Nonlinear Evolution Equations as Fast and Exact Solvers of Estimation Problems

Ilya Pollak, Alan S. Willsky, *Fellow, IEEE*, and Yan Huang

***Abstract*—We develop computationally efficient procedures for solving certain restoration problems in one dimension, including the one–dimensional (1-D) discrete versions of the total variation regularized problem introduced by Sauer and Bouman and the constrained total variation minimization problem introduced by Rudin *et al.* The procedures are exact and have time complexity $O(N \log N)$ and space complexity $O(N)$, where N is the number of data samples. They are based on a simple nonlinear diffusion equation proposed by Pollak *et al.* and related to the Perona–Malik equation. A probabilistic interpretation for this diffusion equation in 1-D is provided by showing that it produces optimal solutions to a sequence of estimation problems. We extend our methods to two dimensions, where they no longer have similar optimality properties; however, we experimentally demonstrate their effectiveness for image restoration.**

***Index Terms*—Estimation, noise removal, partial differential equations, restoration, scale-space, segmentation, SIDE, total variation.**

I. Introduction

RECENT years have seen a great number of exciting developments in the field of image reconstruction, restoration, and segmentation via nonlinear diffusion filtering (see, e.g., a survey article [17]). Since the objective of these filtering techniques is usually extraction of information in the presence of noise, their probabilistic interpretation is important. In particular, a natural question to consider is whether or not these methods solve standard estimation or detection problems. An affirmative answer would help us understand which technique is suited best for a particular application and aid in designing new algorithms. It would also put the tools of the classical detection and estimation theory at our disposal for the analysis of these techniques, making it easier to tackle an even more crucial issue of characterizing the performance of a nonlinear diffusion technique given a probabilistic noise model.

Manuscript received June 14, 2003; revised January 9, 2004. This work was supported in part by a National Science Foundation CAREER award CCR-0093105, funds from Purdue Research Foundation, the Academy of Finland Center of Excellence Program 44876, the Office of Naval Research under Grant N00014-00-1-0089, the Air Force Office of Scientific Research under Grant F49620-00-1-0362, and the Army Research Office under Grant DAAH04-96-1-0445. The associate editor coordinating the review of this paper and approving it for publication was Dr. Hamid Krim.

I. Pollak and Y. Huang are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ipollak@ecn.purdue.edu, yanh@ecn.purdue.edu).

A. S. Willsky is with the Department of Electrical Engineering and Computer Science and Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: willsky@mit.edu).

Digital Object Identifier 10.1109/TSP.2004.840786

Addressing the relationship between nonlinear diffusion filtering and optimal estimation is, however, very difficult, because the complex nature of the nonlinear partial differential equations (PDEs) used in these techniques and of the images of interest make this analysis prohibitively complicated. Some examples of the existing literature on the subject are [23] and [28], which establish qualitative relations between the Perona–Malik equation [10], [15] and gradient descent procedures for estimating random fields modeled by Gibbs distributions. Bayesian ideas are combined in [29] with snakes and region growing for image segmentation. In [4], concepts from robust statistics are used to modify the Perona–Malik equation. In [13], a connection between random walks and diffusions is used to obtain a new evolution equation.

One of the goals of this paper is to move forward the discussion of this issue by establishing that a particular nonlinear diffusion filtering method results in a maximum likelihood (ML) estimate for a certain class of signals. We expand and develop our contributions of [16] and [19], obtaining new methods for solving certain restoration problems. The methods are first developed in one dimension, where they are provably fast and provably optimal. While we do not have analytical results on the two-dimensional (2-D) generalizations of our methods, experiments show that the 2-D algorithms are efficient and robust, as well.

We concentrate most of our attention on the problem of maximum likelihood estimation in additive white Gaussian noise, subject to a constraint on the total variation (TV). We show that this problem, in one dimension, is closely related to the total variation minimization problems posed by Bouman and Sauer in [6] and [21] and by Rudin *et al.* in [20]. This choice of our prototypical problem is motivated by a great success of total variation minimization methods [1], [2], [6], [7], [20], [21], which has demonstrated a critical need for fast computational techniques [3], [5], [8], [11], [25]–[27]. A major contribution of this paper is a new fast and exact algorithm for solving the one–dimensional (1-D) discrete-time versions of these problems, as well as a fast and approximate algorithm for the 2-D versions.

In order to motivate our theoretical results and the ensuing algorithms, we start in Section II by experimentally demonstrating the effectiveness of the methods proposed in this paper. We proceed with a review of background material on nonlinear diffusions in Section III and then focus on one very simple evolution in Section IV. We show in Section V that this evolution results in a fast solver of our ML problem. Section VI summarizes the contributions of this paper and presents several future research directions.

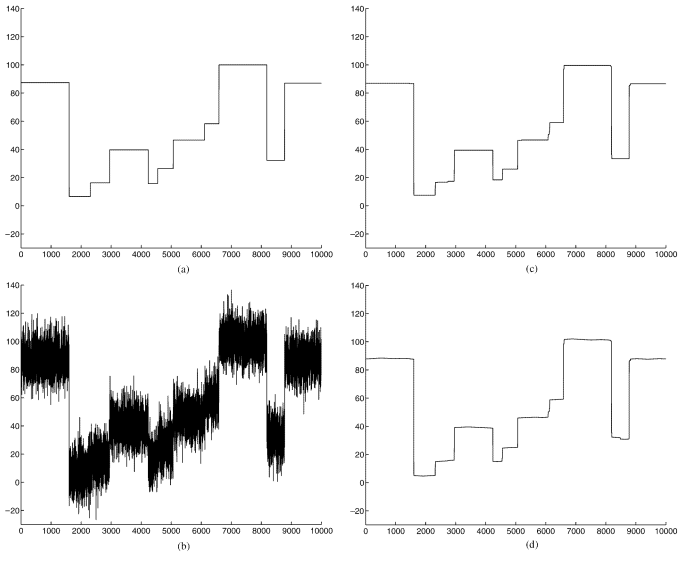


Fig. 1. Experiment in 1-D. Noise removal via constrained total variation minimization using our algorithm and the algorithm in [20]. For this 10 000-point signal, with comparable reconstruction quality and RMS errors, our algorithm takes 0.29 s on a SunBlade 1000 (750 MHz processor), whereas the algorithm of [20], with the most favorable parameter settings, takes 1005 s. (a) Noise-free signal. (b) Noisy signal. (c) Our reconstruction. (d) Reconstruction using [20].

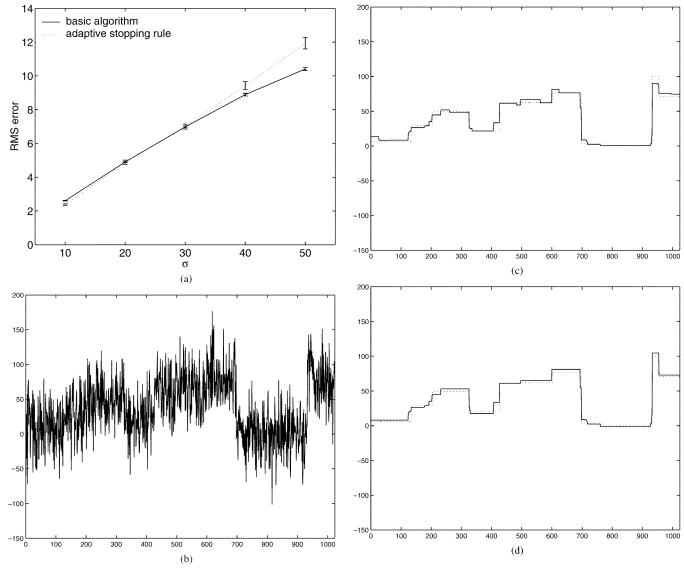


Fig. 2. Experiment in 1-D. (a) RMS errors, as a function of noise standard deviation σ, for our basic constrained total variation minimization method that uses the knowledge of σ (solid line) and for a variant of the method that does not rely on knowing σ but uses an adaptive stopping rule instead (dotted line). (b) Piecewise constant signal with additive white Gaussian noise of $\sigma = 30$. (c) Restoration using our constrained total variation minimization method with known σ (solid line) superimposed onto the noise-free signal (dotted line). (d) Restoration using our adaptive stopping rule.

II. Experimental Results

A. Comparisons to the Original Method of Rudin–Osher–Fatemi in 1-D

The algorithms developed in this paper address the problem of noise removal both in 1-D and 2-D. In particular, the 1-D algorithm described below in Section V-C calculates the solution to the constrained total variation minimization problem originally posed by Rudin *et al.* in [20]. Fig. 1 illustrates the comparison of our method to the original method of [20]. Fig. 1(a) shows a synthetic piecewise constant signal. The same signal, with additive zero-mean white Gaussian noise of standard deviation $\sigma = 10$, is shown in Fig. 1(b). This signal is processed using our algorithm, and the resulting output, which is obtained in 0.29 s on a SunBlade 1000, is shown in Fig. 1(c). Note that the only parameter that our algorithm needs is σ. In this example, we assume that we know σ exactly.

Since the method of [20] is an iterative descent method, no straightforward comparison is possible. Indeed, in addition to σ, that algorithm needs two other parameters: the time step and the stopping criterion. Choosing the time step *a priori* is a difficult task. Generally, the noisier the data, the larger the number of iterations, and therefore, smaller time steps will be required for noisier data to achieve a given precision. To make the comparison as tough for our own method as possible, we chose the largest time step for the algorithm of [20] that can result in the reconstruction quality that is comparable with that of our method. Specifically, the parameters were chosen to produce a root mean square (RMS) error per sample of 1.2 (as compared with 0.9 for our algorithm). We also assume here that the correct value of σ is known. The result, for the input signal of Fig. 1(b), is shown in Fig. 1(d). It took 1005 s to compute the solution, which is in excess of three orders of magnitude more computation time than our method.

TABLE I
Comparison of Computation Times for 50 1024-Point Signals

Algorithm	Mean time	St. dev.	Fastest time	Slowest time
Proposed	0.06	0.004	0.05	0.07
[20]	6.96	6.09	1.11	24.32

The advantages of our method illustrated with this simple example also apply when comparing it with other, more recent approaches to solving the TV minimization problem or related problems (e.g., [3], [5]–[8], [11], [14], [20], [25]–[27]). An important advantage of our method is that a bound on the overall computational complexity is available. Thanks to Proposition 7 of Section V below, our 1-D algorithm is guaranteed to achieve the solution of the Rudin–Osher–Fatemi problem *exactly* (modulo computer precision) in a *finite* number of steps, which is $O(N \log N)$.

In addition, our method relies on only one parameter σ. Moreover, as illustrated in Fig. 2, the requirement to know σ, which tells our recursive algorithm when to stop, can be dispensed with and replaced with an adaptive stopping rule, with essentially no sacrifice in performance.

