IEEE SIGNAL PROC. LETTERS, VOL. 20, NO. 11, PP. 1054–1057, 2013

A Direct Algorithm for 1D Total Variation Denoising

Laurent Condat

***Abstract*—A very fast noniterative algorithm is proposed for denoising or smoothing one-dimensional discrete signals, by solving the total variation regularized least-squares problem or the related fused lasso problem. A C code implementation is available on the web page of the author.**

***Index Terms*—Total variation, denoising, nonlinear smoothing, fused lasso, regularized least-squares, nonparametric regression, convex nonsmooth optimization, taut string**

I. Introduction

The problem of smoothing a signal, to remove or at least attenuate the noise it contains, has numerous applications in communications, control, machine learning, and many other fields of engineering and science [1]. In this paper, we focus on the numerical implementation of total variation (TV) denoising for one-dimensional (1D) discrete signals; that is, we are given a (noisy) signal $y = (y[1], \ldots, y[N]) \in \mathbb{R}^N$ of size $N \geq 1$, and we want to efficiently compute the denoised signal $x^\star \in \mathbb{R}^N$, defined implicitly as the solution to the minimization problem

$$\underset{x\in\mathbb{R}^N}{\text{minimize}}\ \frac{1}{2}\sum_{k=1}^{N}\big|y[k]-x[k]\big|^2+\lambda\sum_{k=1}^{N-1}\big|x[k+1]-x[k]\big|, \quad (1)$$

for some regularization parameter $\lambda \geq 0$ (whose choice is a difficult problem by itself [2]). We recall that, as the functional to minimize is strongly convex, the solution $x^\star$ to the problem exists and is unique, whatever the data y. The TV denoising problem has received large attention in the communities of signal and image processing, inverse problems, sparse sampling, statistical regression analysis, optimization theory, among others. It is not the purpose of this paper to review the properties of the nonlinear TV denoising filter, since numerous papers can be found on this vast topic; see, e.g., [3]–[6] for various insights.

A more general problem, which encompasses TV denoising as a particular case, is the *fused lasso signal approximator*, introduced in [7], which yields a solution that has sparsity in both the coefficients and their successive differences. It consists in solving the problem

$$\underset{z\in\mathbb{R}^N}{\text{minimize}}\ \frac{1}{2}\sum_{k=1}^{N}\big|z[k]-y[k]\big|^2+\lambda\sum_{k=1}^{N-1}\big|z[k+1]-z[k]\big|+\mu\sum_{k=1}^{N}\big|z[k]\big|, \quad (2)$$

for some $\lambda \geq 0$ and $\mu \geq 0$. The fused lasso has many applications, e.g. in bioinformatics [8]–[10]. As shown in [9], the complexity of the fused lasso is the same as TV denoising, since the solution $z^\star$ can be obtained by simple soft-thresholding from the solution $x^\star$ of (1):

$$z^\star[k] = \begin{cases} x^\star[k] - \mu.\text{sign}(x^\star[k]) & \text{if } |x^\star[k]| > \mu \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

It is straightforward to add soft-thresholding steps to the proposed algorithm, for essentially the same computation time. So, for simplicity of the exposition, we focus on the TV denoising problem (1) in the sequel.

To solve the convex nonsmooth optimization problem (1), we mostly find in the literature iterative fixed-point methods [11]. Until not so long ago, such methods applied to TV regularization had rather high computational complexity [12]–[16], but the growing interest for related ℓ_1-norm problems in compressed sensing or sparse recovery [17], [18] has yielded advances in the field. Recent iterative methods based on operator splitting, which exploit both the primal and dual formulations of the problems and use variable stepsize strategies or Nesterov-style accelerations, are quite efficient when applied to TV-based problems [19]–[21]. Graph cuts methods can also be used to solve (1) or its extension on graphs [22]; they actually solve a quantized version of (1): the minimizer $x^\star$ is not searched in $\mathbb{R}^N$ but in $\varepsilon\mathbb{Z}^N$, for some $\varepsilon > 0$, with complexity $O(\log_2(1/\varepsilon)N)$. If ε is small enough, the exact solution in $\mathbb{R}^N$ can be obtained from the quantized one, as shown by Hochbaum [23], [24]. In this paper, we present a novel and very fast algorithm to compute the denoised signal $x^\star$ solution to (1), exactly, in a

