r\in\mathbb{R}}\{F(r) + \alpha \min\{|r - 2\pi - v_k|; |r - v_k|; |r + 2\pi - v_k|\}\} \\
&= \min_{r\in\mathbb{R}} \min\{F(r) + \alpha|r - 2\pi - v_k|; \\
&\qquad F(r) + \alpha|r - v_k|; F(r) + \alpha|r + 2\pi - v_k|\}\} \\
&= \min\{\min_{r\in\mathbb{R}} F(r + 2\pi) + \alpha|r - v_k|; \\
&\qquad \min_{r\in\mathbb{R}} F(r) + \alpha|r - v_k|; \min_{r\in\mathbb{R}} F(r - 2\pi) + \alpha|r - v_k|\}\} \\
&= \min_{r\in\mathbb{R}} F'(r) + \alpha|r - v_k| = F'\square G(v_k).
\end{aligned}$$

Hence, D_k is the infimal convolution of F' and G. We now shift the vector of assumed values v by -2π and 2π and consider the concatenation with v to obtain v', which is given by

$$v' = (v_1 - 2\pi, \ldots, v_K - 2\pi, v_1, \ldots, v_K, v_1 + 2\pi, \ldots, v_K + 2\pi).$$

We note that v' is ordered ascendingly. We let

$$D'_l = F'\square G(v'_l) \quad \text{for } l = 1, \ldots 3K. \tag{9}$$

By Theorem 3, we can compute (9) in $\mathcal{O}(K)$ using Algorithm 1. Eventually, we observe that

$$D_k = F'\square G(v_k) = D'_{K+k} \quad \text{for } k = 1, \ldots, K,$$

which completes the proof. ∎

3.4. Complete algorithm. The complete procedure is described in Algorithm 3. Summarizing, we have obtained the following result.

THEOREM 5. *Let $y \in \mathbb{R}^N$ and $V = \mathrm{Val}(y)$, or $y \in \mathbb{T}^N$ and $V = \mathrm{Val}(y) \cup \mathrm{Val}(\tilde{y})$. Further let K be the number of elements in V. Then Algorithm 3 computes a global minimizer of the L^1-TV problem with real-valued* (1) *or circle-valued data* (2) *in $\mathcal{O}(KN)$. In particular, if data is quantized to a finite set, then the algorithms for real-valued or circle-valued signals are in $\mathcal{O}(N)$.*

4. Numerical results. We illustrate the effects of L^1-TV minimization for real- and circle-valued data. We consider both synthetic and real-life data.

Algorithm 3. Exact algorithm for the L^1-TV problem of real- or circle-valued signals.

Input: Data $y \in \mathbb{R}^N$ or $y \in \mathbb{T}^N$; regularization parameter $\alpha > 0$; weights $w \in (\mathbb{R}_0^+)^N$;
Output: Global minimizer $\hat{x}$ of (1) or (2);
begin
  /* 1. *Init candidate values* */
  $V \leftarrow \mathrm{Val}(y)$; /* *Real-valued case* */
  $V \leftarrow \mathrm{Val}(y) \cup \mathrm{Val}(\tilde{y})$; /* *Circle-valued case* */
  $v \leftarrow K$-tuple of elements of V, sorted ascendingly;
  /* 2. *Tabulation* */
  for $k \leftarrow 1$ **to** K **do**
    $B_k^1 \leftarrow w_1 d(v_k, y_1)$;
  end
  for $n \leftarrow 2$ **to** N **do**
    $D \leftarrow \mathrm{distTransReal}(B^n, v, \alpha)$; /* *Real-valued case* */
    $D \leftarrow \mathrm{distTransCirc}(B^n, v, \alpha)$; /* *Circle-valued case* */
    for $k \leftarrow 1$ **to** K **do**
      $B_k^n \leftarrow w_n d(v_k, y_n) + D_k$;
    end
  end
  /* 3. *Backtracking* */
  $l \leftarrow \arg\min_{k=1,\dots,K} B_k^N$;
  $\hat{x}_n \leftarrow v_l$;
  for $n \leftarrow N-1, N-2, ..., 1$ **do**
    $l \leftarrow \arg\min_{k=1,\dots,K} B_k^n + \alpha\, d(v_k, \hat{x}_{n+1})$;
    $\hat{x}_n \leftarrow v_l$;
  end
  return $\hat{x}$;
end