To further compare our method with that of [20] in one dimension, we refer to Table I, which details the results of a Monte Carlo simulation. Fifty randomly generated piecewise constant signals of length $N = 1024$ with additive white Gaussian noise $(\text{SNR} = 1)$ are used in the experiment. For each signal, we run our algorithm and calculate the RMS error e. We then try the descent algorithm of [20] with several different time steps and choose the time step for which the algorithm can reach the RMS error of $1.1e$ (i.e., be within 10% of our RMS error) with the

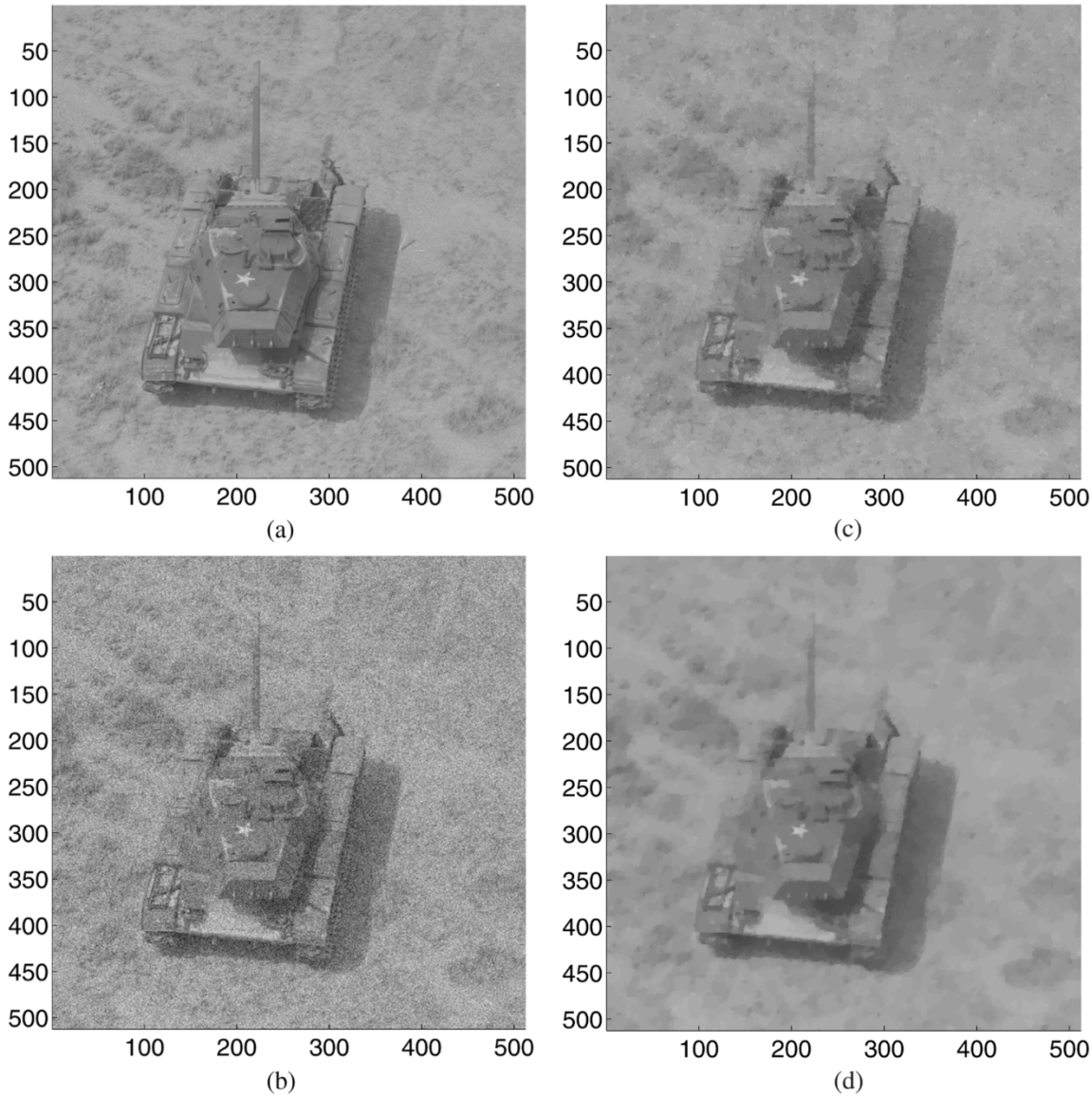


Fig. 3. Experiment in 2-D. Noise removal via constrained total variation minimization using our algorithm and algorithm in [20]. With comparable reconstruction quality and RMS errors, our algorithm takes 14 s (on a SunBlade 1000 750-MHz processor), whereas the algorithm of [20] takes 151 s. (a) Image. (b) Noisy image. (c) Our reconstruction. (d) Reconstruction using [20].

fewest iterations. (If it cannot reach $1.1e$ in 150 000 iterations, which happened in three out of 50 cases, we stop.) We then look at the computation time for the algorithm of [20], with the optimal time step. The results of this experiment are summarized in Table I, showing once again that our 1-D method is much faster: The fastest run of the algorithm of [20] is more than an order of magnitude slower than the slowest run of our algorithm.

We now test an adaptive version of our algorithm, which is outlined below in Section V. Our algorithm is a recursive procedure: When the standard deviation σ of the noise is known, this parameter can be used to stop the recursion. If, however, σ is unknown, we can apply a different, adaptive, criterion to determine when to stop. As Fig. 2 shows, the RMS errors for the adaptive algorithm are very similar to those for the case when the correct value of σ is known. Moreover, since we use the same basic procedure to calculate the solution, the computational complexity of the adaptive method is the same.

B. Experiments in 2-D

While our 1-D algorithms extend to 2-D, our main theoretical results do not. Propositions 5–7 of Section V do not hold in 2-D, i.e., we can no longer claim that our algorithm exactly solves the respective optimization problems in 2-D. Moreover, it can be shown that the time complexity of the 2-D version of our algorithm is actually $O(N^2 \log N)$ [not $O(N \log N)$], where N is the total number of pixels in the image. Our algorithm nevertheless solves these problems approximately and still has a bound on computational complexity. We have also observed that its actual time complexity is typically lower than the asymptotic bound of $O(N^2 \log N)$. To illustrate our algorithm, we use two images: one image from [20], shown in Fig. 3(a), and another image shown in Fig. 4(a). The two images that are corrupted by zero-mean additive white Gaussian noise are shown in Figs. 3(b) and 4(b), respectively. The noise variance for the tank image is

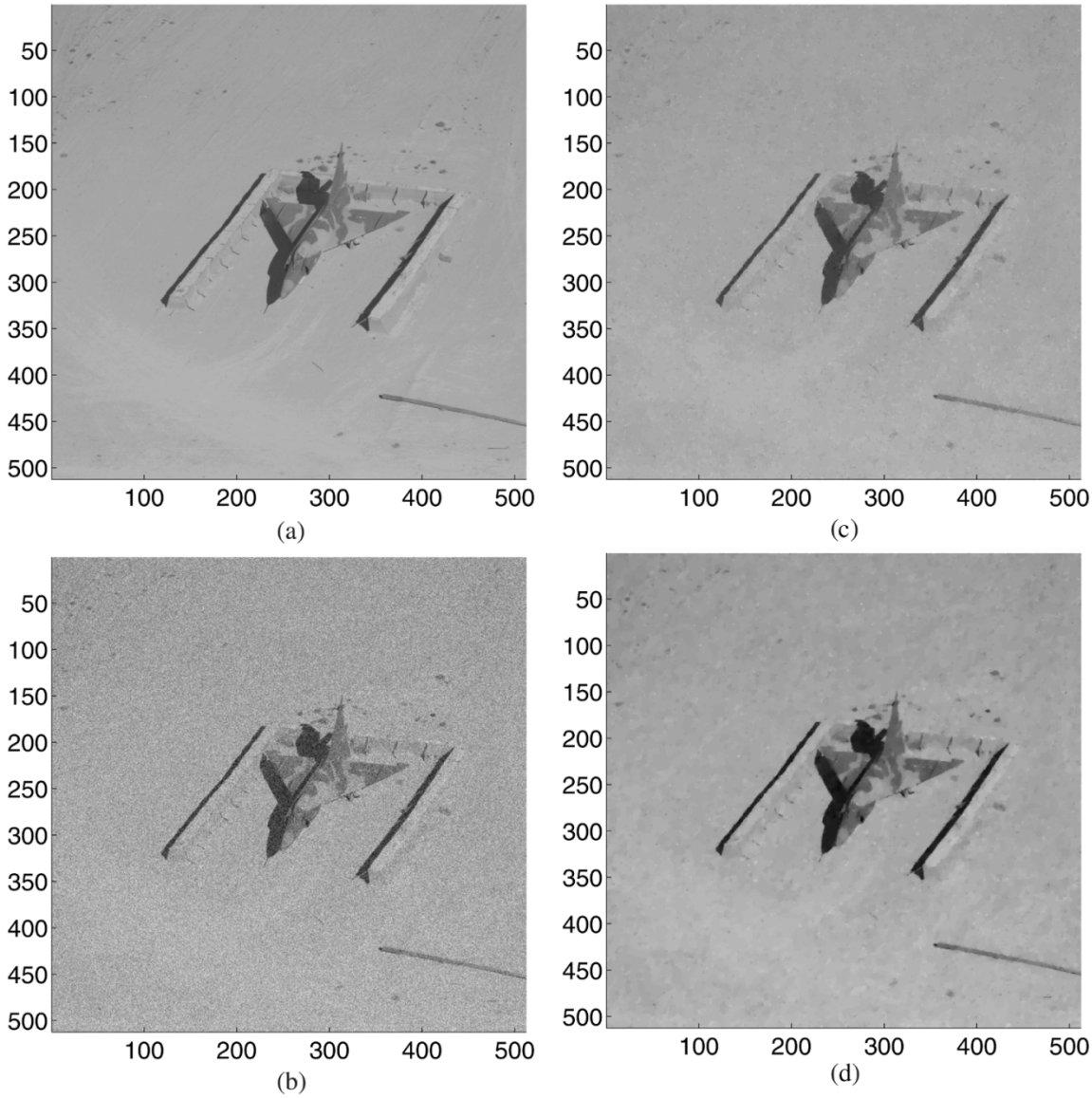


Fig. 4. Another 2-D experiment. With comparable reconstruction quality and RMS errors, our algorithm takes 14 s (on a SunBlade 1000 750-MHz processor), whereas the algorithm of [20] takes 54 s. (a) Image. (b) Noisy image. (c) Our reconstruction. (d) Reconstruction using [20].

chosen to achieve the same SNR as in the original paper [20]. The outputs of our algorithm[1] are in Figs. 3(c) and 4(c). The computation time is approximately 14 s in both cases. We use the parameter settings in the algorithm of [20] to achieve a comparable RMS error, in both cases, and the results are depicted in Figs. 3(d) and 4(d). The computation times were 151 s for the tank image and 54 s for the airplane image. To better illustrate the reconstructions, we provide in Fig. 5 enlarged versions of two patches from the images of Fig. 3. Fig. 5(a) illustrates where the two patches are taken from. Fig. 5(b)–(e) is the first patch: original, noisy, restored with our method, and restored with the method of [20], respectively. Fig. 5(f)–(i) contains similar images for the second patch. The visual quality of the two reconstructions is similar; both methods are good at recovering edges and corners.

[1]We only show the outputs of the adaptive version of our algorithm since the results when the noise variance is known are similar.

III. Background, Notation, and Definitions

A. Nonlinear Diffusions

The basic problem considered in this paper is restoration of noisy 1-D signals and 2-D images. We build on the results in [18], where a family of systems of ordinary differential equations, called Stabilized Inverse Diffusion Equations (SIDEs), was proposed for restoration, enhancement, and segmentation of signals and images. The (discretized) signal or image $\mathbf{u}^0$ to be processed is taken to be the initial condition for the equation, and the solution $\mathbf{u}(t)$ of the equation provides a fine-to-coarse family of segmentations of the image. This family is indexed by the "scale" (or "time") variable t, which assumes values from 0 to ∞.