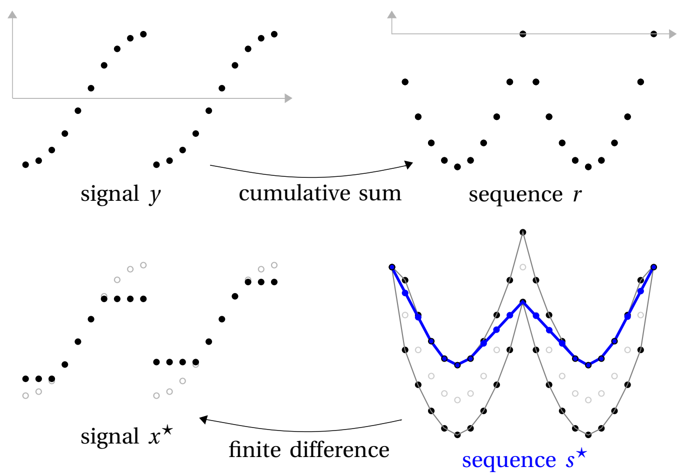

Fig. 1. Total variation denoising can be interpreted as pulling the discrete primitive r of the signal y taut in a tube around it. The taut string (blue polyline) interpolates the sequence $s^\star$ solution to (4) (blue dots), which after discrete differentiation yields the denoised sequence $x^\star$ solution to (1). The proposed algorithm is not based on this interpretation.

L. Condat is with GIPSA-lab, a joint research unit of the CNRS and the University of Grenoble, France. Contact: see http://www.gipsa-lab.fr/~laurent.condat/

1D TV Denoising Algorithm

Input: integer size $N \geq 1$, real sequence $(y[1],\dots,y[N])$, real parameter $\lambda > 0$. Output: real sequence $(x^\star[1],\dots,x^\star[N])$ solution to (1).

1. Set $k = k_0 = k_- = k_+ \leftarrow 1$, $v_{\min} \leftarrow y[1] - \lambda$, $v_{\max} \leftarrow y[1] + \lambda$, $u_{\min} \leftarrow \lambda$, $u_{\max} \leftarrow -\lambda$.
2. If $k = N$, set $x^\star[N] \leftarrow v_{\min} + u_{\min}$ and terminate.
3. If $y[k+1] + u_{\min} < v_{\min} - \lambda$, set $x^\star[k_0] = \cdots = x^\star[k_-] \leftarrow v_{\min}$, $k = k_0 = k_- = k_+ \leftarrow k_- + 1$, $v_{\min} \leftarrow y[k]$, $v_{\max} \leftarrow y[k] + 2\lambda$, $u_{\min} \leftarrow \lambda$, $u_{\max} \leftarrow -\lambda$.
4. Else, if $y[k+1] + u_{\max} > v_{\max} + \lambda$, set $x^\star[k_0] = \cdots = x^\star[k_+] \leftarrow v_{\max}$, $k = k_0 = k_- = k_+ \leftarrow k_+ + 1$, $v_{\min} \leftarrow y[k] - 2\lambda$, $v_{\max} \leftarrow y[k]$, $u_{\min} \leftarrow \lambda$, $u_{\max} \leftarrow -\lambda$.
5. Else, set $k \leftarrow k+1$, $u_{\min} \leftarrow u_{\min} + y[k] - v_{\min}$ and $u_{\max} \leftarrow u_{\max} + y[k] - v_{\max}$.
6. If $u_{\min} \geq \lambda$, set $v_{\min} \leftarrow v_{\min} + (u_{\min} - \lambda)/(k - k_0 + 1)$, $u_{\min} \leftarrow \lambda$, $k_- \leftarrow k$.
   If $u_{\max} \leq -\lambda$, set $v_{\max} \leftarrow v_{\max} + (u_{\max} + \lambda)/(k - k_0 + 1)$, $u_{\max} \leftarrow -\lambda$, $k_+ \leftarrow k$.
7. If $k < N$, go to 3.
8. If $u_{\min} < 0$, set $x^\star[k_0] = \cdots = x^\star[k_-] \leftarrow v_{\min}$, $k = k_0 = k_- \leftarrow k_- + 1$, $v_{\min} \leftarrow y[k]$, $u_{\min} \leftarrow \lambda$, $u_{\max} \leftarrow y[k] + \lambda - v_{\max}$. Then go to 2.
9. Else, if $u_{\max} > 0$, set $x^\star[k_0] = \cdots = x^\star[k_+] \leftarrow v_{\max}$, $k = k_0 = k_+ \leftarrow k_+ + 1$, $v_{\max} \leftarrow y[k]$, $u_{\max} \leftarrow -\lambda$, $u_{\min} \leftarrow y[k] - \lambda - v_{\min}$. Then go to 2.
10. Else, set $x^\star[k_0] = \cdots = x^\star[N] \leftarrow v_{\min} + u_{\min}/(k - k_0 + 1)$ and terminate.