Experimental setup. We have implemented our algorithms in MATLAB. The experiments were conducted on a desktop computer with 3.5 GHz Intel Xeon E5 and 32 GB memory. The weight vector w can be employed to account for nonequidistant sampling; it means that the data $y_n = f(t_n)$ is the sampling of a (continuously defined) signal f at nonequidistant knots $t_1 < t_2 < \cdots < t_N$. A reasonable choice for w_n is the average distance of the sampling point t_n to its nearest neighbors t_{n-1}, t_{n+1}, i.e., $w_n = (t_n - t_{n-1} + t_{n+1} - t_n)/2 = (t_{n+1} - t_{n-1})/2$. In our experiments, we focus on equidistant sampling so that we have $w_n = 1$ for all $n = 1, \dots, N$. To quantify the denoising performance, we occasionally give the manifold-valued version of the *signal-to-noise ratio improvement* (see [35, Chapter 10] and [38]). It is given by

$$\Delta\mathrm{SNR} = 10 \log_{10} \left(\frac{\sum_n d(\bar{y}_n, y_n)^2}{\sum_n d(\bar{y}_n, x_n^*)^2} \right),$$

where $\bar{y}$ denotes the ground truth. For real-valued data, we let d denote the Euclidean metric. If not mentioned explicitly, the regularization parameter α is adjusted empirically. The higher the value of α we choose, the more strongly we smooth the signal.

Circular L^1-TV on synthetic data. In the introductory experiment (Figure 1), we have computed the TV minimizer for a circle sample signal with known ground truth $\bar{y}$. The signal y was created by corrupting the phase angle $\bar{\varphi}$ of the original signal by

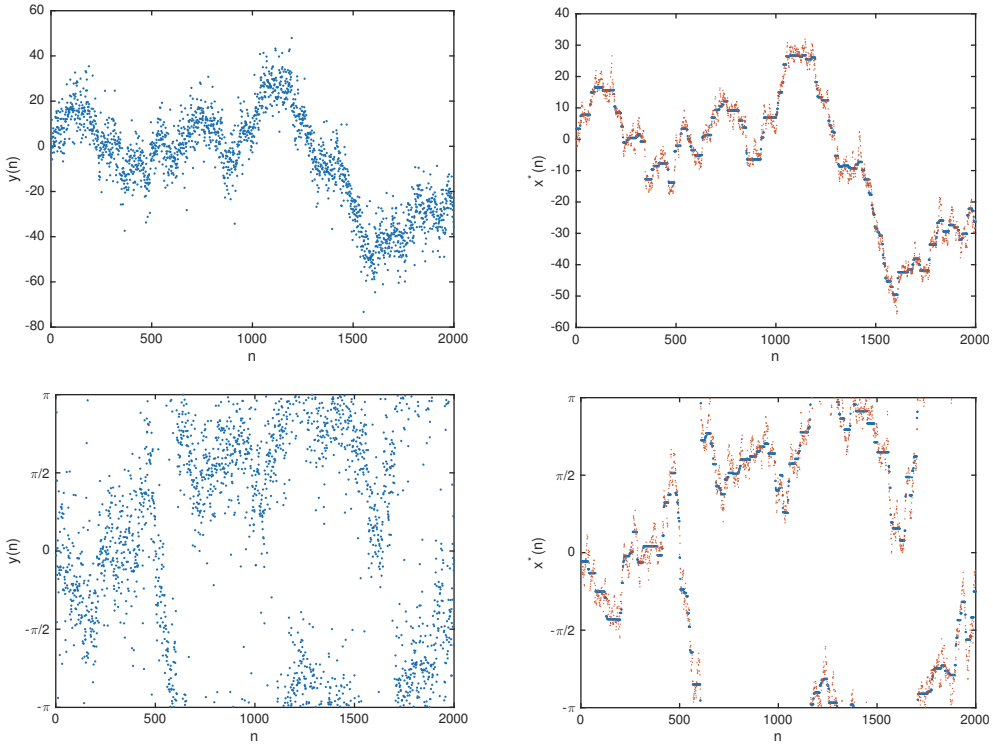

FIG. 2. *Top left: Realization of a Levy process with Laplacian increments ($\rho = 1.0$), corrupted with Laplacian white noise ($\xi = \xi_1 = \cdots = \xi_N = 5.0$). Top right: The minimizer of the L^1-TV functional with parameter $\alpha = \xi/\rho$ is the MAP estimate (ΔSNR: 8.8; ground truth displayed as small red points). Bottom: Analogous experiment for circle-valued data with $\rho = 0.1$, $\xi = 0.5$, and $\alpha = 5.0$ (ΔSNR: 9.4).*

