NONLOCAL LINEAR IMAGE REGULARIZATION AND SUPERVISED SEGMENTATION

GUY GILBOA AND STANLEY OSHER

Abstract. A nonlocal quadratic functional of weighted differences is examined. The weights are based on image features and represent the affinity between different pixels in the image. By prescribing different formulas for the weights, one can generalize many local and nonlocal linear denoising algorithms, including the nonlocal means filter and the bilateral filter. In this framework one can easily show that continuous iterations of the generalized filter obey certain global characteristics and converge to a constant solution. The linear operator associated with the Euler-Lagrange equation of the functional is closely related to the graph Laplacian. We can thus interpret the steepest descent for minimizing the functional as a nonlocal diffusion process. This formulation allows a convenient framework for nonlocal variational minimizations, including variational denoising, Bregman iterations and the recently proposed inverse-scale-space.

It is also demonstrated how the steepest descent flow can be used for segmentation. Following kernel based methods in machine learning, the generalized diffusion process is used to propagate sporadic initial user's information to the entire image. Unlike classical variational segmentation methods the process is not explicitly based on a curve length energy and thus can cope well with highly non-convex shapes and corners. Reasonable robustness to noise is still achieved.

Key words. Denoising, regularization, image segmentation, interactive segmentation, nonlocal evolutions, diffusion, scale-space, spectral graph theory, kernel methods.

AMS subject classifications. 47A52, 68U10, 49N10, 68R10

1. Introduction. Evolutions based on partial differential equations (PDE's) have shown to provide very effective tools in image processing and computer vision. For some recent theory and applications see [3, 47, 46, 19] and the references therein. Here we will try to give a unified approach to both denoising and segmentation tasks using nonlocal functionals and their respective nonlocal evolutions. In this paper we focus on the simpler case of quadratic functionals and linear evolutions.

This study relates to many image processing disciplines and mathematical methods, some of which are not necessarily related to PDE's: spectral graph theory [20, 42], segmentation by seeded region growing [1, 65], graph-based segmentation [52, 48, 60], the Beltrami flow on Riemannian manifolds [34, 54, 33], relations between the graph Laplacian and the Laplace-Beltrami and other operators [6, 44] and more.

More specifically, the study was inspired by some recent studies on diffusion geometries [21, 44, 56], denoising by non-local means [10] and interactive segmentation [8, 50, 38].

We summarize below only the most relevant results which will be used later in the paper.

1.1. Spectral Graph Theory. Our framework can be viewed as a continuous generalization of graphs and relates to concepts from spectral graph theory [20, 42]. A fundamental operator used in this field is the graph Laplacian.

Let G = (V, E) be a connected undirected weighted graph with (a finite set of) vertices (nodes) V and edges E. To each edge $e_{kl} \in E$ between nodes k and l a corresponding weight $w_{kl} \in E$ is defined. The weights are non-negative and symmetric: $w_{kl} \geq 0$, $w_{kl} = w_{lk}$. We assume that a discrete function u is defined on the nodes of the graph and denote by $u(k) \in V$ the value of u at node k. The (weighted) graph Laplacian is

$$\Delta_G(u(k)) := \sum_{l \in \mathcal{N}_k} w_{kl}(u(l) - u(k)), \quad k, l \in V,$$
(1.1)

where $l \in \mathcal{N}_k$ is the set of nodes with edges connected to k. Note that we define here the Laplacian with an *opposite sign* to the usual graph theoretic definition so it will coincide with the continuous definition.

The weight can be viewed as the inverse square distance between nodes. For example, in a standard two dimensional grid with grid size Δx , by assigning $w_{kl} = \frac{1}{(\Delta x)^2}$ to the four nearest neighbors of each node and zero otherwise we get that (1.1) reduces to the familiar discretized approximation of the Laplacian: $\Delta(u(k)) \approx \frac{1}{(\Delta x)^2} (\sum_{l \in \mathcal{N}_k} u(l) - 4u(k)).$

1.2. Nonlocal Means. In [10] Buades-Coll-Morel suggested the following non-local filter for image denoising:

$$NL(u)(x) = \frac{1}{c(x)} \int_{\Omega} e^{-d_a(u(x), u(y))/h^2} u(y) dy$$
(1.2)

where

$$d_a(u(x), u(y)) = \int_{\Omega} G_a(t) |u(x+t) - u(y+t)|^2 dt$$
(1.3)

 G_a is a Gaussian with standard deviation a, and c(x) is a normalization factor:

$$c(x) = \int_{\Omega} e^{-d_a(u(x), u(y))/h^2} dy.$$
 (1.4)

The corresponding discrete formulation is:

$$NL(u)(i) = \sum_j \alpha(i,j) u(j)$$

where

$$\alpha(i,j) = \frac{1}{c(i)} e^{-\|u(B_i) - v(B_j)\|_{2,a}^2 / h^2}$$

 $u(B_i) = (u(k) : k \in B_i), B_i$ is a small ball (patch, in general) around pixel *i*.