direct, noniterative, way, possibly in-place. It is appropriate for real-time processing of an incoming stream of data, as it locates the jumps in $x^\star$ one after the other by forward scans, almost online. The possibility of such an algorithm sheds light on the relatively local nature of the TV denoising filter [25].

After this work was completed, the author found that there already exists a direct, linear time, method for 1D TV denoising, called the *taut string algorithm* [26], see also [27]–[31]. Although known by some statisticians, this method seems to be largely ignored, at least in the signal processing community. Evidence of this is that iterative methods are regularly proposed for 1D TV denoising [32]–[34]. To understand the principle of the taut string method, define the sequence of running sums r by $r[k] = \sum_{i=1}^{k} y[i]$ for $1 \leq k \leq N$, and consider the problem:

$$\underset{s\in\mathbb{R}^{N+1}}{\text{minimize}} \sum_{k=1}^{N} \sqrt{1 + \left|s[k] - s[k-1]\right|^2} \quad \text{subject to}$$
$$s[0] = 0,\ s[N] = r[N],\ \text{and} \max_{1\leq k\leq N} \left|s[k] - r[k]\right| \leq \lambda. \tag{4}$$

Then, the problems (1) and (4) are equivalent, in the sense that their respective solutions $x^\star$ and $s^\star$ are related by $x^\star[k] = s^\star[k] - s^\star[k-1]$, for $1 \leq k \leq N$ [26]. Thus, the formulation (4) allows to express the TV solution $x^\star$ as the discrete derivative of a string threaded through a tube around the discrete primitive of the data, and pulled taut such that its length is minimized. This principle is illustrated in Fig. 1. The taut string algorithm [26] is directly based on this formulation; it consists in alternating between the computation of the greatest convex minorant and least concave majorant of the upper and lower strings $r + \lambda$ and $r - \lambda$. By contrast, the proposed algorithm does not manipulate any running sum, does not require any auxiliary memory buffer, and only performs forward scans. We describe it and discuss its performances in the next section.

II. Proposed Method

We first introduce the (Fenchel-Moreau-Rockafellar) *dual* problem to the *primal* problem (1) [11]:

$$\underset{u\in\mathbb{R}^{N+1}}{\text{minimize}} \sum_{k=1}^{N} \left|y[k] - u[k] + u[k-1]\right|^2 \quad \text{s.t.}$$
$$|u[k]| \leq \lambda,\ \forall k = 1,\dots,N-1,\ \text{and}\ u[0] = u[N] = 0. \tag{5}$$

Once the solution $u^\star$ to the dual problem is found, one recovers the primal solution $x^\star$ by

$$x^\star[k] = y[k] - u^\star[k] + u^\star[k-1],\ \forall k = 1,\dots,N. \tag{6}$$

Actually, the method of [14] and its accelerated version [19] solve (5) iteratively, using forward-backward splitting [11].