Laplacian distributed white noise of standard deviation $\sigma = 0.5$. That is, the signal y is given by $y_j = e^{\mathrm{i}(\bar{\varphi}_j + \eta_j)}$ where η denotes the noise vector. The experiment illustrates the denoising capabilities of minimization for circle-valued data. In particular, we observe that the phase jumps by 2π are taken into account properly. The runtime was 3.5 seconds.

L^1-TV as MAP estimator. Under certain assumptions on the signal and the noise, the Bayesian framework gives a suggestion for the parameter α. For an introduction to the related statistical concepts we exemplarily refer to the book [35].

Assume that the true signal $\bar{y} \in \mathbb{R}^N$ (or $\bar{y} \in \mathbb{T}^N$) is generated according to a Levy process with Laplacian increments; that is, $\bar{y}$ is a random vector and the increments follow a distribution with density $P(\bar{y}_n|\bar{y}_{n-1}) \sim e^{-d(\bar{y}_n, \bar{y}_{n-1})/\rho}$. Also assume that the noise is distributed according to $P(y_n|\bar{y}_n) \sim e^{-d(y_n, \bar{y}_n)/\xi_n}$. Here, ρ and ξ_n are positive parameters. The MAP estimator is given by

$$\begin{aligned} x^*_{MAP} &= \arg\max_x P(y|x) = \arg\max_x P(x)P(x|y) \\ &= \arg\max_x \prod_{n=1}^{N-1} e^{-d(x_n, x_{n+1})/\rho} \prod_{n=1}^{N} e^{-d(x_n, y_n)/\xi_n} \\ &= \arg\min_x \frac{1}{\rho} \sum_{n=1}^{N-1} d(x_n, x_{n+1}) + \sum_{n=1}^{N} \frac{1}{\xi_n} d(x_n, y_n). \end{aligned}$$