This method emerged as a generalization of the Yaroslavsky filter [62] and patch based methods, proposed for texture synthesis by [22]. In [11] the asymptotic relation of neighborhood filters to Perona-Malik type PDE's [49] is shown and a linear correction to avoid staircasing is suggested. In [39] a fast algorithm was designed for computing the fully nonlocal version. The study of [32] presented a statistical analysis of the problem and suggested to use an adaptive window approach which minimizes a local risk measure. The studies [39] and [32] both conclude that a more careful selection of the pixels which are to be considered in the averaging can improve the performance. This will be our conclusion also. Our iterative process will in effect produce an adaptive window implicitly, although each iteration uses a simple fixed window. Remark on the normalization: The normalization using (1.4) does not guarantee that the mean value of the filtered image u is the same as the mean value of the input image. For white Gaussian noise (with mean zero) this property is not desired. Also, normalizing in this manner introduces some bias from the original distance between points with many similar regions (c(x) high) to more rare and singular points (c(x) low). Dividing by c(x) tends to diminish this distinction. Moreover, the normalization breaks down the symmetry of the similarity between points in the image. Although $d_a(u(x), u(y)) = d_a(u(y), u(x))$ the final similarity measure (after normalization) between points x and y is not the same as between points y and x. Thus, simply iterating Eq. (1.2) is not strictly a diffusion process. In fact it can be viewed as taking steps in Jacobi's method (see more details in Appendix D). We will show a different normalization, very standard in iterative parabolic processes, which retains symmetric similarities between points, ensures the preservation of the mean value and does not tend to blur singular regions. We believe this may explain, in part, why our proposed iterative process outperforms the original filter.

1.2.1. A variational viewpoint. In [35] Kindermann-Osher-Jones interpreted the NL-means and neighborhood filters in general as regularizations based on nonlocal functionals in the general form:

$$J_{KOJ}(u) := \int_{\Omega \times \Omega} g\left(\frac{|u(x) - u(y)|^2}{h^2}\right) w(|x - y|) dx dy, \tag{1.5}$$

where the Yaroslavsky functional is

$$J_{Yar}(u) := \int_{\Omega \times \Omega} \left(1 - exp(\frac{-|u(x) - u(y)|^2}{h^2}) \right) w(|x - y|) dx dy,$$

and the NL-means functional is

$$J_{BCM}(u) := \int_{\Omega \times \Omega} \left(1 - exp(\frac{-d_a(u(x), u(y))}{h^2}) \right) w(|x - y|) dx dy.$$

Filtering is obtained by solving a minimization problem using the above functionals. In the above cases w(|x - y|) is a simple symmetric window and $g(\cdot)$ determines the characteristics of the regularizer. The main problem is that in general these type of functionals are not convex.

We follow this approach, simplifying the functional to a quadratic one by changing the roles of g and w.

1.3. Graph-based segmentation algorithms.

1.3.1. Supervised and interactive segmentation. Boykov et al. [8] proposed an interactive segmentation algorithm in which the user gives initial indications on the object to be segmented and the background with additional feedbacks to correct possible errors in the segmentation. The method is based on representing the image as a weighted graph and using a graph cut algorithm [9, 36] to solve the segmentation problem. Improvements and extensions to the method were proposed in [38, 50, 59]. We will show how our proposed nonlocal evolution can be used to perform segmentation with initial user inputs (supervised). The algorithm can be easily extended to an interactive algorithm (where feedbacks on the results are given).