The Karush-Kuhn-Tucker conditions characterize the unique solutions $x^\star$ and $u^\star$ [11]. They yield, in addition to (6),

$$u^\star[0] = u^\star[N] = 0 \quad \text{and} \quad \forall k = 1,\dots,N-1,$$
$$\begin{cases} u^\star[k] \in [-\lambda,\lambda] & \text{if}\ x^\star[k] = x^\star[k+1], \\ u^\star[k] = -\lambda & \text{if}\ x^\star[k] < x^\star[k+1], \\ u^\star[k] = \lambda & \text{if}\ x^\star[k] > x^\star[k+1]. \end{cases} \tag{7}$$

Hence, the proposed algorithm consists in running forwardly through the samples $y[k]$; at location k, it tries to prolongate the current segment of $x^\star$ by $x^\star[k+1] = x^\star[k]$. If this is not possible without violating (6) and (7), it goes back to the last location where a jump can be introduced in $x^\star$, validates the current segment until this location, starts a new segment, and continues. In more details, the proposed algorithm, given at the top of the page, works as follows. The variables are initialized at Step 1. At Step 2., we are at some location k and we are building a segment starting at k_0, with value $v = x^\star[k_0] = \cdots = x^\star[k]$. v is unknown but we know the values $v_{\min}$ and $v_{\max}$ such that $v \in [v_{\min}, v_{\max}]$. The auxiliary values $u_{\min}$ and $u_{\max}$ are the values of $u^\star[k]$ in the hypothetic cases $v = v_{\min}$ and $v = v_{\max}$, respectively. Now, we are trying to prolongate the segment with $x^\star[k+1]=v$, by updating the four variables $v_{\min}$, $v_{\max}$, $u_{\min}$, $u_{\max}$, for the location $k+1$. There are three possible cases, corresponding to Steps 3., 4., 5. If the test at Step 3. is satisfied, we cannot update $u_{\min}$ without violating (6) and (7), because $v_{\min}$ is too high. This means that the assumption $x^\star[k_0]=\cdots=x^\star[k+1]$ was wrong, so that the segment must be broken, and the negative jump necessarily takes place at the last location k_- where $u_{\min}$ was equal to λ. By the same reasoning, if the test at Step 4. is satisfied, a positive jump must be introduced at the last location k_+ where $u_{\max}$ was equal to $-\lambda$. Else, at Step 5., no jump is necessary yet and we can continue. It may be necessary to update the bounds $v_{\min}$ and $v_{\max}$; this is done at Step 6. Once the end of the signal is reached, with $k=N$, we must test if the hypothesis of a segment $x^\star[k_0]=\cdots=x^\star[N]$ does not violate the condition $u^\star[N]=0$. The three possible cases correspond to Steps 8., 9., 10. If the test at Step 8. is satisfied, $v_{\min}$ is too high and a negative jump is necessary. Similarly, if the test at Step 9. is satisfied, $v_{\max}$ is too low and a positive jump is necessary. Else, a segment is constructed until the end of the signal and the algorithm terminates, either at Step 2. or at Step 10.

We note that the dual solution $u^\star$ is not computed. We can still recover it recursively from $x^\star$ using (6).

In Fig. 2, we illustrate the behavior of the algorithm by means of the taut string analogy. The main difference between the taut string algorithm [26] and the proposed algorithm, is that the former computes and maintains in memory a convex and a concave sequences bounding the string segment under construction, while the majorant and minorant are affine in our case and only represented by their slopes $v_{\min}$ and $v_{\max}$.