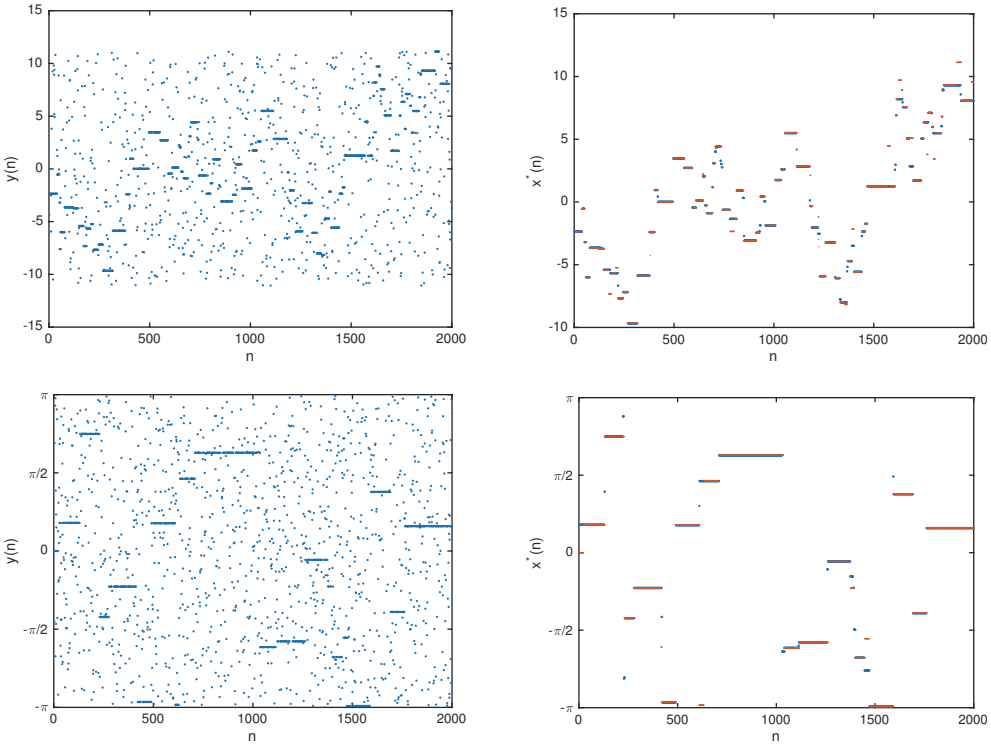

FIG. 3. *Top left: Realization of a (real-valued) Levy process with compound Poisson distributed increments ($\lambda = 0.1$), corrupted with impulsive noise with compound Poisson distribution ($\lambda' = 0.5$). Top right: The minimizer of the L^1-TV functional with parameter $\alpha = 5.0$ (ΔSNR: 17.6). Bottom: Analogous experiment for circle-valued data with $\lambda = 0.01$, $\lambda' = 1.0$, and $\alpha = 10.0$ (ΔSNR: 13.4).*

The last equality has been obtained by taking the logarithm. This derivation reveals that the L^1-TV model is the MAP estimator in the above probabilistic setup. Therefore, the natural parameter choice is $\alpha = 1/\rho$ and $w_n = 1/\xi_n$ for $n = 1, \dots, N$. In particular, for uniform parameters $\xi = \xi_1 = \cdots = \xi_N$, we have $\alpha = \xi/\rho$. Figure 2 shows the realization of such signals and their MAP estimates.

Robustness to impulsive noise. Besides the above Laplacian noise and innovation models, the (real-valued) L^1-TV estimator is known for its robustness to impulsive noise and for its good performance on piecewise constant signals; see, e.g., [22, 10, 17, 26]. In the following, we illustrate this observation and reveal a similarly good performance for circle-valued data with impulsive noise imposed. We create a compound Poisson distributed random vector $s \in \mathbb{R}^N$; that is, $s_n = 0$ with probability $e^{-\lambda}$ and s_n is uniformly distributed in $[-a, a]$ with probability $1 - e^{-\lambda}$ (see [35]). (Here, we use $a = 4$.) The (true) signal $\bar{y}$ is given as the summation process of the increments s; that is, $\bar{y}_n = \sum_{j=1}^{n} s_j$. Then, we corrupt the signal by impulsive noise which is also distributed according to a compound distribution with λ' and $a' = \max_n |\bar{y}_n|$. This means that $y_n = \bar{y}$ with probability $e^{-\lambda'}$ as well as that y_n is uniformly distributed in $[-a, a]$ with probability $1 - e^{-\lambda'}$. In the case of circle-valued data, we create a random signal in the same way with $a = a' = \pi$. Then we consider the corresponding signal as phase angle. Figure 3 shows the realization of such stochastic processes and their L^1-TV estimates.