1.3.2. Algorithms using the graph Laplacian. In [52] Shi and Malik suggested to threshold the second smallest eigenvector of the graph Laplacian to approximate the normalized cuts criterion. Weiss [60] suggested to use a combination of the smallest eigenvectors and has shown the connection to the algorithm of [48]. Partially labelled data was incorporated to the algorithm as constraints in [64]. Grady et al [29, 28] used foreground-background marks, similar to [8], and solved the Laplace equation with the marks as constraints. A more general machine learning approach with kernel methods is presented in [37] and [6]. These kernels are adapted to the given data structure and are able to generalize well information of a training set to a much larger data set, or as in our case, infer from partially labelled data (regarding points belonging to the object of background) to the entire image. We will use these techniques to obtain a nonlocal segmentation algorithm.

1.3.3. Main contributions of the paper. A general quadratic variational framework is presented for image and signal regularization. It consists of a preprocessing stage, where affinities between different regions in the image are established, and a regularizing stage (using a descent flow or by solving a minimization problem). In particular we show how the nonlocal means filter can be generalized in this way, introducing a consistent simplifying nonlocal procedure which produces superior results than the original method. We also show how the same evolution can be used for both tasks of denoising and segmentation, by simply changing the initial conditions.

2. The Regularizing Functional. In this paper the following nonlocal functional is examined:

$$J(u) := \frac{1}{4} \int_{\Omega \times \Omega} (u(x) - u(y))^2 w(x, y) dx dy,$$
 (2.1)

where $\Omega \in \mathbb{R}^n$, $x = (x_1, \ldots, x_n) \in \Omega$ and $y = (y_1, \ldots, y_n) \in \Omega$. For images we have n = 2. The weight function $w(x, y) \in \Omega \times \Omega$ is positive: $w(x, y) \ge 0$ and symmetric: w(x, y) = w(y, x). For image processing tasks the weight function is based on image features and can be understood as the proximity between two points x and y, based on features in their neighborhood. The way to obtain such functional (1.5) is the role of w(x, y) which is much more significant in our case. It basically determines the type of regularization. We will show that this linear and simple framework still allows relatively complicated image-processing tasks, due to its non-local nature.

The corresponding Euler-Lagrange descent flow is:

$$u_t(x) = -J'(u)(x) = -\int_{\Omega} (u(x) - u(y))w(x,y)dy.$$
 (2.2)

Let us define the following linear operator:

$$Lu(x) = \int_{\Omega} (u(y) - u(x))w(x, y)dy.$$
(2.3)

We assume our initial condition is the input image f. Then a steepest descent based on (2.2) can be written as

$$u_t(x) = Lu(x), \quad u_{t=0} = f(x).$$
 (2.4)

We show below that the operator L has many properties which are similar to the Laplacian, or more precisely to the elliptic operator: div $(c(x)\nabla)$ with the symmetric

matrix c(x) > 0. Note that L can be viewed as a natural continuous generalization of the graph Laplacian. Therefore, one may interpret (2.4) as a nonlocal weighted linear diffusion equation.

2.1. Variational Denoising. In the usual way, one can add a convex fidelity term to the convex functional J. For the L^2 fidelity, the denoised image u is the minimizer of

$$E(u, f) = J(u) + \frac{\lambda}{2} ||u - f||_2^2, \qquad (2.5)$$

and u satisfies the Euler-Lagrange equation

$$-Lu + \lambda(u - f) = 0. \tag{2.6}$$

As commonly done, one can also view this as a constrained problem:

$$u := \arg\min J(u), \quad \text{s.t. } \|u - f\|_2^2 = |\Omega|\sigma_n^2,$$
 (2.7)

where σ_n^2 is the variance of an additive noise in a noisy image f. Then λ is viewed as a Lagrange multiplier and one can compute the constrained problem by initializing e.g. with $u|_{t=0} = f$, $\lambda = 1$ and iterating:

$$u_t = Lu + \lambda(f - u), \tag{2.8}$$

$$\lambda = \frac{1}{|\Omega|\sigma_n^2} \int_{\Omega} (u - f) L u dx, \qquad (2.9)$$

using the gradient projection method, as in [51].

2.2. Multichannel signals. Let $\mathbf{f}(x) := (f^1, f^2, \dots f^M)(x)$ be a M channel signal. A multi-valued affinity function is used to compute w(x, y) based on \mathbf{f} (where w(x, y) is the same for all channels). Let $\mathbf{u}(x) := (u^1, u^2, \dots u^M)(x)$ be the regularized signal. The regularizing functional is

$$J_{mc}(u) := \frac{1}{4} \sum_{i=1}^{M} \int_{\Omega \times \Omega} (u^{i}(x) - u^{i}(y))^{2} w(x, y) dx dy, \qquad (2.10)$$

The multi-channel evolution for each channel u^i is

$$u_t^i(x) = \int_{\Omega} (u^i(y) - u^i(x)) w(x, y) dy, \quad u^i|_{t=0} = f^i.$$
(2.11)

See a Fourier analysis of the model and connections to general parabolic equations in Appendix A, extension of the model to Bregman iterations [45] and inverse-scalespace [13] in Appendix B and a nonlocal L^1 functional in Appendix C.