The usefulness of SIDEs for segmentation was shown in [18]; in particular, it was experimentally demonstrated that SIDEs are robust to noise outliers and blurring. They are faster

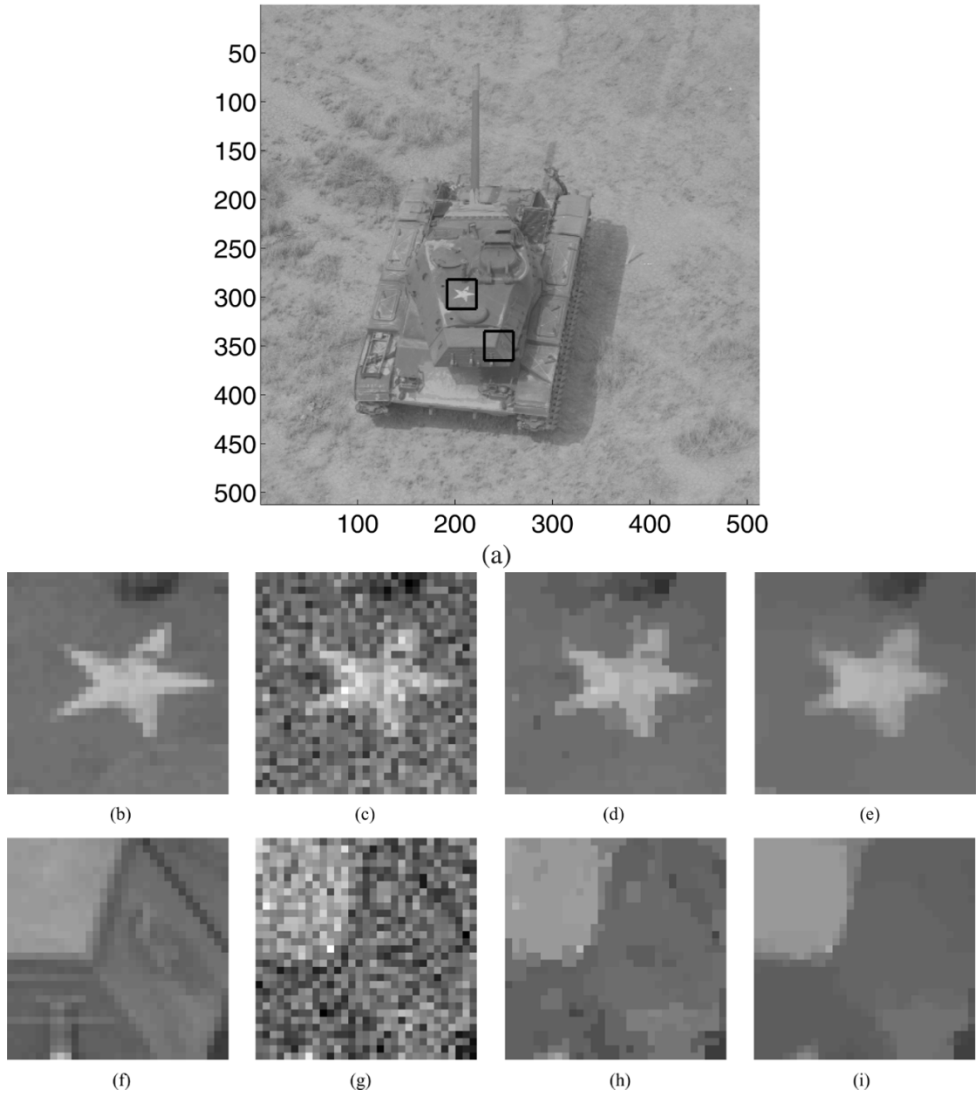


Fig. 5. Zooming in on the tank experiment (a) Two patches. (b)-(e) Enlarged versions of the images in Fig. 3 for the first patch. (f)–(i) Enlarged versions of the images in Fig. 3 for the second patch. (b) Patch. (c) Noisy patch. (d) Our reconstruction. (e) Reconstruction using [20]. (f) Patch. (g) Noisy patch. (h) Our reconstruction. (i) Reconstruction using [20].

TABLE II
NOTATION FOR THE SEGMENTATION PARAMETERS OF A SIGNAL

Segmentation parameters of signal $\mathbf{u}$	Notation
Number of regions.	$p(\mathbf{u})$
Index of the left endpoint of region i.	$n_i(\mathbf{u})$
Size of region i.	$m_i(\mathbf{u})$
Number of neighbors of region i.	$\rho_i(\mathbf{u})$
Is region i a max, a min, or neither?	$\beta_i(\mathbf{u})$

than other image processing algorithms based on evolution equations, since region merging reduces the dimensionality of the system during evolution. In [12], SIDEs were successfully incorporated as part of an algorithm for segmenting dermatoscopic images.

Our first effort in probabilistic analysis of SIDEs is [19], where we showed that a simple SIDE optimally solves certain detection problems. In the present paper, we prove that a SIDE is also an optimal solver for several estimation problems.

B. Notation in 1-D

In this subsection, we introduce notation that will be used throughout the paper to analyze our 1-D SIDE.

The number of samples in the signals under consideration is always denoted by N. The signals themselves are denoted by boldface lowercase letters and viewed as vectors in $\mathbb{R}^N$. The samples of a signal are denoted by the same letter as the signal itself, but normal face, with subscripts 1 through N, e.g., $\mathbf{u} = (u_1, \ldots, u_N)^T$. We say that u_n is the *intensity* of the nth sample of $\mathbf{u}$.

A set of the indexes $\{n, n+1, \ldots, k\}$ of consecutive samples of a signal $\mathbf{u}$, which have equal intensities $u_n = u_{n+1} = \cdots = u_k$, is called a *region* if this set cannot be enlarged [in other words, if $u_{n-1} \neq u_n$ (or $n = 1$) and $u_k \neq u_{k+1}$ (or $k = N$)]. Any pair of consecutive samples with unequal intensities is called an *edge*. The number of distinct regions in a signal $\mathbf{u}$ is

denoted by $p(\mathbf{u})$. The indexes of the left endpoints of the regions are denoted by $n_i(\mathbf{u}), i = 1, \ldots, p(\mathbf{u})$ (the n_is are ordered from left to right: $n_1(\mathbf{u}) < n_2(\mathbf{u}) < \cdots < n_{p(\mathbf{u})}(\mathbf{u})$); the intensity of each sample within region i is denoted by $\mu_i(\mathbf{u})$ and is referred to as the *intensity* of region i. This means that $n_1(\mathbf{u}) = 1$ and that

$$\mu_i(\mathbf{u}) = u_{n_i(\mathbf{u})} = \cdots = u_{n_{i+1}(\mathbf{u})-1}, \text{ for } i = 1, \ldots, p(\mathbf{u})$$

where we use the convention $n_{p(\mathbf{u})+1} = N + 1$.

The length $m_i(\mathbf{u})$ of the ith region of $\mathbf{u}$ (i.e., the number of samples in the region) satisfies

$$m_i(\mathbf{u}) = n_{i+1}(\mathbf{u}) - n_i(\mathbf{u}).$$

Two regions are called *neighbors* if they have consecutive indexes. The number of neighbors of the ith region of $\mathbf{u}$ is denoted by $\rho_i(\mathbf{u})$ and is equal to one for the leftmost and rightmost regions and two for all other regions:

$$\rho_i(\mathbf{u}) = \begin{cases} 1, & i = 1, p(\mathbf{u}) \\ 2, & \text{otherwise.} \end{cases}$$

We call region i a *maximum (minimum)* of a signal $\mathbf{u}$ if its intensity $\mu_i(\mathbf{u})$ is larger (smaller) than the intensities of all its neighbors.[2] Region i is an *extremum* if it is either a maximum or a minimum. We let

$$\beta_i(\mathbf{u}) = \begin{cases} 1, & \text{if region } i \text{ is a maximum of } \mathbf{u} \\ -1, & \text{if region } i \text{ is a minimum of } \mathbf{u} \\ 0, & \text{otherwise.} \end{cases}$$

The parameter $p(\mathbf{u})$ and the four sets of parameters $n_i(\mathbf{u})$, $m_i(\mathbf{u})$, $\rho_i(\mathbf{u})$, and $\beta_i(\mathbf{u})$ for $i = 1, \ldots, p(\mathbf{u})$ are crucial to both the analysis of our 1-D algorithms and the description of their implementation. Collectively, these parameters will be referred to as *segmentation parameters* of signal $\mathbf{u}$. They are summarized in Table II. When it is clear from the context which signal is being described by these parameters, we will drop their arguments, and write, for example, n_i instead of $n_i(\mathbf{u})$.

The TV of a signal $\mathbf{u} \in \mathbb{R}^N$ is defined by

$$\text{TV}(\mathbf{u}) \stackrel{\text{def}}{=} \sum_{n=1}^{N-1} |u_{n+1} - u_n|$$

and $\|\cdot\|$ stands for the usual Euclidean (ℓ^2) norm

$$\|\mathbf{u}\|^2 \stackrel{\text{def}}{=} \sum_{n=1}^{N} u_n^2.$$

The following alternative form for $\text{TV}(\mathbf{u})$ can be obtained through a simple calculation:

$$\text{TV}(\mathbf{u}) = \sum_{i=1}^{p(\mathbf{u})} \beta_i(\mathbf{u}) \rho_i(\mathbf{u}) \mu_i(\mathbf{u}). \tag{1}$$

We are now ready to describe the 1-D SIDE which is analyzed in this paper.

[2]The term "local maximum (minimum)" would be more appropriate, but we omit the word "local" for brevity.

C. A SIDE in 1-D

In [19], we provided a probabilistic interpretation for a special case of SIDEs in 1-D in the context of binary change detection problems. In Section IV, we generalize these results to restoration problems. Specifically, we show that certain restoration problems are optimally solved by evolving the following system of equations:

$$\begin{aligned} \dot{u}_k(t) = \frac{1}{m_i(\mathbf{u}(t))} \Big\{ & \text{sgn}\left[\mu_{i+1}(\mathbf{u}(t)) - \mu_i(\mathbf{u}(t))\right] \\ & - \text{sgn}\left[\mu_i(\mathbf{u}(t)) - \mu_{i-1}(\mathbf{u}(t))\right] \Big\}, \text{ for} \\ k = n_i(\mathbf{u}(t)), \ldots, & n_{i+1}(\mathbf{u}(t)) - 1, \text{ and} \\ i = 1, \ldots, & p(\mathbf{u}(t)) \end{aligned} \tag{2}$$

with the initial condition

$$\mathbf{u}(0) = \mathbf{u}^0 \tag{3}$$

where $\mathbf{u}^0$ is the signal to be processed. Note that when $i = 1$ and when $i = p(\mathbf{u}(t))$, (2) involves quantities μ_0 and $\mu_{p(\mathbf{u}(t))+1}$, which have not been defined. We use the following conventions for these quantities:

$$\begin{aligned} \text{sgn}[\mu_1(\mathbf{u}(t)) - \mu_0(\mathbf{u}(t))] &= \text{sgn}[\mu_{p+1}(\mathbf{u}(t)) - \mu_p(\mathbf{u}(t))] \\ &\stackrel{\text{def}}{=} 0. \end{aligned} \tag{4}$$

Equation (2) says that the intensities of samples within a region have the same dynamics and, therefore, remain equal to each other. A region cannot therefore be broken into two or more regions during this evolution. The opposite, however, will happen. As soon as the intensity of some region becomes equal to that of its neighbor ($\mu_j(\mathbf{u}(\tau)) = \mu_{j+1}(\mathbf{u}(\tau))$ for some j and for some time instant τ), the two become a single region; this follows from our definition of a region. This merging of two regions into one will express itself in a change of the segmentation parameters of $\mathbf{u}(t)$ in (2):

$$m_j(\mathbf{u}(\tau)) = m_j(\mathbf{u}(\tau^-)) + m_{j+1}(\mathbf{u}(\tau^-)) \tag{5}$$

$$p(\mathbf{u}(\tau)) = p(\mathbf{u}(\tau^-)) - 1 \tag{6}$$

$$n_i(\mathbf{u}(\tau)) = n_{i+1}(\mathbf{u}(\tau^-)), \text{ for } i = j+1, \ldots, p(\mathbf{u}(\tau)) \tag{7}$$

$$m_i(\mathbf{u}(\tau)) = m_{i+1}(\mathbf{u}(\tau^-)), \text{ for } i = j+1, \ldots, p(\mathbf{u}(\tau)). \tag{8}$$

Equation (5) says that the number of samples in the newly formed region is equal to the total number of samples in the two regions that are being merged. Equation (6) reflects the reduction in the total number of regions by one. Equation (7) and (8) express the fact that, since region $j+1$ is getting merged onto region j, the region that used to have index $j+2$ will now have index $j+1$; the region that used to have index $j+3$ will now have index $j+2$, etc. Borrowing our terminology from [18], we call such time instant τ when two regions get merged a *hitting time*. Note that between two consecutive hitting times, the segmentation parameters of $\mathbf{u}(t)$ remain constant. We denote the hitting times by $t_1, \ldots, t_{p(\mathbf{u}^0)-1}$, where t_1 is the earliest hitting time, and $t_{p(\mathbf{u}^0)-1} \stackrel{\text{def}}{=} t_f$ is the final hitting time:

$$0 < t_1 \leq t_2 \leq \cdots \leq t_{p(\mathbf{u}^0)-1}.$$

Note that two hitting times can be equal to each other, when more than one pair of regions are merged at the same time. Proposition 1 of Section IV shows that the evolution (2) will reach a constant signal in finite time, starting from any initial condition. All samples will therefore be merged into one region, within finite time, which means that there will be exactly $p(\mathbf{u}^0)-1$ hitting times: one fewer than the initial number of regions.