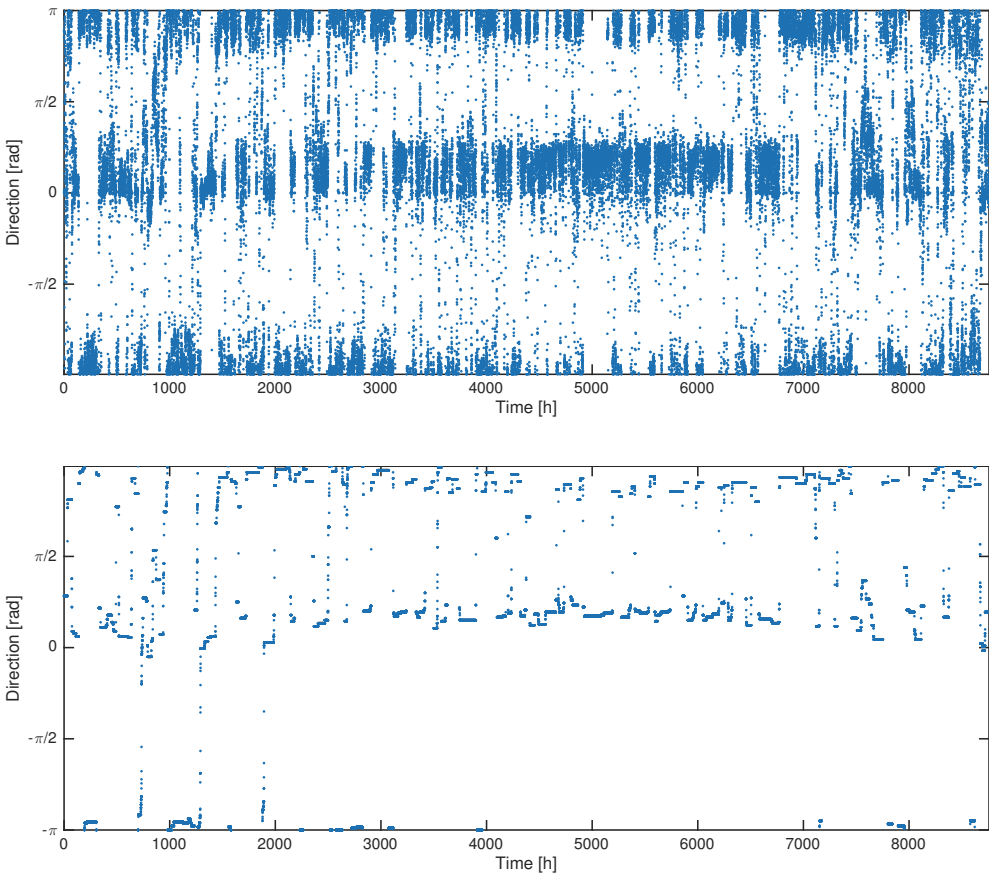

FIG. 4. *Top: Wind directions at Station WPOW*1 *(West Point, WA) recorded every* 10 *minutes* 2014. *Bottom: TV regularization with parameter* $\alpha = 50$. *The data is given quantized to* $K = 360$ *angles. The time computation amounts to only* 19.5 *seconds for the signal of length* $N = 52{,}543$.

Real-life data: Estimation of wind orientations. Next, we apply our algorithm to real-life data. The first data set consists of wind directions at the station WPOW1 (West Point, WA) recorded every 10 minutes in 2014. The second data set consists of wind directions at the station VENF1 (Venice, FL) recorded every 60 minutes in the same year.[1] The data is given quantized to integer angles in degrees, thus $K = 360$. The regularized signal facilitates identifying the time intervals of an approximately constant wind direction. For the estimate of the first data set (Figure 4), we observe a relatively regular and sudden alternation of the wind orientation between around 0.5 and 2.9 radians every third to fifth day. For the estimate of the second data set (Figure 5), we observe an inclination toward the orientation angle 0.9 radians in the middle of the year and to 0.8 radians in the months of autumn. Despite the lengths of the signals (N=52,543 and $N = 8755$), the computational times amount to only around 20 seconds and around 3 seconds, respectively.

[1]Data are available at http://www.ndbc.noaa.gov/historical_data.shtml.