2.3. Properties of L. In the following we show several basic properties of the linear operator L, which will then help establish some results regarding the flow (2.4) and the variational problem (2.5).

PROPOSITION 2.1. The operator L defined by Eq. (2.3) admits the following properties:

(a) If $u(x) \equiv const$ then $Lu(x) \equiv 0$. For w(x,y) > 0, $\forall x, y \in \Omega$, if $Lu(x) \equiv 0$ then $u(x) \equiv const$.

- (b) Let $u(x_0) \ge u(x)$, $\forall x \in \Omega$, then $Lu(x_0) \le 0$. Similarly for the minimum, let $u(x_1) \le u(x)$, $\forall x \in \Omega$, then $Lu(x_1) \ge 0$.
- (c) -L is a positive semidefinite operator, that is $\langle -Lu(x), u(x) \rangle \ge 0$, where $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product.
- (d) $\int_{\Omega} Lu(x) = 0.$

Proof. The first part of Property (a) is immediate. For the second part it is easy to see that for a given point x we have Lu(x) = 0 if for all $y \in \Omega$ either u(x) = u(y) or w(x, y) = 0. Since w(x, y) > 0 we get that u is a constant. Actually one can obtain a weaker condition than a strictly positive w(x, y). This will be shown later in the proof of Lemma 2.2 where the weaker condition (2.12) is used.

Property (b) is straightforward since $w(x, y) \ge 0$.

Property (c) can be validated (using the symmetry w(x, y) = w(y, x)) by:

$$\begin{array}{lll} \langle -Lu(x), u(x) \rangle &=& \int_{\Omega \times \Omega} (u(x) - u(y))w(x,y)u(x)dydx \\ &=& \frac{1}{2} \int_{\Omega \times \Omega} [(u(x) - u(y))u(x)w(x,y) + \\ &+ (u(y) - u(x))u(y)w(y,x)]dydx \\ &=& \frac{1}{2} \int_{\Omega \times \Omega} (u(x) - u(y))^2 w(x,y)dydx \ge 0. \end{array}$$

Property (d) is easily seen by

$$\int_{\Omega} Lu(x) = \frac{1}{2} \int_{\Omega \times \Omega} [(u(x) - u(y))w(x, y) + (u(y) - u(x))w(y, x)]dydx = 0.$$

Let us further require a technical condition on the weight function. Although w(x, y) can have zero values, we shall assume a certain level of connectivity, such that there will not be any disjoint regions where no information is exchanged between them throughout the evolution. We consider the following condition:

$$-L$$
 has a zero eigenvalue of multiplicity 1. (2.12)

This condition is equivalent to stating that -L has only a constant function in its null-space. In graphs, this condition is equivalent to a connected graph, when the linear operator is the graph Laplacian [42]. We can establish a similar relation in our case:

LEMMA 2.2. Condition (2.12) holds if and only if for any two points x, y there exists a sequence: z_1, \ldots, z_k such that $w(x, z_1)w(z_1, z_2)w(z_k, y) > 0$ (that is, every element in the sequence is strictly positive).

Proof. We begin by assuming the sequence exists and showing that the only eigenvector is a constant. First, we notice that if Lu = 0, then for any three points x, y, z where w(x, z) > 0, w(z, y) > 0 we have u(x) = u(z) = u(y). Extending this to larger sequences we have that if Lu = 0 and a sequence exists for x, y, as defined above, then u(x) = u(y). Let us assume there exists an eigenvector v for the zero eigenvalue which is not a constant. Then Lv = 0 and since it is not a constant there must be $x, y \in \Omega$ where $v(x) \neq v(y)$. We reach a contradiction as we assume that between any x, y there exists a sequence.