The rest of the paper analyzes the SIDE (2), with initial condition (3), conventions (4), and the ensuing merging rules (5)–(8). An example of its evolution, for $N = 50$, is depicted in Fig. 6. Fig. 6(a) shows the solution at all times from $t = 0$ until $t = 20$. The initial signal $\mathbf{u}(0)$ has $p(\mathbf{u}(0)) = 5$ regions, and therefore, there are four hitting times. They are $t_1 = 3.2$, $t_2 = 6.54$, $t_3 = 9.21$, and $t_4 = t_f = 15.53$. The initial condition and the solution at the four hitting times are plotted in Fig. 6(b).

We conclude this subsection by deriving an alternative form for (2), which will be utilized many times in the remainder of the paper. If region i is a maximum of $\mathbf{u}(t)$, (2) and conventions (4) say that each of its neighbors contributes $-1/m_i$ to the rate of change of its intensity $\mu_i(\mathbf{u}(t))$. Similarly, if region i is a minimum, each neighbor contributes $1/m_i$. If region i is not an extremum, then it necessarily has two neighbors, one of which contributes $1/m_i$ and the other one $-1/m_i$. In this latter case, the rate of change of $\mu_i(\mathbf{u}(t))$ is zero. Combining these considerations and using our notation from the previous subsection, we obtain an alternative form for (2):

$$\dot{\mu}_i = -\frac{\beta_i \rho_i}{m_i}, \quad i = 1, \ldots, p \tag{9}$$

where, to simplify notation, we did not explicitly indicate the dependence of the segmentation parameters on $\mathbf{u}(t)$.

IV. Basic Properties of the 1-D SIDE

In this section, we study the system (2) and prove a number of its properties, which both allow us to gain significant insight into its behavior and are critical for developing optimal estimation algorithms presented in Section V. The most basic properties, which are illustrated in Fig. 6 and proved in [18], assert that the system has a unique solution that is continuous and becomes a constant signal in finite time.

Throughout this section, $\mathbf{u}(t)$ stands for the solution of (2) and (3), with initial condition $\mathbf{u}^0$. All the segmentation parameters encountered in this section are those of $\mathbf{u}(t)$. The final hitting time is denoted by t_f.

Proposition 1: The solution of the SIDE (2) and (3) exists, is unique, is a continuous function of the time parameter t for all values of t, and is a differentiable function of t for all t, except possibly the hitting times. Every pair of neighboring regions is merged in finite time. After the final hitting time t_f, the solution is a constant:

$$u_1(t) = u_2(t) = \cdots = u_N(t) = \frac{1}{N}\sum_{n=1}^{N} u_n^0 \text{ for } t \geq t_f.$$

Proof: See [18]. ∎

Our plan is to use the system (2) to solve a Gaussian estimation problem with a TV constraint, as well as related problems originally posed in [6], [20], and [21]. As we will see

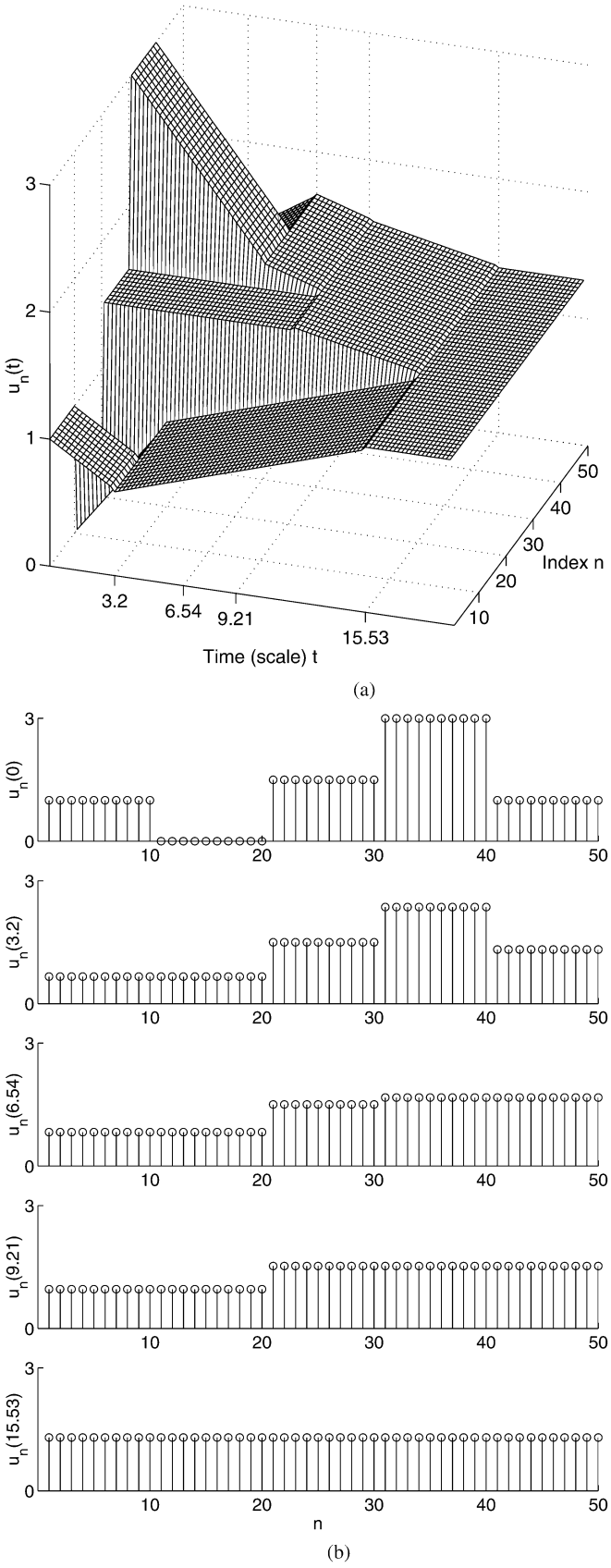


Fig. 6. Evolution of SIDE. (a) Mesh plot of the solution $\mathbf{u}(t)$ as a function of time t and index n, with hitting times $t_1 = 3.2$, $t_2 = 6.54$, $t_3 = 9.21$, and $t_4 = t_f = 15.53$. (b) Plots of the initial signal (top) and the solution at the four hitting times.

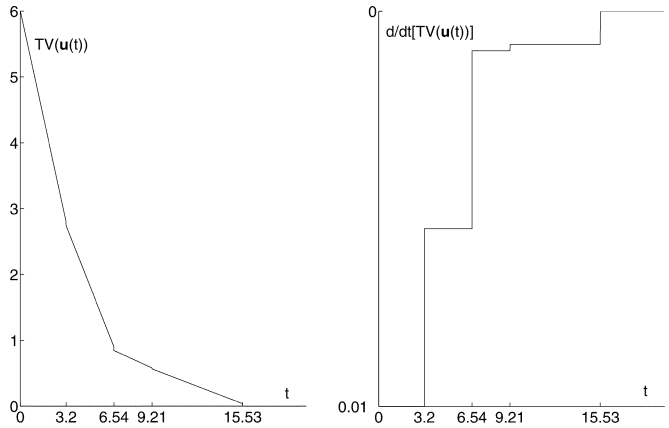


Fig. 7. (a) Total variation of the solution, as a function of time for the evolution of Fig. 6. (b) Its time derivative.

in Section V, this necessitates understanding the time behavior of $\mathrm{TV}(\mathbf{u}(t))$ and $-\|\mathbf{u}(t)-\mathbf{u}^0\|^2$. Propositions 2 and 3 below show that both these quantities are Lyapunov functionals of our system (i.e., decrease as functions of time). In the same propositions, we also derive formulas for computing the time derivatives of these quantities.

Proposition 2: For $t \in [0, t_f]$, the total variation $\mathrm{TV}(\mathbf{u}(t))$ is a monotonically decreasing function of time, which changes continuously from $\mathrm{TV}(\mathbf{u}(0)) = \mathrm{TV}(\mathbf{u}^0)$ to $\mathrm{TV}(\mathbf{u}(t_f)) = 0$ (see Fig. 7). It is a differentiable function of time, except at the hitting times, and its rate of change is

$$\dot{\mathrm{TV}}(\mathbf{u}(t)) = -\sum_{i=1}^{p} \frac{\beta_i^2 \rho_i^2}{m_i}. \tag{10}$$

Proof: Using the expression (1) for the TV of $\mathbf{u}(t)$, differentiating it with respect to t, and substituting (9) for $\dot{\mu}_i$ results in (10). This identity is valid for all $t \in [0, t_f]$, where the solution $\mathbf{u}(t)$ is differentiable and where the segmentation parameters are not changing, i.e., for all t except the hitting times. Since, by Proposition 1, the solution $\mathbf{u}(t)$ is a continuous function of t, so is $\mathrm{TV}(\mathbf{u}(t))$. We therefore conclude that $\mathrm{TV}(\mathbf{u}(t))$ is a monotonically decreasing, continuous function of time for $t \in [0, t_f]$. Its value at t_f is zero, since, by Proposition 1, $\mathbf{u}(t_f)$ is a constant signal. ∎

Proposition 3: Let $\alpha(t) = \|\mathbf{u}(t) - \mathbf{u}^0\|^2$. Then, for $t \in [0, t_f]$, $\alpha(t)$ is a monotonically increasing function of time, which changes continuously from $\alpha(0) = 0$ to $\alpha(t_f)$. It is a differentiable function of time except at the hitting times, and its rate of change is

$$\dot{\alpha}(t) = 2t \sum_{i=1}^{p} \frac{\beta_i^2 \rho_i^2}{m_i}. \tag{11}$$

Proof: The following identity is a direct corollary of Lemma 2, which is proved in the Appendix:

$$\sum_{k=n_i}^{n_{i+1}-1} u_k(t) = \sum_{k=n_i}^{n_{i+1}-1} u_k^0 + t(-\beta_i \rho_i). \tag{12}$$

We are now ready to show that (11) holds.

$$\begin{aligned}
\frac{d}{dt}\|\mathbf{u}(t)-\mathbf{u}^0\|^2 &= \frac{d}{dt}\sum_{n=1}^{N}\left(u_n(t)-u_n^0\right)^2 \\
&= 2\sum_{n=1}^{N}\left(u_n(t)-u_n^0\right)\dot{u}_n(t) \\
&\overset{\text{Eq. (2)}}{=} 2\sum_{i=1}^{p}\sum_{k=n_i}^{n_{i+1}-1}\left(u_k(t)-u_k^0\right)\left(-\frac{\beta_i\rho_i}{m_i}\right) \\
&\overset{\text{Eq. (12)}}{=} 2\sum_{i=1}^{p} t(-\beta_i\rho_i)\left(-\frac{\beta_i\rho_i}{m_i}\right) \\
&= 2t\sum_{i=1}^{p}\frac{\beta_i^2\rho_i^2}{m_i}.
\end{aligned}$$

This identity is valid for all $t \in [0, t_f]$ where the solution $\mathbf{u}(t)$ is differentiable and where the segmentation parameters are not changing, i.e., for all t except the hitting times. Since, by Proposition 1, the solution $\mathbf{u}(t)$ is a continuous function of t, so is $\alpha(t)$. We therefore conclude that $\alpha(t)$ is a monotonically increasing, continuous function of time for $t \in [0, t_f]$. ∎

Corollary 1: Let $\alpha(t) = \|\mathbf{u}(t) - \mathbf{u}^0\|^2$, and let t_l, t_{l+1} be two consecutive hitting times. Then

$$\alpha(t) = \alpha(t_l) + \left(t^2 - t_l^2\right)\sum_{i=1}^{p}\frac{\beta_i^2\rho_i^2}{m_i}, \text{for any } t \in [t_l, t_{l+1}).$$

Proof: This formula is simply the result of integrating (11) from t_l to t and using the fact that segmentation parameters remain constant between two hitting times. ∎

Proposition 4 characterizes the behavior of the functional $\|\mathbf{u}(t) - \mathbf{x}\|^2$, where $\mathbf{x}$ is an arbitrary fixed signal satisfying $\mathrm{TV}(\mathbf{x}) \leq \nu$. This result is critical in demonstrating the optimality of our algorithms of Section V.