A. Performance Analysis

The worst case complexity of the algorithm is $O(N+(N-1)+\cdots+1)=O(N^2)$. Indeed, every added segment has size at least one, but the algorithm may have to scan all the remaining samples to validate it in one of the steps 3., 4., 8., 9. However, this worst case scenario is encountered only when $x^\star$ is a ramp with *very small* slope of order N^{-2}, except at the boundaries; for instance, consider that $\lambda=1$ and $y[1]=-2$, $y[k]=\alpha(k-2)$ for $2\leq k\leq N-1$, $y[N]=\alpha(N-3)+2$, where $\alpha=4/((N-2)(N-3))$. The solution $x^\star$ is such that $x^\star[1]=y[1]+1$, $x^\star[k]=y[k]$ for $2\leq k\leq N-1$, $x^\star[N]=y[N]-1$. Actually, such a pathological case, for which there is no interest in applying TV denoising, is only a curiosity; the complexity is $O(N)$ in all practical situations, because the segments of $x^\star$ are validated with a delay which does not depend on N.

The algorithm was implemented in C and run on a Apple laptop with a 2.3 GHz Intel Core i7 processor. The computation time was around 25ms with $N=10^6$, for various test signals and noise levels. Importantly, the computation time is insensitive to the value of λ. The taut string algorithm was implemented in C as well, by adapting Matlab code written by Lutz Dümbgen and available online[1]. In the same conditions, the computation time was around 55ms. Thus, the taut string algorithm is efficient, but the proposed algorithm outperforms it by a constant factor consistently. We also implemented the popular iterative method FISTA [19] to solve the dual problem (5). The C code was quite optimized, with only one loop of size N per iteration. As a result, on the same machine, the computation time was around $10^{-8}N$ seconds per iteration. Thus, we can consider that the proposed algorithm takes roughly the same time as three iterations of FISTA. We should keep in mind that an iterative method like FISTA may need several thousands of iterations to converge within reasonable precision, especially for large values of N and λ.

For illustration purpose, we consider the example of a noisy Lévy process [6]: $y[k]=x_0[k]+e[k]$ for $1\leq k\leq N=1000$, where $e\sim\mathcal{N}(0,\mathrm{I}_N)$ and the ground truth x_0 has a fixed value $x_0[1]$ and i.i.d. random increments $d[k]=x_0[k]-x_0[k-1]$ for $2\leq k\leq N$. We chose a sparse Bernoulli-Gaussian law for the increments, since TV denoising is close to optimality for such signals [5], [6]; that is, the probability density function of $d[k]$ is $p\delta(t)+\frac{1-p}{\sigma\sqrt{2\pi}}\exp\left(-\frac{t^2}{2\sigma^2}\right)$, $\forall t\in\mathbb{R}$, where $p=0.95$, $\sigma=4$ and $\delta(t)$ is the Dirac distribution. We found empirically that the mean squared error $\|x_0-x^\star\|_2^2/N$ is minimized for $\lambda=2$. The computation time of $x^\star$, averaged over several runs and realizations, was 25 microseconds. One realization of the experiment is depicted in Fig. 3. For this example, FISTA needs 10,000 iterations to converge within machine precision.

III. Conclusion

In this article, we proposed a direct and very fast algorithm for denoising 1D signals by total variation (TV) minimization or fused lasso approximation. Since the algorithm computes the proximity operator [11] of the 1D TV seminorm, it can be used as a basic unit within iterative splitting methods, to solve inverse problems in signal processing and imaging. This approach will be developed in a forthcoming paper.

It would be worth investigating the possibility of extending the algorithm to complex-valued or multi-valued signals [10] and to data of higher dimensions, like 2D images or graphs [29]. Besides, path-following, a.k.a. homotopy, algorithms have been proposed for ℓ_1-penalized problems; they can find the smallest value of λ and the associated $x^\star$ in (1), such that $x^\star$ has at most m segments, with complexity $O(mN)$ [9], [18], [27], [35]–[37]. Their relationship to the approach in [38] and to the proposed algorithm should be studied. This is left for future work.

[1] http://www.imsv.unibe.ch/content/staff/personalhomepages/duembgen/software/multiscale_densities/index_eng.html

References

[1] M. A. Little and N. S. Jones, "Generalized methods and solvers for noise removal from piecewise constant signals. I. Background theory & II. New methods," *Proc. R. Soc. A*, vol. 467, 2011.

[2] S. Vaiter, C.-A. Deledalle, G. Peyré, C. Dossal, and J. Fadili, "Local behavior of sparse analysis regularization: Applications to risk estimation," *Applied and Computational Harmonic Analysis*, 2013, in press.