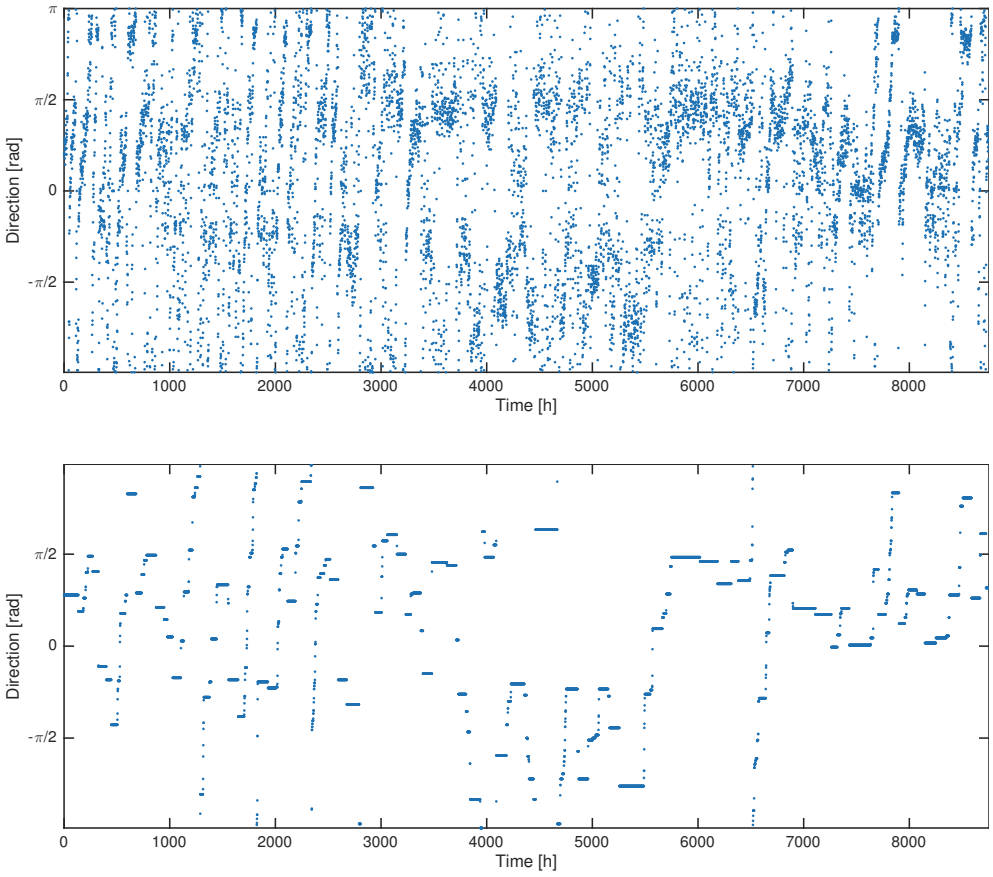

FIG. 5. *Top: Wind directions at Station VENF1 (Venice, FL) recorded every* 60 *minutes in* 2014. *Bottom: TV regularization with parameter* $\alpha = 20$ *(*$N = 8755$*; CPU time:* 3.3 *seconds).*

5. Discussion. We have derived exact algorithms for the L^1-TV problem with scalar and circle-valued data. A first crucial point was the reduction of the search space to a finite set which allowed us to employ the Viterbi algorithm. The second key ingredient was a reduction of the computational complexity based on a generalization of distance transforms. The algorithms have quadratic complexity in the worst case. The complexity is linear when the signal is quantized to a finite set. We note that quantized signals appear frequently in practice, for example, in digitalized audio signals and images, or when angular data is given in integer degrees as in the considered time series of wind directions.

The circular version is the first exact solver for TV regularization of circle-valued signals. Besides the application for jump preserving denoising of angular signals, it can also be used as a building block for higher-dimensional problems as a in [39] or as a benchmark for iterative strategies, e.g., for those of [12, 38, 25].

Next we discuss the differences to previously proposed exact solvers for the real-valued case. The solver of Dümbgen and Kovac [17] is based on a generalization of the taut string algorithm combining isotonic and antitonic regression functions which is quite distinct from our approach. The recent paper of Kolmogorov, Pock, and

Rolinek [26] seems at first glance to be related to our method because it also utilizes dynamic programming. However, the strategy is fundamentally different: in [26], the algorithm is based on dynamically removing and appending breakpoints, whereas our method performs an efficient scanning over the elements of the finite search space V^N. The solvers of [17] and [26] have complexity $\mathcal{O}(N \log N)$ and $\mathcal{O}(N \log \log N)$, respectively. Our method is competitive in terms of algorithmic complexity when the data is quantized, which leads to the complexity $\mathcal{O}(N)$.

The proposed approach appears to be unique for L^1 data terms. In particular, we have provided counterexamples that the utilized search space reduction is not valid for quadratic data terms. An exact and efficient algorithm for L^2-TV regularization of circle-valued signals remains as an open question.

REFERENCES

[1] S. Alliney, *A property of the minimum vectors of a regularizing functional defined by means of the absolute norm*, IEEE Trans. Signal Process., 45 (1997), pp. 913–917.

[2] M. Baust, L. Demaret, M. Storath, N. Navab, and A. Weinmann, *Total variation regularization of shape signals*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2075–2083.

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

[4] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[5] R. Bellman and R. Roth, *Curve fitting by segmented straight lines*, J. Amer. Statist. Assoc., 64 (1969), pp. 1079–1084.

[6] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, Mass., 1987.

[7] 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.

[8] T. Chan and S. Esedoglu, *Aspects of total variation regularized L^1 function approximation*, SIAM J. Appl. Math., 65 (2005), pp. 1817–1837.

[9] T. Chan, S. Kang, and J. Shen, *Total variation denoising and enhancement of color images based on the CB and HSV color models*, J. Visual Commun. Image Represent., 12 (2001), pp. 422–435.

[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] L. Condat, *A direct algorithm for 1-D total variation denoising*, IEEE Signal Proces. Lett., 20 (2013), pp. 1054–1057.