Proposition 4: Let $\mathbf{u}(t)$ be the solution of (2) and (3), with $\mathrm{TV}(\mathbf{u}(0)) > \nu$ and $\mathrm{TV}(\mathbf{u}(t_\nu)) = \nu$, for some positive constants ν, t_ν. Suppose that $\mathbf{x} \in \mathbb{R}^N$ is an arbitrary signal with $\mathrm{TV}(\mathbf{x}) \leq \nu$. Then, for all $t \in [0, t_\nu]$ except possibly the hitting times, we have

$$\frac{1}{2}\frac{d}{dt}\|\mathbf{u}(t)-\mathbf{x}\|^2 \leq \nu - \mathrm{TV}(\mathbf{u}(t)) \tag{13}$$

$$\frac{1}{2}\frac{d}{dt}\|\mathbf{u}(t)-\mathbf{u}(t_\nu)\|^2 = \nu - \mathrm{TV}(\mathbf{u}(t)). \tag{14}$$

Proof: is in Appendix. ∎

V. Optimal Estimation in 1-D

In this section, we present 1-D estimation problems that can be efficiently solved using our evolution equation.

A. ML Estimation With a TV Constraint

Our first example is constrained maximum likelihood (ML) estimation [24] in additive white Gaussian noise. Specifically, suppose that the observation $\mathbf{u}^0$ is an N-dimensional vector of independent Gaussian random variables of variance σ^2, whose mean vector $\mathbf{x}$ is unknown. The only available information

about $\mathbf{x}$ is that its total variation is not larger than some known threshold ν. Given the data $\mathbf{u}^0$, the objective is to produce an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$.

The ML estimate maximizes the likelihood of the observation [24]

$$p(\mathbf{u}^0|\mathbf{x}) = \left(\sqrt{2\pi}\sigma\right)^{-N} e^{-(1/2\sigma^2)\|\mathbf{u}^0-\mathbf{x}\|^2}.$$

Simplifying the likelihood and taking into account the constraint $\mathrm{TV}(\mathbf{x}) \leq \nu$, we obtain the following problem:

$$\text{Find } \hat{\mathbf{x}} = \arg \min_{\mathbf{x}:\mathrm{TV}(\mathbf{x})\leq\nu} \|\mathbf{u}^0 - \mathbf{x}\|^2. \tag{15}$$

In other words, we seek the point of the constraint set $\{\mathbf{x} : \mathrm{TV}(\mathbf{x}) \leq \nu\}$, which is the closest to the data $\mathbf{u}^0$. We now show that a fast way of solving this optimization problem is to use (2) and (3).

Proposition 5: If $\mathrm{TV}(\mathbf{u}^0) \leq \nu$, then the solution to (15) is $\hat{\mathbf{x}} = \mathbf{u}^0$. Otherwise, a recipe for obtaining $\hat{\mathbf{x}}$ is to evolve the system (2) and (3) forward in time until the time instant t_ν when the solution $\mathbf{u}(t_\nu)$ of the system satisfies $\mathrm{TV}(\mathbf{u}(t_\nu)) = \nu$. Then, $\hat{\mathbf{x}} = \mathbf{u}(t_\nu)$. The ML estimate is unique and can be found in $O(N \log N)$ time and with $O(N)$ memory requirements, where N is the size of the data vector.

Proof of Proposition 5, Part 1—Optimality: The first sentence of the proposition is trivial: If the data satisfies the constraint, then the data itself is the maximum likelihood estimate. Proposition 2 of the previous section shows that if $\mathbf{u}(t)$ is the solution of the system (2) and (3), then $\mathrm{TV}(\mathbf{u}(t))$ is a monotonically decreasing function of time, which changes continuously from $\mathrm{TV}(\mathbf{u}^0)$ to 0 in finite time. Therefore, if $\mathrm{TV}(\mathbf{u}^0) > \nu$, then there exists a unique time instant t_ν when the total variation of the solution is equal to ν. We now show that $\mathbf{u}(t_\nu)$ is indeed the ML estimate sought in (15) and that this estimate is unique.

Let us denote $\phi(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|^2$. To show that $\mathbf{u}(t_\nu)$ is the unique ML estimate, we need to prove that for any $\mathbf{x} \neq \mathbf{u}(t_\nu)$ such that $\mathrm{TV}(\mathbf{x}) \leq \nu$, the following holds:

$$\phi(\mathbf{u}^0, \mathbf{u}(t_\nu)) < \phi(\mathbf{u}^0, \mathbf{x}). \tag{16}$$

To compare $\phi(\mathbf{u}^0, \mathbf{u}(t_\nu))$ with $\phi(\mathbf{u}^0, \mathbf{x})$, note that

$$\begin{aligned}\phi(\mathbf{u}^0, \mathbf{u}(t_\nu)) &= \phi(\mathbf{u}(t_\nu), \mathbf{u}(t_\nu)) \\ &\quad - \int_0^{t_\nu} \frac{d}{dt}\{\phi(\mathbf{u}(t), \mathbf{u}(t_\nu))\}\, dt \\ \phi(\mathbf{u}^0, \mathbf{x}) &= \phi(\mathbf{u}(t_\nu), \mathbf{x}) - \int_0^{t_\nu} \frac{d}{dt}\{\phi(\mathbf{u}(t), \mathbf{x})\}\, dt.\end{aligned}$$

Since $\phi(\mathbf{u}(t_\nu), \mathbf{u}(t_\nu)) = 0$ and $\phi(\mathbf{u}(t_\nu), \mathbf{x}) > 0$ for $\mathbf{x} \neq \mathbf{u}(t_\nu)$, our task would be accomplished if we could show that

$$-\int_0^{t_\nu} \frac{d}{dt}\{\phi(\mathbf{u}(t), \mathbf{u}(t_\nu))\}\, dt \leq -\int_0^{t_\nu} \frac{d}{dt}\{\phi(\mathbf{u}(t), \mathbf{x})\}\, dt. \tag{17}$$

Note that by Proposition 1, $(d/dt)\phi(\mathbf{u}(t), \mathbf{x})$ is well defined for all t, except possibly on the finite set of the hitting times, where the left derivative may not be equal to the right derivative. Both integrals in (17) are, therefore, well defined. Moreover, (17) would follow if we could prove that

$$\frac{d}{dt}\phi(\mathbf{u}(t), \mathbf{u}(t_\nu)) \geq \frac{d}{dt}\phi(\mathbf{u}(t), \mathbf{x}) \text{ for almost all } t \in [0, t_\nu] \tag{18}$$

but this is exactly what Proposition 4 of Section III states.

Finding the constrained ML estimate thus amounts to solving our system of ordinary differential (2) and (3) at a particular time instant t_ν. We now develop a fast algorithm for doing this. Roughly speaking, the low computational cost is the consequence of Propositions 2 and 3 and Corollary 1 (which provide formulas for the efficient calculation and update of the quantities of interest) and the fast sorting of the hitting times through the use of a binary heap [9].

Proof of Proposition 5, Part 2—Computational Complexity: Equations (9) and (10) show that, between the hitting times, every intensity value $\mu_i(\mathbf{u})$ changes at a constant rate, and so does $\mathrm{TV}(\mathbf{u}(t))$, as illustrated in Figs. 6 and 7. It would thus be straightforward to compute the solution once we know what the hitting times are and which regions are merged at each hitting time.

Since a hitting time is, by definition, an instant when the intensities of two neighboring regions become equal, the hitting times are determined by the absolute values of the first differences $v_i(t) = |\mu_{i+1}(\mathbf{u}(t)) - \mu_i(\mathbf{u}(t))|$ for $i = 1, \ldots p(\mathbf{u}(t)) - 1$. Let $r_i(t) = \dot{v}_i(t)$ be the rate of change of $v_i(t)$. It follows from (9) that, for a fixed i, the rate $r_i(t)$ is constant between two successive hitting times:

$$v_i(t + \Delta t) = v_i(t) + \Delta t r_i(t).$$

Suppose that after some time instant $t = \tau$, the rate $r_i(t)$ never changes: $r_i(t) = r_i(\tau)$ for $t \geq \tau$. If this were the case, we would then be able to infer from the above formula that $v_i(t)$ would become zero at the time instant $\tau + E_i(\tau)$, where $E_i(\tau)$ is defined by

$$E_i(\tau) = \begin{cases} \frac{-v_i(\tau)}{r_i(\tau)}, & \text{if } r_i(\tau) < 0 \\ \infty, & \text{otherwise} \end{cases} \tag{19}$$

but as soon as one of the v_is becomes equal to zero, the two corresponding regions are merged. The first hitting time is therefore $t_1 = \min_i E_i(0)$. Similarly, the second hitting time is $t_2 = t_1 + \min_i E_i(t_1)$, and, in general

$$\text{The } (l+1)\text{st hitting time is } t_{l+1} = t_l + \min_i E_i(t_l).$$

We now show how to compute $v_i(t), r_i(t)$, and the hitting times without explicitly computing $\mathbf{u}(t)$.

Let the signal $\mathbf{u}^*(t)$ be comprised of average values of the initial condition $\mathbf{u}^0$, taken over the regions of $\mathbf{u}(t)$:

$$\begin{aligned}u^*_{n_i}(t) &= u^*_{n_i+1}(t) = \cdots = u^*_{n_{i+1}-1}(t) \\ &= \frac{1}{m_i} \sum_{k=n_i}^{n_{i+1}-1} u^0_k, \text{ for } i = 1, 2, \ldots, p\end{aligned} \tag{20}$$

where the segmentation parameters are those of $\mathbf{u}(t)$. One of the key reasons for the low time complexity of the algorithm is that the solution $\mathbf{u}(t)$ is never explicitly computed until $t = t_\nu$. Keeping track of $\mathbf{u}^*(t)$ is enough, because of a simple relationship between the two signals. We first derive this relationship and then show that keeping track of $\mathbf{u}^*(t)$ indeed leads to a computationally efficient algorithm.

An immediate corollary of the definition (20) is that $\mathbf{u}^*(t)$ has the same number of regions as $\mathbf{u}(t)$: $p(\mathbf{u}^*(t)) = p$, $\rho_i(\mathbf{u}^*(t)) = \rho_i$. It is also immediate that edges occur in the same places in these two signals: $n_i(\mathbf{u}^*(t)) = n_i$ for $i = 1, \ldots, p$. It follows from (12) that the intensity values within the regions are related:

$$\mu_i(\mathbf{u}(t)) = \mu_i(\mathbf{u}^*(t)) - \frac{\beta_i \rho_i}{m_i} t. \tag{21}$$

Finally, note that (21) implies that the minima and maxima of $\mathbf{u}^*(t)$ occur in the same places as the minima and maxima of $\mathbf{u}(t)$. To see this, suppose that $\mu_i(\mathbf{u}(t)) < \mu_{i+1}(\mathbf{u}(t))$. Then, $\beta_i \leq 0$ and $\beta_{i+1} \geq 0$, and therefore

$$\mu_{i+1}(\mathbf{u}^*(t)) - \mu_i(\mathbf{u}^*(t)) = (\mu_{i+1}(\mathbf{u}(t)) - \mu_i(\mathbf{u}(t))) + \left(\frac{\beta_{i+1}\rho_{i+1}}{m_{i+1}} - \frac{\beta_i \rho_i}{m_i} \right) t > 0.$$

It is analogously shown that if $\mu_i(\mathbf{u}(t)) > \mu_{i+1}(\mathbf{u}(t))$, then $\mu_i(\mathbf{u}^*(t)) > \mu_{i+1}(\mathbf{u}^*(t))$. Therefore, the ith region of $\mathbf{u}(t)$ is a maximum (minimum) if and only if the ith region of $\mathbf{u}^*(t)$ is a maximum (minimum), which means that $\beta_i(\mathbf{u}^*(t)) = \beta_i$. We have thus shown that the signals $\mathbf{u}^*(t)$ and $\mathbf{u}(t)$ have identical segmentation parameters and that the intensities of these signals are related through (21).

Putting these computations together, we have the following algorithm.