[3] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, 1992.

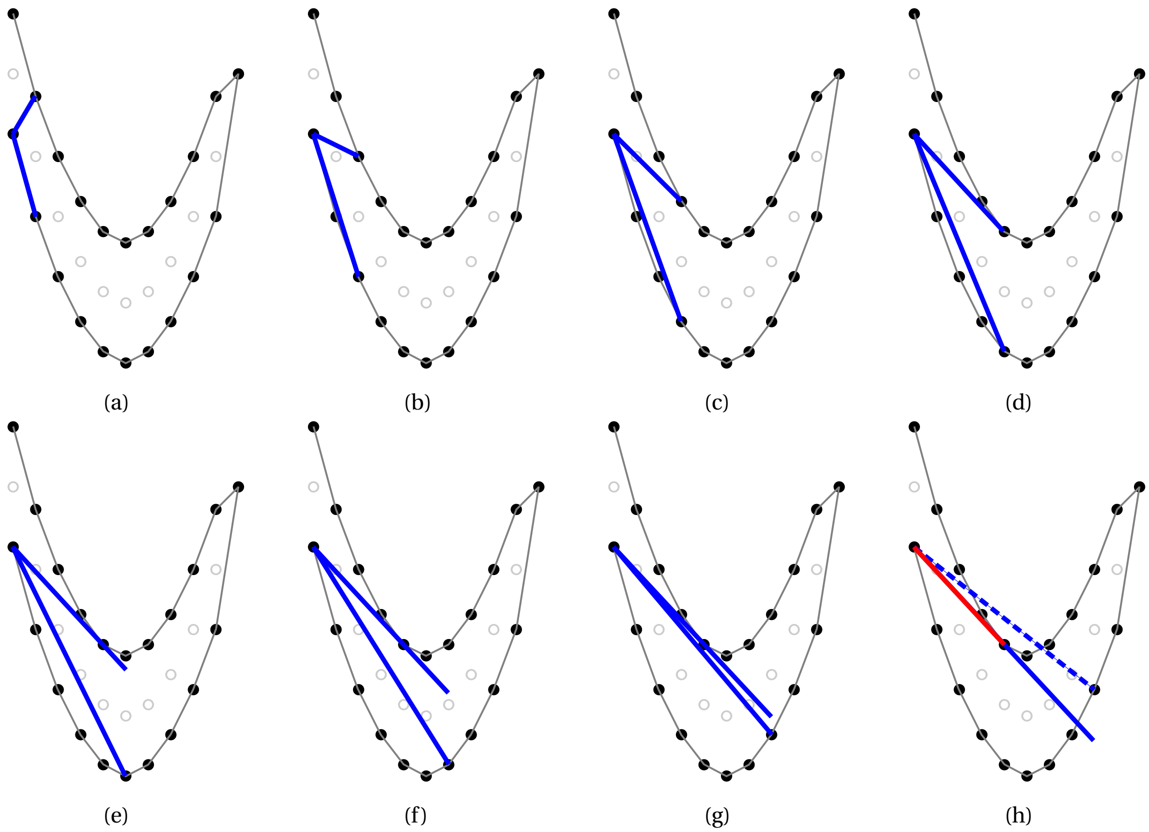

Fig. 2. Taut string interpretation of the progression of the proposed algorithm, for the construction of the segment from $x^\star[11]$ to $x^\star[14]$ in the example of Fig. 1. The values $v_{\min}$ and $v_{\max}$ are the respective slopes of the affine minorant and majorant (in blue) of the seeked string segment (in red in (h)). Each subfigure from (a) to (h) shows the progression during one pass (Steps 3 to 6) of the algorithm. The index k_+ keeps track of the last position where the majorant touches the upper string. Note that $v_{\max}$ and k_+ are updated (Step 6) in (a)–(d) but not after. The jump is detected in (h) because an update of $v_{\min}$ would violate $v_{\min} \leq v_{\max}$. Consequently, the segment ends at $k_+ = 14$.