[12] D. Cremers and E. Strekalovskiy, *Total cyclic variation and generalizations*, J. Math. Imaging Vision, 47 (2013), pp. 258–277.

[13] J. Darbon and M. Sigelle, *Image restoration with discrete constrained total variation part* I: *Fast and exact optimization*, J. Math. Imaging Vision, 26 (2006), pp. 261–276.

[14] P. Davies and A. Kovac, *Local extremes, runs, strings and multiresolution*, Ann. Statist., 29 (2001), pp. 1–65.

[15] J. Davis and R. Sampson, *Statistics and Data Analysis in Geology*, Wiley, New York, 2002.

[16] Y. Dong, M. Hintermüller, and M. Neri, *An efficient primal-dual method for L^1 TV image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 1168–1189.

[17] L. Dümbgen and A. Kovac, *Extensions of smoothing via taut strings*, Electron. J. Stat., 3 (2009), pp. 41–75.

[18] P. Felzenszwalb and D. Huttenlocher, *Distance Transforms of Sampled Functions*, Technical report, Cornell University, Ithaca, NY, 2004.

[19] P. Felzenszwalb and D. Huttenlocher, *Efficient belief propagation for early vision*, Int. J. Comput. Vis., 70 (2006), pp. 41–54.

[20] P. Felzenszwalb and R. Zabih, *Dynamic programming and graph algorithms in computer vision*, IEEE Trans. Pattern Anal. Machine Intelligence, 33 (2011), pp. 721–740.

[21] G. Forney, Jr., *The Viterbi algorithm*, Proc. IEEE, 61 (1973), pp. 268–278.

[22] 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.

[23] M. Giaquinta, G. Modica, and J. Souček, *Variational problems for maps of bounded variation with values in S^1*, Calc. Var., 1 (1993), pp. 87–121.

[24] T. Goldstein and S. Osher, *The split Bregman method for L^1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.
[25] P. Grohs and M. Sprecher, *Total Variation Regularization by Iteratively Reweighted Least Squares on Hadamard Spaces and the Sphere*, Technical Report 2014-39, Seminar für Angewandte Mathematik, Eidgenössische Technische Hochschule, 2014.
[26] V. Kolmogorov, T. Pock, and M. Rolinek, *Total Variation on a Tree*, preprint, arXiv:1502.07770, 2015.
[27] J. Lellmann, E. Strekalovskiy, S. Koetter, and D. Cremers, *Total variation regularization for functions with values in a manifold*, in proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013, pp. 2944–2951.
[28] E. Mammen and S. van de Geer, *Locally adaptive regression splines*, Ann. Statist., 25 (1997), pp. 387–413.
[29] C. Micchelli, L. Shen, Y. Xu, and X. Zeng, *Proximity algorithms for the L^1/TV image denoising model*, Adv. Comput. Math., 38 (2013), pp. 401–426.
[30] M. Nikolova, *Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers*, SIAM J. Numer. Anal., 40 (2002), pp. 965–994.
[31] T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[32] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[33] Y. Sowa, A. D. Rowe, M. C. Leake, T. Yakushi, M. Homma, A. Ishijima, and R. M. Berry, *Direct observation of steps in rotation of the bacterial flagellar motor*, Nature, 437 (2005), pp. 916–919.
[34] M. Storath, A. Weinmann, and M. Unser, *Jump-penalized least absolute values estimation of scalar or circle-valued signals*, submitted.
[35] M. Unser and P. Tafti, *An Introduction to Sparse Stochastic Processes*, Cambridge University Press, Cambridge, UK, 2014.
[36] A. Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Trans. Inform. Theory, 13 (1967), pp. 260–269.
[37] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.
[38] A. Weinmann, L. Demaret, and M. Storath, *Total variation regularization for manifold-valued data*, SIAM J. Imaging Sci., 7 (2014), pp. 2226–2257.
[39] A. Weinmann, L. Demaret, and M. Storath, *Mumford-Shah and Potts regularization for manifold-valued data*, J. Math. Imaging Vision, 2016, DOI:10.1007/s10851-015-0628-2.
[40] A. Weinmann, M. Storath, and L. Demaret, *The L^1-Potts functional for robust jump-sparse reconstruction*, SIAM J. Numer. Anal., 53 (2015), pp. 644–673.