The other direction can also be proved by contradiction. We assume there exist two points x, y for which no sequence can be established. Thus for any z where w(x, z) > 0 we have w(z, y) = 0 (or else we get a sequence). We can extend this argument and say that all points which have a sequence to x do not have a sequence to y. Let us define by Ω_x the region containing all points with a sequence to x and by

6

 Ω_y the region containing all points with a sequence to y. Surely $\Omega_x \cap \Omega_y = \emptyset$ and we can construct the following function $v(z) = k_1$ if $z \in \Omega_x$, k_2 if $z \in \Omega_y$, 0 otherwise, where $k_1 \neq k_2$ are two constants. We get that Lv = 0, thus the zero eigenvalue has a multiplicity greater than 1.

We can now establish several properties of the flow (2.4).

PROPOSITION 2.3. The flow (2.4) admits the following properties:

(i) The mean value is preserved,

$$\frac{1}{|\Omega|} \int_{\Omega} u(x,t) dx = \frac{1}{|\Omega|} \int_{\Omega} f(x) dx, \, \forall t \ge 0.$$

(ii) The extremum principle holds,

$$\min_{x \in \Omega} (f(x)) \le u(x,t) \le \max_{x \in \Omega} (f(x)), \ \forall x \in \Omega, \ \forall t \ge 0.$$

(iii) For w(x, y) which admits condition (2.12), the solution converges to a constant,

$$u(x,t\to\infty)\equiv const=\int_\Omega f(x)dx.$$

(iv) The following estimate holds:

$$\frac{1}{2}\frac{d}{dt}\|u(x,t)\|_{L^2}^2 \le 0.$$

Proof. (i) can be shown by computing the time derivative of the mean value and using Property (d):

$$\frac{d}{dt} \Big(\frac{1}{|\Omega|} \int_{\Omega} u(x) dx \Big) = \frac{1}{|\Omega|} \int_{\Omega} Lu(x) dx = 0.$$

(ii) is validated by Property (b) as any point x where u(x) is maximal is non-increasing with time, and, similarly, any point x where u(x) is minimal is non-decreasing with time.

Let us first validate (iv). Using Property (c) we can easily obtain:

$$\frac{1}{2}\frac{d}{dt}\|u(x,t)\|_{L^2}^2 = \langle u(x), u_t(x)\rangle = \langle u(x), Lu(x)\rangle \le 0.$$

To prove (iii) we can use the estimate of (iv). It can be shown that the estimate is strictly negative unless $Lu(x) \equiv 0$. Then we use condition (2.12) which dictates that the only steady state solution $u_t = Lu = 0$ is a constant.

Note that from properties (iv) and (i) it follows that $\frac{d}{dt} \operatorname{var}(u(t)) \leq 0$, where $\operatorname{var}(u)$ is the (empirical) variance of u:

$$\operatorname{var}(u) := \frac{1}{|\Omega|} \int_{\Omega} \left(u(x) - \int_{\Omega} u(y) dy \right)^2 dx.$$

Similar results can be obtained for the variational formulation, Eq. (2.5). PROPOSITION 2.4. The minimizer u^{λ} of (2.5) admits the following properties: (i) The mean value is preserved,

$$\frac{1}{|\Omega|} \int_{\Omega} u^{\lambda}(x) dx = \frac{1}{|\Omega|} \int_{\Omega} f(x) dx, \ \forall \lambda \ge 0.$$

(ii) The extremum principle holds,

$$\min_{x}(f(x)) \le u^{\lambda}(x) \le \max_{x}(f(x)), \ \forall x \in \Omega, \ \forall \lambda \ge 0.$$

(iii) For w(x, y) which admits condition (2.12), the solution converges to a constant as $\lambda \to 0$,

$$\lim_{\lambda \to \infty} u^{\lambda}(x) \equiv const = \int_{\Omega} f(x) dx.$$

(iv) The following estimate holds:

$$\frac{1}{2}\frac{d}{d\lambda}\|f - u^{\lambda}\|_{L^2}^2 \le 0.$$

Proof. (i) can be shown by integrating the E-L equation (2.6), using Property (d). One can prove (ii) by contradiction: Let us assume $\max_x(u(x)) > \max_x(f(x))$. Denoting $x_0 \in \{x : u(x) = \max_x(u(x))\}$, we see that $-Lu(x_0) \ge 0$ and $\lambda(u(x_0) - f(x_0)) > 0$, thus the E-L equation is not satisfied. To validate (iv), we can compute the derivative with respect to λ of the E-L equation (2.6) to obtain: $f - u^{\lambda} = (\lambda - L) \frac{du^{\lambda}}{d\lambda}$. Then, using the positive semidefiniteness of -L (Property (c)), we have

$$\begin{split} \frac{1}{2} \frac{d}{d\lambda} \|f - u^{\lambda}\|_{L^{2}}^{2} &= \langle f - u^{\lambda}, -\frac{du^{\lambda}}{d\lambda} \rangle \\ &= -\lambda \|\frac{du^{\lambda}}{d\lambda}\|_{2}^{2} - \langle -L\frac{du^{\lambda}}{d\lambda}, \frac{du^{\lambda}}{d\lambda} \rangle \leq 0 \end{split}$$