[4] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," in *Theoretical Foundations and Numerical Methods for Sparse Recovery*, vol. 9. De Gruyter, Radon Series Comp. Appl. Math., 2010, pp. 263–340.

[5] R. Gribonval, "Should penalized least squares regression be interpreted as maximum a posteriori estimation?" *IEEE Trans. Signal Processing*, vol. 59, no. 5, pp. 2405–2410, May 2011.

[6] U. S. Kamilov, P. Pad, A. Amini, and M. Unser, "MMSE estimation of sparse Lévy processes," *IEEE Trans. Signal Processing*, vol. 61, no. 1, pp. 137–147, Jan. 2013.

[7] R. Tibshirani, M.Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. The Royal Statistical Society Series B*, vol. 67, no. 1, pp. 91–108, 2005.

[8] R. Tibshirani and P. Wang, "Spatial smoothing and hot spot detection for CGH data using the fused lasso," *Biostatistics*, vol. 9, pp. 18–29, 2008.

[9] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Ann. Appl. Statist.*, vol. 1, no. 2, pp. 302–332, 2007.

[10] K. Bleakley and J.-P. Vert, "The group fused lasso for multiple change-point detection," June 2011, preprint hal-00602121.

[11] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York: Springer, 2011.

[12] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM J. Sci. Comput.*, vol. 17, pp. 227–238, 1996.

[13] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 231–241, Feb. 2001.

[14] A. Chambolle, "Total variation minimization and a class of binary MRF models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, ser. Lecture Notes in Computer Science, 2005, vol. 3757, pp. 136–152.

[15] T. Chan, S. Esedoglu, F. Park, and M. Yip, "Recent developments in total variation image restoration," in *Mathematical Models of Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. Springer Verlag, 2005.

[16] B. Wohlberg and P. Rodriguez, "An iteratively reweighted norm algorithm for minimization of total variation functionals," *IEEE Signal Processing Lett.*, vol. 14, no. 12, pp. 948–951, 2007.

[17] M. Zibulevsky and M. Elad, "L1-L2 optimization in signal and image processing," *IEEE Signal Processing Mag.*, vol. 27, no. 3, pp. 76–88, 2010.

[18] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[19] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Processing*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[20] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[21] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.

[22] A. Chambolle and J. Darbon, "On total variation minimization and surface evolution using parametric maximal flows," *International Journal of Computer Vision*, vol. 84, no. 3, 2009.

[23] ——, "A parametric maximum flow approach for discrete total variation regularization," in *Image Processing and Analysis with Graphs: Theory and Practice*, ser. Digital Imaging and Computer Vision, R. Lukac, Ed. CRC Press / Taylor and Francis, 2012.

[24] D. S. Hochbaum, "An efficient and effective tool for image segmentation, total variations and regularization," in *Proc. of Int. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM)*, 2011, vol. LNCS 6667, pp. 338–349.

[25] C. Louchet and L. Moisan, "Total variation as a local filter," *SIAM J. on Imaging Sciences*, vol. 4, no. 2, pp. 651–694, 2011.