1) **Initialize.** If $\mathrm{TV}(\mathbf{u}(0)) \leq \nu$, output $\mathbf{u}(0)$ and stop. Else, assign $l = 0$, $t_l = 0$, $\mathrm{TV}(\mathbf{u}(t_l)) = \mathrm{TV}(\mathbf{u}(0))$, $\mathbf{u}^*(t_l) = \mathbf{u}(0)$; initialize the segmentation parameters; use (10) to initialize $\dot{TV}(\mathbf{u}(t))$.
2) **Find possible hitting times.** For each $i = 1, \ldots, p-1$, find $v_i(t_l) = |\mu_{i+1}(\mathbf{u}(t_l)) - \mu_i(\mathbf{u}(t_l))|$ from $\mathbf{u}^*(t_l)$, using (21). Use (9) to calculate $r_i(t_l) = \dot{v}_i(t_l)$ from the segmentation parameters of $\mathbf{u}^*(t_l)$, and use (19) to find all the candidates for the next hitting time: $t_l + E_1(t_l), \ldots, t_l + E_{p-1}(t_l)$.
3) **Sort.** Store these candidates on a binary heap **[9]**.
4) **Find the next hitting time.** Find

$$j = \arg\min_i [t_l + E_i(t_l)];$$

find the next hitting time $t_{l+1} = t_l + E_j(t_l)$.
5) **Find the new TV.** Calculate the TV at $t = t_{l+1}$:

$$\mathrm{TV}(\mathbf{u}(t_{l+1})) = \mathrm{TV}(\mathbf{u}(t_l)) + (t_{l+1} - t_l)\dot{TV}(\mathbf{u}(t_l)).$$

6) **Merge.** If $\mathrm{TV}(\mathbf{u}(t_{l+1})) \geq \nu$, then merge regions j and $j+1$, as follows:
   i) remove $t_l + E_j(t_l)$ from the heap;
   ii) update $\mathbf{u}^*$:

$$\mu_j(\mathbf{u}^*(t_{l+1})) = \frac{m_j(\mathbf{u}^*(t_l))\mu_j(\mathbf{u}^*(t_l)) + m_{j+1}(\mathbf{u}^*(t_l))\mu_{j+1}(\mathbf{u}^*(t_l))}{m_j(\mathbf{u}^*(t_l)) + m_{j+1}(\mathbf{u}^*(t_l))}$$

   iii) update the time derivative of TV, using (10).
   iv) update the segmentation parameters via (5)–(8);
   v) increment l by 1;
   vi) update the heap;
   vii) go to Step 4.
7) **Output.** Calculate t_ν and $\mathbf{u}(t_\nu)$, output $\mathbf{u}(t_\nu)$, and stop.

The rest of this section explains and analyzes this algorithm.

It is easy to see that both Steps 1 and 2 have $O(N)$ time complexity. Indeed, both of them involve a constant number of operations per region of the initial signal, and since the total number of regions p cannot exceed the total number of samples N, Steps 1 and 2 can take at most $O(N)$ time. Note that Step 2 does not require the knowledge of $\mathbf{u}(t_l)$. A binary heap is built in Step 3; the complexity of this is also $O(N)$ [9].

Next follows a loop consisting of Steps 4–6. Step 4 finds the minimum of E_is, which is equivalent to extracting the root of a binary heap and is done in $O(1)$ time [9]. There are at most $N-1$ hitting times; the worst-case scenario is that the loop (Steps 4–6) terminates after the last of these hitting times, i.e., after $N-1$ iterations. In this case, the contribution of Step 4 to the total time complexity is $O(N)$.

The calculation in Step 5 is $O(1)$. Removing one number from a binary heap is $O(\log N)$ [9]. The calculation of $\mu_j(\mathbf{u}^*(t_{l+1}))$ in Step 6 is $O(1)$. Obtaining $\dot{TV}(\mathbf{u}(t_{l+1}))$ from $\dot{TV}(\mathbf{u}(t_l))$ via (10) is also $O(1)$. The reassignments (5) and (6) are $O(1)$, whereas the reassignments (7) and (8) are never explicitly performed since the underlying data structure is a dynamic linked list.

Note that a merging of two regions does not change the speed of evolution of any other regions. Therefore, updating the heap in Step 6 amounts to changing and resorting two (at most) entries that involve the newly formed region. As follows from our discussion of Step 2 above, changing the two entries takes $O(1)$ time. Resorting them on the heap is $O(\log N)$[9]. One execution of Steps 5 and 6, therefore, takes $O(\log N)$, and so, the contribution of Steps 5 and 6 to the time complexity (i.e., after $N-1$ or fewer iterations of the loop) is $O(N \log N)$.

The loop terminates when $\mathrm{TV}(\mathbf{u}(t_{l+1})) < \nu \leq \mathrm{TV}(\mathbf{u}(t_l))$. Therefore, t_ν can be found from

$$\mathrm{TV}(\mathbf{u}(t_\nu)) = \nu = \mathrm{TV}(\mathbf{u}(t_l)) + (t_\nu - t_l)\dot{TV}(\mathbf{u}(t_l))$$
$$\Rightarrow t_\nu = \frac{\nu - \mathrm{TV}(\mathbf{u}(t_l))}{\dot{TV}(\mathbf{u}(t_l))} + t_l.$$

Since $\mathbf{u}(t_\nu)$ and $\mathbf{u}(t_l)$ have the same segmentation parameters, it follows that $\mathbf{u}^*(t_\nu) = \mathbf{u}^*(t_l)$. Therefore, we can use (21) to calculate $\mathbf{u}(t_\nu)$ from $\mathbf{u}^*(t_l)$ with $O(N)$ time complexity.

The total time complexity of the algorithm is therefore $O(N \log N)$.

Storing the initial N-point signal and the accompanying parameters requires $O(N)$ memory. As the number of regions decreases, the memory is released accordingly. Therefore, the total space required is $O(N)$. ■

B. Bouman–Sauer Problem

The problem treated in the previous subsection is closely related to the tomographic reconstruction strategy considered by Bouman and Sauer in [6] and [21]. They proposed minimizing a functional that is the sum of a quadratic penalty on the difference between the data and the estimate and a regularization term that is a discretization of the total variation. Specializing their formulation to a 1-D restoration problem yields the following:

$$\text{Find } \hat{\mathbf{x}}_{BS} = \arg\min_{\mathbf{x}} \mathcal{E}_{BS}(\mathbf{x}) \tag{22}$$

$$\text{where } \mathcal{E}_{BS}(\mathbf{x}) = \|\mathbf{u}^0 - \mathbf{x}\|^2 + \lambda \text{TV}(\mathbf{x}) \tag{23}$$

where $\mathbf{u}^0$ is the data, and $\lambda > 0$ is a known parameter that controls the amount of regularization. This problem has a simple probabilistic interpretation. Just like in the previous subsection, we are estimating the mean vector $\mathbf{x}$ of a conditionally Gaussian random vector $\mathbf{u}^0$; however, now, $\mathbf{x}$ is modeled as random, with prior distribution

$$Ke^{-(\lambda/2\sigma^2)\text{TV}(\mathbf{x})}$$

where K is a normalizing constant. The optimum $\hat{\mathbf{x}}_{BS}$ is then the maximum *a posteriori* (MAP) estimate [24] of $\mathbf{x}$ based on observing $\mathbf{u}^0$.

We now show that in order to solve this problem, we can still evolve (2) and (3) using our algorithm of the previous subsection but with a different stopping rule.

Proposition 6: The solution $\hat{\mathbf{x}}_{BS}$ to Problem (22) is unique and can be calculated from the solution $\mathbf{u}(t)$ to the system (2) and (3). If $\lambda/2 < t_f$, where t_f is the final hitting time, then $\hat{\mathbf{x}}_{BS} = \mathbf{u}(\lambda/2)$; otherwise, $\hat{\mathbf{x}}_{BS} = \mathbf{u}(t_f)$. The time complexity of this calculation is $O(N \log N)$, and the space complexity is $O(N)$, where N is the size of the data.

Proof: First, note that $\mathcal{E}_{BS}(\mathbf{u}^0) = \lambda \text{TV}(\mathbf{u}^0)$, whereas if $\text{TV}(\mathbf{x}) > \text{TV}(\mathbf{u}^0)$, then $\mathcal{E}_{BS}(\mathbf{x}) > \lambda \text{TV}(\mathbf{u}^0)$. Therefore, the total variation of the solution to the Bouman–Sauer problem (22) cannot exceed the total variation of $\mathbf{u}^0$.

Let ν be a fixed real number such that $0 \leq \nu \leq \text{TV}(\mathbf{u}^0)$. As shown in Proposition 2, there is a unique time instant t_ν during the evolution of (2) and (3) when $\text{TV}(\mathbf{u}(t_\nu)) = \nu$. Moreover, it is a direct consequence of Proposition 5 that $\mathbf{u}(t_\nu)$ minimizes $\|\mathbf{u}^0 - \mathbf{x}\|^2$ over the set X_ν of all signals $\mathbf{x}$ for which $\text{TV}(\mathbf{x}) = \nu$. Since, for every signal $\mathbf{x}$ in the set X_ν, $\mathcal{E}_{BS}(\mathbf{x}) = \|\mathbf{u}^0 - \mathbf{x}\|^2 + \lambda\nu$, it follows that $\mathbf{u}(t_\nu)$ minimizes $\mathcal{E}_{BS}(\mathbf{x})$ over the set X_ν. In order to find the global minimum of $\mathcal{E}_{BS}(\mathbf{x})$, we therefore need to minimize $\mathcal{E}_{BS}(\mathbf{u}(t_\nu))$ over ν or, equivalently, find the time instant t that minimizes $\mathcal{E}_{BS}(\mathbf{u}(t))$. Combining Propositions 2 and 3, we obtain

$$\frac{d}{dt}\mathcal{E}_{BS}(\mathbf{u}(t)) = (2t - \lambda)\sum_{i=1}^{p} \frac{\beta_i^2 \rho_i^2}{m_i}.$$

For $t < \lambda/2$, this time derivative is negative; for $t > \lambda/2$, it is positive. Therefore, if $\lambda/2 < t_f$, the unique minimum is achieved when $t = \lambda/2$; if $\lambda/2 \geq t_f$, then the unique minimum is achieved when $t = t_f$. Thus, $\hat{\mathbf{x}}_{BS} = \mathbf{u}(\min(\lambda/2, t_f))$, which can be computed using our algorithm of the previous subsection whose time and space complexity are $O(N \log N)$ and $O(N)$, respectively. ■

Similar to the SIDEs, Bouman and Sauer's optimization method [6], [21], called *segmentation based optimization*, uses the idea of merging neighbors with equal intensities. In their method, a merge-and-split strategy is utilized. Neighboring pixels with equal intensities are temporarily merged. This is alternated with a split step which follows a Gauss–Seidel-type approach and optimizes over each pixel separately.

C. Rudin–Osher–Fatemi Problem

In [20], Rudin *et al.* proposed to enhance images by minimizing the total variation subject to an L^2-norm constraint on the difference between the data and the estimate. In this section, we analyze the 1-D discrete[3] version of this problem:

$$\text{Find } \hat{\mathbf{x}}_{\text{ROF}} = \arg \min_{\mathbf{x}:\|\mathbf{u}^0 - \mathbf{x}\|^2 \leq \sigma^2} \text{TV}(\mathbf{x}) \tag{24}$$

where $\mathbf{u}^0$ is the signal to be processed, and σ is a known parameter. We now show that in order to solve this problem, we can still evolve (2) and (3) using our algorithm of Section V-A but with a different stopping rule.

Proposition 7: If $\sigma^2 \geq \|\mathbf{u}(t_f) - \mathbf{u}^0\|^2$, then a solution to (24) that achieves zero total variation is $\hat{\mathbf{x}}_{\text{ROF}} = \mathbf{u}(t_f)$. Otherwise, the solution to (24) is unique and is obtained by evolving the system (2) and (3) forward in time, with the following stopping rule: Stop at the time instant when

$$\|\mathbf{u}(t) - \mathbf{u}^0\|^2 = \sigma^2. \tag{25}$$

This solution can be found in $O(N \log N)$ time and with $O(N)$ memory requirements, where N is the size of the data vector.

Proof: According to Proposition 3, $\|\mathbf{u}(t) - \mathbf{u}^0\|^2$ is a continuous, monotonically increasing function of time, and therefore, a time instant for which (25) happens is guaranteed to exist and be unique, as long as $0 \leq \sigma^2 < \|\mathbf{u}(t_f) - \mathbf{u}^0\|^2$. Let ν be the total variation of the solution at that time instant, and denote the time instant itself by t_ν. Proposition 5 of Section V-A says that $\mathbf{u}(t_\nu)$ is the unique solution of the following problem:

$$\min \|\mathbf{u}^0 - \mathbf{x}\|^2, \text{ subject to } \text{TV}(\mathbf{x}) \leq \nu.$$

In other words, if $\mathbf{x} \neq \mathbf{u}(t_\nu)$ is any signal with $\text{TV}(\mathbf{x}) \leq \nu$, then $\|\mathbf{u}^0 - \mathbf{x}\|^2 > \|\mathbf{u}^0 - \mathbf{u}(t_\nu)\|^2 = \sigma^2$. This means that if $\mathbf{x} \neq \mathbf{u}(t_\nu)$ is any signal with $\|\mathbf{u}^0 - \mathbf{x}\|^2 \leq \sigma^2$, then we must have $\text{TV}(\mathbf{x}) > \nu$. Therefore, $\mathbf{u}(t_\nu)$ is the unique solution of (24).