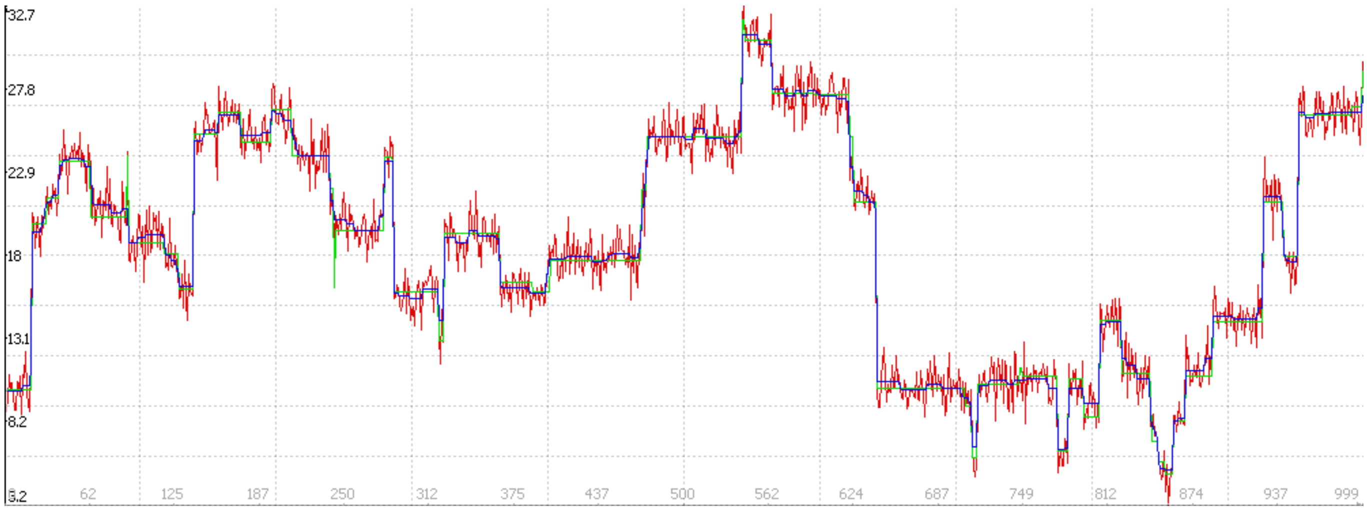

Fig. 3. In this example, y (in red) is a piecewise constant process of size $N = 1000$ (unknown ground truth, in green) corrupted by additive Gaussian noise of unit variance. The TV-denoised signal $x^{\star}$, with $\lambda = 2$, is in blue.

[26] P. L. Davies and A. Kovac, "Local extremes, runs, strings and multiresolution," *The Annals of Statistics*, vol. 29, no. 1, pp. 1–65, 2001.

[27] E. Mammen and S. van de Geer, "Locally adaptive regression splines," *The Annals of Statistics*, vol. 25, no. 1, pp. 387–413, 1997.

[28] M. Grasmair, "The equivalence of the taut string algorithm and BV-regularization," *J. Math. Imaging Vision*, vol. 27, no. 1, pp. 59–66, 2007.

[29] W. Hinterberger, M. Hintermüller, K. Kunisch, M. von Oehsen, and O. Scherzer, "Tube methods for BV regularization," *Journal of Mathematical Imaging and Vision*, vol. 19, no. 3, pp. 219–235, 2003.

[30] G. Steidl, J. Weickert, T. Brox, P. Mrazek, and M. Welk, "On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs," *SIAM J. Numer. Anal.*, vol. 42, no. 2, pp. 686–713, 2004.

[31] L. Dümbgen and A. Kovac, "Extensions of smoothing via taut strings," *Electron. J. Statist.*, vol. 3, pp. 41–75, 2009.

[32] J. Liu, L. Yuan, and J. Ye, "An efficient algorithm for a class of fused lasso problems," in *Proc. of ACM SIGKDD*, 2010, pp. 323–332.

[33] A. Barbero and S. Sra, "Fast Newton-type methods for total variation regularization," in *Proc. of Int. Conf. Machine Learning (ICML)*, June 2011, pp. 313–320.

[34] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An ADMM algorithm for a class of total variation regularized estimation problems," in *Proc. of IFAC Symposium on System Identification*, July 2012.

[35] H. Höfling, "A path algorithm for the fused lasso signal approximator," *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, pp. 984–1006, 2010.

[36] R. J. Tibshirani and J. Taylor, "The solution path of the generalized lasso," *The Annals of Statistics*, vol. 39, no. 3, pp. 1335–1371, 2011.

[37] H. Zhou and K. Lange, "A path algorithm for constrained estimation," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 261–283, 2013.

[38] I. Pollak, A. S. Willsky, and Y. Huang, "Nonlinear evolution equations as fast and exact solvers of estimation problems," *IEEE Trans. Signal Processing*, vol. 53, no. 2, pp. 484–498, Feb. 2005.