We note moreover that the new stopping rule does not change the overall computational complexity of our algorithm of Section V-A. Every time two regions are merged, two terms in the sum $\sum_i \beta_i \rho_i / m_i$ disappear, and one new term appears. All other terms stay the same, and therefore, updating the sum after a merge takes $O(1)$ time. Once this sum is updated, computing the new $\|\mathbf{u}(t) - \mathbf{u}^0\|^2$ is also $O(1)$ thanks to the Corollary of Proposition 3. ■

[3]This means that our signals are objects in $\mathbb{R}^N$, rather than L^2.

D. Adaptive Stopping

We now outline a strategy for a simple heuristic modification of our algorithm for the case when the parameters ν, λ, and σ of Sections V-A–C, respectively, are unavailable and cannot be easily estimated.

In the absence of these parameter values, a different stopping rule for our evolution equation can be designed, based on the following qualitative considerations. The underlying algorithm is a region-merging procedure that starts with a fine segmentation and then recursively removes regions. The regions that are removed at the beginning of the process typically correspond to noise. However, as shown above, the process will eventually aggregate all signal samples into one region, and therefore, it will at some point start removing "useful" regions (i.e., the regions that are due to the underlying signal that is to be recovered). A good stopping rule would stop this evolution when most of the "noise" regions have been removed but most of the "useful" regions are still intact. A good heuristic for distinguishing the two types of region is the energy: We would expect the "noise" regions to have a small energy and the "useful" regions to have a high energy. Specifically, we track the quantity $\|\mathbf{u}^*(t_l) - \mathbf{u}^*(t_{l-1})\|^2$. At each hitting time, this quantity is evaluated and compared to its average over the previous hitting times. When this quantity abruptly changes, the evolution is stopped.

VI. Conclusions and Future Research Directions

In this paper, we presented a relationship between nonlinear diffusion filtering and optimal estimation. We showed that this relationship is precise in 1-D and results in fast restoration algorithms both in 1-D and 2-D. In particular, we developed $O(N \log N)$ algorithms to exactly solve the following three problems in 1-D:

- the problem of finding the TV-constrained maximum likelihood estimate of an unknown N-point signal corrupted by additive white Gaussian noise;
- the Bouman–Sauer problem [6], [21] of minimizing a TV-regularized functional;
- the Rudin–Osher–Fatemi problem [20] of constrained TV minimization.

In each case, our algorithm requires one parameter—the stopping criterion—which is related to the noise variance. When this parameter is unavailable, we can use an adaptive version of our procedure.

The generalization of our algorithm to 2-D is achieved by defining a region as any connected set of pixels and defining two regions to be neighbors if they share a boundary, as described in [18]. The 2-D algorithm was illustrated above and shown to produce comparable results to the existing methods, with considerable computational savings.

We believe that there are several interesting theoretical and practical issues for future research. Since our basic algorithm is a region-merging algorithm, the estimates it produces are piecewise-constant. An issue therefore arises of modifying the algorithm in such a way that it produces better estimates in smooth regions. One possibility is to seek approximations by higher order splines, rather than by staircase functions.

A more thorough investigation of the adaptive stopping rules is another open research issue. This includes finding better statistics for differentiating between the "noise" regions and the "useful" regions, as well as utilizing better change detection algorithms.

Finally, we would like to obtain results similar to Proposition 5 for Laplacian noise, i.e., for the case when the quantity to be minimized in (15) is the ℓ^1 norm rather than the square of the ℓ^2 norm.

Appendix

The proof of the central result of this paper (Proposition 5 of Section V) uses (18), which follows from Proposition 4 of Section III. We prove this proposition below. The proof is organized in several modules: the main body of the proof and auxiliary lemmas, which prove one equality and two inequalities used in the main body.

A. Proof of Four Auxiliary Lemmas

In the first lemma, we consider the solution $\mathbf{u}(t)$ of (2) and (3) at some fixed time instant $t = \tau$. For the case when $\mathbf{u}(\tau)$ has an edge at $(n, n+1)$ (i.e., $u_n(\tau) \neq u_{n+1}(\tau)$), we calculate the rate of change of the cumulative sum $\sum_{k=1}^{n} u_k(t)$ during the time interval $0 \leq t \leq \tau$.

Lemma 1: Let $\mathbf{u}(t)$ be the solution of (2) and (3), with final hitting time t_f, and let τ be a fixed time instant with $0 \leq \tau < t_f$. Suppose that index n is such that

$$u_n(\tau) < u_{n+1}(\tau). \tag{26}$$

Then

$$\frac{d}{dt} \sum_{k=1}^{n} u_k(t) = 1 \text{ for any } t \in [0, \tau]. \tag{27}$$

Similarly, if $u_n(\tau) > u_{n+1}(\tau)$, then

$$\frac{d}{dt} \sum_{k=1}^{n} u_k(t) = -1 \text{ for any } t \in [0, \tau]. \tag{28}$$

Proof: Suppose (26) holds. Then, necessarily

$$u_n(t) < u_{n+1}(t), \text{ for any } t \in [0, \tau]. \tag{29}$$

Indeed, suppose this was not true, i.e., there was a time instant $t \in [0, \tau]$ for which $u_n(t) \geq u_{n+1}(t)$. Then, owing to the continuity of the solution, there would be a time instant $T \in [t, \tau]$ such that $u_j(T) = u_{j+1}(T)$. At that time instant, the samples n and $n+1$ would be merged, and their intensities would stay equal for all future time, resulting in $u_n(\tau) = u_{n+1}(\tau)$ and contradicting (26).

It follows from (29) that samples n and $n+1$ belong to different regions of $\mathbf{u}(t)$, for any $t \in [0, \tau]$, which means that $n+1$ is always the left endpoint of a region. Suppose that at some time instant t, the point $n+1$ is the left endpoint of region j: $n+1 = n_j(\mathbf{u}(t))$.

It follows from (2) that the time derivatives of all samples of the ith region of $\mathbf{u}(t)$ are the same and equal to $\dot{\mu}_i(\mathbf{u}(t))$, and therefore

$$\begin{aligned}\frac{d}{dt}\sum_{k=1}^{n} u_k(t) &= \sum_{k=1}^{n} \dot{u}_k(t) = \sum_{i=1}^{j-1} m_i(\mathbf{u}(t))\dot{\mu}_i(\mathbf{u}(t)) \\ &\overset{\text{Eq. (2)}}{=} \sum_{i=1}^{j-1}\Big\{\operatorname{sgn}[\mu_{i+1}(\mathbf{u}(t)) - \mu_i(\mathbf{u}(t))] \\ &\qquad - \operatorname{sgn}[\mu_i(\mathbf{u}(t)) - \mu_{i-1}(\mathbf{u}(t))]\Big\} \\ &= \operatorname{sgn}[\mu_j(\mathbf{u}(t)) - \mu_{j-1}(\mathbf{u}(t))] \\ &= \operatorname{sgn}[u_{n+1}(t) - u_n(t)] = 1.\end{aligned}$$

It is similarly shown that (28) holds if $u_n(\tau) > u_{n+1}(\tau)$. ∎

In the next lemma, we consider a region of the solution $\mathbf{u}(t)$ to (2) and (3) at a particular time instant τ and use Lemma 1 to calculate the rate of change for the sum of intensities within the region for $t \in [0, \tau]$.

Lemma 2: Let $\mathbf{u}(t)$ be the solution of (2) and (3), with final hitting time t_f, and let τ be a fixed time instant with $0 \le \tau < t_f$. Then

$$\frac{d}{dt}\sum_{k=n_i(\mathbf{u}(\tau))}^{n_{i+1}(\mathbf{u}(\tau))-1} u_k(t) = -\beta_i(\mathbf{u}(\tau))\rho_i(\mathbf{u}(\tau)) \quad \text{for any } t \in [0,\tau]. \tag{30}$$

In other words, for any region of $\mathbf{u}(\tau)$, the sum of values within the region evolves with a constant velocity (given by the right-hand side of (30)) from $t = 0$ until $t = \tau$.

Proof: Representing the left-hand side of (30) as

$$\frac{d}{dt}\sum_{k=1}^{k_1} u_k(t) - \frac{d}{dt}\sum_{k=1}^{k_2} u_k(t)$$

with $k_1 = n_{i+1}(\mathbf{u}(\tau)) - 1$ and $k_2 = n_i(\mathbf{u}(\tau)) - 1$, we get (30) as a direct corollary of Lemma 1. ∎

The next lemma essentially says that a local averaging operation cannot result in an increase of the total variation. This is natural to expect, since the total variation is a measure of "roughness."

Lemma 3: Let $\mathbf{x}, \mathbf{u} \in \mathbb{R}^N$ be arbitrary signals, and let $p = p(\mathbf{u})$; $n_i = n_i(\mathbf{u})$ for $i = 1, \ldots, p$; and $n_{p+1} = N + 1$. Let $\mathbf{x}^* \in \mathbb{R}^N$ be the result of averaging $\mathbf{x}$ over the regions of $\mathbf{u}$:

$$\begin{aligned}x^*_{n_i} &= x^*_{n_i+1} = \cdots = x^*_{n_{i+1}-1} \\ &= \frac{1}{n_{i+1} - n_i}\sum_{k=n_i}^{n_{i+1}-1} x_k, \text{for } i = 1, \ldots, p.\end{aligned}$$

Then, $\mathrm{TV}(\mathbf{x}^*) \le \mathrm{TV}(\mathbf{x})$.

Proof: For $i = 1, \ldots, p$, we define

$$\begin{aligned}x_{i,\max} &= \max_{n_i \le k \le n_{i+1}-1} x_k \\ x_{i,\min} &= \min_{n_i \le k \le n_{i+1}-1} x_k \\ s_i &= \begin{cases} x_{i+1,\max} - x_{i,\min}, & \text{if } x^*_{n_{i+1}} \ge x^*_{n_i} \\ x_{i,\max} - x_{i+1,\min}, & \text{if } x^*_{n_{i+1}} < x^*_{n_i}. \end{cases}\end{aligned}$$

Note that

$$\sum_{i=1}^{p-1} s_i \le \sum_{n=1}^{N-1} |x_{n+1} - x_n| = \mathrm{TV}(\mathbf{x}). \tag{31}$$

Since $x^*_{n_i}$, $x_{i,\min}$, and $x_{i,\max}$ are the mean, min, and max, respectively, of the numbers $\{x_k\}_{k=n_i}^{n_{i+1}-1}$, we have $x_{i,\min} \le x^*_{n_i} \le x_{i,\max}$. Therefore, it follows from the first line of the definition of s_i that if $x^*_{n_{i+1}} \ge x^*_{n_i}$, then

$$\left|x^*_{n_{i+1}} - x^*_{n_i}\right| = x^*_{n_{i+1}} - x^*_{n_i} \le x_{i+1,\max} - x_{i,\min} = s_i.$$

Similarly, from the second line of the definition of s_i, we have that if $x^*_{n_{i+1}} < x^*_{n_i}$, then

$$\left|x^*_{n_{i+1}} - x^*_{n_i}\right| = x^*_{n_i} - x^*_{n_{i+1}} \le x_{i,\max} - x_{i+1,\min} = s_i.$$

In both cases, $\left|x^*_{n_{i+1}} - x^*_{n_i}\right| \le s_i$. Summing both sides of this inequality from $i = 1$ to $i = p - 1$ and using (31), we get

$$\mathrm{TV}(\mathbf{x}^*) \le \sum_{i=1}^{p-1} s_i \le \mathrm{TV}(\mathbf{x}).$$

∎

The next lemma says that if, in (1), one uses incorrect locations of the extrema, the result will be less than or equal to the actual total variation.

Lemma 4: Let $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{R}^N$ be two signals whose segmentation parameters are identical, except for β_is. In other words, the extrema of the two signals do not necessarily occur at the same locations, but $p(\mathbf{x}^1) = p(\mathbf{x}^2)$, and $n_i(\mathbf{x}^1) = n_i(\mathbf{x}^2)$ for $i = 1, \ldots, p$. Then

$$\mathrm{TV}(\mathbf{x}^1) \ge \sum_{i=1}^{p} \beta_i(\mathbf{x}^2)\rho_i\mu_i(\mathbf{x}^1). \tag{32}$$

Proof: Let $i_1 < i_2 < \cdots < i_q$ be the regions that are the extrema of $\mathbf{x}^2$. Without loss of generality, suppose that the leftmost extremum i_1 is a minimum. The right-hand side of (32) can then be rewritten as follows:

$$\begin{aligned}\sum_{i=1}^{p} \beta_i(\mathbf{x}^2)\rho_i\mu_i(\mathbf{x}^1) &= -\mu_{i_1}(\mathbf{x}^1) + 2\sum_{r=2}^{q-1}(-1)^r\mu_{i_r}(\mathbf{x}^1) \\ &\quad + (-1)^q\mu_{i_q}(\mathbf{x}^1) \\ &= (\mu_{i_2}(\mathbf{x}^1) - \mu_{i_1}(\mathbf{x}^1)) \\ &\quad + (\mu_{i_2}(\mathbf{x}^1) - \mu_{i_3}(\mathbf{x}^1)) \\ &\quad + (\mu_{i_4}(\mathbf{x}^1) - \mu_{i_3}(\mathbf{x}^1)) + \cdots \\ &= \sum_{r=1}^{q-1} |\mu_{i_{r+1}}(\mathbf{x}^1) - \mu_{i_r}(\mathbf{x}^1)| \\ &\le \sum_{i=1}^{p(\mathbf{x}^1)-1} |\mu_{i+1}(\mathbf{x}^1) - \mu_i(\mathbf{x}^1)| \\ &= \mathrm{TV}(\mathbf{x}^1).\end{aligned}$$

∎

B. Proof of Proposition 4

Note that, by Proposition 1, we can differentiate at all time points except possibly the hitting times.

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\|\mathbf{u}(t)-\mathbf{x}\|^2 &= \frac{1}{2}\frac{d}{dt}\sum_{n=1}^{N}(u_n(t)-x_n)^2\\ &= \sum_{n=1}^{N}(u_n(t)-x_n)\dot{u}_n(t)\\ &= \sum_{n=1}^{N}u_n(t)\dot{u}_n(t)-\sum_{n=1}^{N}x_n\dot{u}_n(t). \quad (33)\end{aligned}$$

Let us now calculate the two terms in (33) separately. In this first calculation, all segmentation parameters are those of $\mathbf{u}(t)$.

$$\begin{aligned}\sum_{n=1}^{N}u_n(t)\dot{u}_n(t) &= \sum_{i=1}^{p}\sum_{k=n_i}^{n_{i+1}-1}u_k(t)\dot{u}_k(t)\\ &\overset{\text{Eq. (2)}}{=} \sum_{i=1}^{p}m_i\mu_i\dot{\mu}_i \overset{\text{Eq. (9)}}{=} -\sum_{i=1}^{p}\beta_i\rho_i\mu_i\\ &\overset{\text{Eq. (1)}}{=} -\mathrm{TV}(\mathbf{u}(t)). \quad (34)\end{aligned}$$

When $\mathbf{x}=\mathbf{u}(t_\nu)$ with $t_\nu \geq t$, the second term of (33) is evaluated as follows (where now the segmentation parameters are those of $\mathbf{u}(t_\nu)$):

$$\begin{aligned}-\sum_{n=1}^{N}x_n\dot{u}_n(t) &= -\sum_{n=1}^{N}u_n(t_\nu)\dot{u}_n(t)\\ &= -\sum_{i=1}^{p}\sum_{k=n_i}^{n_{i+1}-1}u_k(t_\nu)\dot{u}_k(t)\\ &= -\sum_{i=1}^{p}\mu_i\sum_{k=n_i}^{n_{i+1}-1}\dot{u}_k(t)\\ &\overset{\text{Lemma 2}}{=} -\sum_{i=1}^{p}\mu_i(-\beta_i\rho_i) \quad (35)\\ &\overset{\text{Eq. (1)}}{=} \mathrm{TV}(\mathbf{u}(t_\nu))\\ &= \nu. \quad (36)\end{aligned}$$

Using segmentation parameters of $\mathbf{u}(t)$ in this third calculation, we have, for a general $\mathbf{x}$ with $\mathrm{TV}(\mathbf{x})\leq\nu$:

$$\begin{aligned}-\sum_{n=1}^{N}x_n\dot{u}_n(t) &= -\sum_{i=1}^{p}\sum_{k=n_i}^{n_{i+1}-1}x_k\dot{u}_k(t)\\ &\overset{\text{Eq. (9)}}{=} -\sum_{i=1}^{p}\left(\sum_{k=n_i}^{n_{i+1}-1}x_k\right)\left(-\frac{\beta_i\rho_i}{m_i}\right)\\ &= \sum_{i=1}^{p}(m_ix^*_{n_i})\left(\frac{\beta_i\rho_i}{m_i}\right)\\ &= \sum_{i=1}^{p}\beta_i\rho_ix^*_{n_i} \overset{\text{Lemma 4}}{\leq} \mathrm{TV}(\mathbf{x}^*)\\ &\overset{\text{Lemma 3}}{\leq} \mathrm{TV}(\mathbf{x})\\ &\leq \nu \quad (37)\end{aligned}$$

where $\mathbf{x}^*$ is as defined in Lemma 3, i.e., obtained by averaging $\mathbf{x}$ over the regions of $\mathbf{u}(t)$. Substituting (34), (36), and (37) into (33), we obtain (13) and (14), which we needed to verify. ■

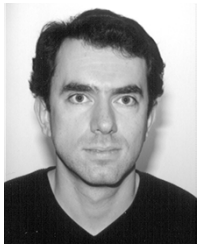
ACKNOWLEDGMENT

The authors thank Profs. D. Mumford and H. Krim for many helpful discussions. After this paper was accepted for publication, it came to our attention that another research group has independently developed a different proof of the optimality part of Proposition 6 [22].

REFERENCES

[1] R. Acar and C. R. Vogel, "Analysis of bounded variation penalty methods for ill-posed problems," *Inv. Prob.*, vol. 10, 1994.
[2] S. Alliney, "A property of the minimum vectors of a regularizing functional defined by means of the absolute norm," *IEEE Trans. Signal Process.*, vol. 45, no. 4, Apr. 1997.
[3] S. Alliney and S. A. Ruzinsky, "An algorithm for the minimization of mixed ℓ_1 and ℓ_2 norms with applications to Bayesian estimation," *IEEE Trans. Signal Process.*, vol. 42, no. 3, Mar. 1994.
[4] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, May 1998.
[5] P. Blomgren and T. F. Chan, "Modular solvers for constrained image restoration problems," *Numer. Linear Algebra Applicat.*, vol. 9, no. 5, pp. 347–358, 2002.
[6] C. Bouman and K. Sauer, "An edge-preserving method for image reconstruction from integral projections," in *Proc. Conf. Info. Sci. Syst.*, Baltimore, MD, Mar. 1991, pp. 383–387.
[7] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numer. Math.*, vol. 76, 1997.
[8] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for TV-based image restoration," in *Proc. ICAOS: Images, Wavelets, PDEs*, Paris, France, June 1996, pp. 241–252.
[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*: MIT Press, 1990.
[10] S. K. Djumagazieva, "Numerical integration of a certain partial differential equation," *U.S.S.R. Comput. Maths. Math. Phys.*, vol. 23, no. 4, pp. 45–49, 1983.
[11] D. Dobson and F. Santosa, "An image enhancement technique for electrical impedance tomography," *Inv. Prob.*, vol. 10, 1994.
[12] M. G. Fleming, C. Steger, J. Zhang, J. Gao, A. B. Cognetta, I. Pollak, and C. R. Dyer, "Techniques for a structural analysis of dermatoscopic imagery," *Computerized Med. Imaging Graphics*, vol. 22, 1998.
[13] H. Krim and Y. Bao, "A stochastic diffusion approach to signal denoising," in *Proc. ICASSP*, Phoenix, AZ, 1999.
[14] A. Marquina and S. Osher, "Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal," *SIAM J. Sci. Comput.*, vol. 22, no. 2, pp. 387–405, 2000.
[15] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.
[16] I. Pollak, "Nonlinear Scale-Space Analysis in Image Processing," Ph.D. dissertation, Lab. Inform. Decision Syst., Mass. Inst. Technol., Cambridge, MA, 1999.
[17] ——, "Nonlinear multiscale filtering," *IEEE Signal Process. Mag.*, pp. 26–36, Sep. 2002.
[18] I. Pollak, A. S. Willsky, and H. Krim, "Image segmentation and edge enhancement with stabilized inverse diffusion equations," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 256–266, Feb. 2000.
[19] ——, "A nonlinear diffusion equation as a fast and optimal solver of edge detection problems," in *Proc. ICASSP*, Phoenix, AZ, 1999.
[20] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, 1992.
[21] K. Sauer and C. Bouman, "Bayesian estimation of transmission tomograms using segmentation based optimization," *IEEE Trans. Nuclear Sci.*, vol. 39, no. 4, pp. 1144–1152, Aug. 1992.
[22] G Steidl, J. Weickert, T. Brox, P. Mrazek, and M. Welk, "On the equalvalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDES," *SIAM J. Numer. Anal.*, vol. 42, no. 2, pp. 686–713, 2004.

[23] P. C. Teo, G. Sapiro, and B. Wandell, "Anisotropic smoothing of posterior probabilities," in *Proc. ICIP*, Santa Barbara, CA, 1997.
[24] H. van Trees, *Detection, Estimation, and, Modulation Theory*. New York: Wiley, 1968, vol. 1.
[25] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Trans. Image Process.*, vol. 7, no. 6, pp. 813–824, Jun. 1998.
[26] J. Weickert, J. Heers, C. Schnörr, K. J. Zuiderveld, O. Scherzer, and H. S. Stiehl, "Fast parallel algorithms for a broad class of nonlinear variational diffusion approaches," *Real-Time Imaging*, vol. 7, pp. 31–45, 2001.
[27] J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 398–410, Mar. 1998.
[28] S. C. Zhu and D. Mumford, "Prior learning and Gibbs reaction-diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 11, pp. 1236–1250, Nov. 1997.
[29] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 9, pp. 884–900, Sep. 1996.



Ilya Pollak received the B.S. and M.Eng. degrees in 1995 and the Ph.D. degree in 1999, all in electrical engineering from the Massachusetts Institute of Technology, Cambridge.

From 1999 to 2000, he was a post-doctoral researcher with the Division of Applied Mathematics, Brown University, Providence, RI. Since 2000, he has been an Assistant Professor of electrical and computer engineering at Purdue University, West Lafayette, IN. In the Summer of 1996, he held a visiting position at the Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France. In the Summer of 2001, he held a visiting position at Tampere University of Technology, Tampere, Finland. His research interests are in image and signal processing, specifically, nonlinear scale-spaces, adaptive representations, hierarchical statistical models, and estimation algorithms.

Dr. Pollak received a CAREER award from the National Science Foundation in 2001.

Alan S. Willsky (S'70–M'73–SM'82–F'86) joined the faculty of the Massachusetts Institute of Technology (MIT), Cambridge, in 1973 and is currently the Edwin Sibley Webster Professor of Electrical Engineering. He is a founder, member of the Board of Directors, and Chief Scientific Consultant of Alphatech, Inc. From 1998 to 2002, he served as a member of the U.S. Air Force Scientific Advisory Board. He has held visiting positions in England and France. He has delivered numerous keynote addresses and is co-author of the undergraduate text *Signals and Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1996, Second ed.). His research interests are in the development and application of advanced methods of estimation and statistical signal and image processing. Methods he has developed have been successfully applied in a variety of applications including failure detection, surveillance systems, biomedical signal and image processing, and remote sensing.

Dr. Willsky has received several awards, including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, and the 1980 IEEE Browder J. Thompson Memorial Award. He has held various leadership positions in the IEEE Control Systems Society (which made him a Distinguished Member in 1988).

Yan Huang received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 2000 and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2004.

She is currently a post-doctoral researcher with the School of Electrical and Computer Engineering, Purdue University. Her research interests are in image and signal processing.