To prove (iii) we can use (iv) and for $\lambda = 0$ we have shown in the previous proposition that the solution of the E-L equation with no fidelity term is a constant. \Box

2.4. Weights based on affinity functions. We now explain and formulate the weights w(x, y) which are based on affinity functions. The weights determine the type of regularization induced by the functional J(u).

The basic affinity structure is of similarity between image features. Every data point $x \in \mathbb{R}^2$ is assigned with a feature vector $F_f(x)$. It stands for image features such as gray level value, edge indicator, dominant direction, dominant frequency, etc. We denote by $|\cdot|$ the magnitude of the vector. The region $\Omega_w(x) \subseteq \Omega$ stands for a neighborhood around x where the weights are non-zero. $\Omega_w(x)$ should be symmetric, such that $y \in \Omega_w(x)$ iff $x \in \Omega_w(y)$.

Let us define the following general weight function based on affinities:

$$w(x,y) = \begin{cases} g(F_f(x), F_f(y)) & y \in \Omega_w(x), \\ 0 & \text{otherwise,} \end{cases}$$
(2.13)

where $g(s_1, s_2)$ is a similarity function with the following properties:

- (a) Positive, $g(s_1, s_2) > 0$.
- (b) Symmetric, $g(s_1, s_2) = g(s_2, s_1)$.

(c) Bounded, $g(s_1, s_2) \leq M < \infty$.

(d) Maximal at equality,
$$g(s_1, s_1) \ge g(s_1, s_2), \forall s_1, s_2$$
.

For features in a suitable Banach space (a complete normed space), equipped with the norm $\|\cdot\|_B$, a typical similarity function is

$$g(s_1, s_2) = e^{-(\|s_1 - s_2\|_B/h)^p},$$
(2.14)

where h is a soft threshold parameter which determines the norm values that are considered similar. The power $p \ge 1$ is often set to p = 2 when the Euclidian norm is used .

2.5. Weights examples. Below are some weight functions examples. The first ones are commonly used in image segmentation. We add the nonlocal Yaroslavsky [62] and BCM [10] affinities, which may be useful for some applications.

Intensity, local:

$$g(F_f(x), F_f(y)) = e^{-(|F_f(x) - F_f(y)|/h)^2},$$

$$F_f(x) = f(x),$$

$$\Omega_w(x) = \{y \in \Omega : |y - x| \le \Delta x\},$$

(2.15)

where Δx is the grid size (for images usually $\Delta x = 1$). This results in a 4 nearest neighbors discretization.

Intensity, weighted , semi-local:

$$g(F_f(x), F_f(y)) = e^{-(|F_f(x) - F_f(y)|/h)^2} e^{-|x - y|^2/(2\sigma_d^2)},$$

$$F_f(x) = f(x),$$

$$\Omega_w(x) = \{y \in \Omega : |y - x| \le r\},$$
(2.16)

where σ_d controls the spatial decay and r is the window radius (r should be in the order of σ_d).

For textures, let $K^1(x), \ldots, K^M(x)$ be M linear filters of different directions and frequencies. Let $v^i := u * K^i$, where * denotes convolution. The weights can be computed by (see also [52, 57]):

$$g(F_f(x), F_f(y)) = e^{-(|F_f(x) - F_f(y)|/h)^2},$$

$$F_f(x) = (v^1, \dots, v^M)(x),$$

$$\Omega_w(x) = \{y \in \Omega : |y - x| \le r\},$$
(2.17)

The nonlocal version of Yaroslavsky [62] affinity (in its weighted form) is very similar to (2.15) except that the neighborhood $\Omega_w(x)$ contains the entire image,

$$g(F_f(x), F_f(y)) = e^{-(|F_f(x) - F_f(y)|/h)^2}, F_f(x) = f(x), \Omega_w(x) = \Omega,$$
(2.18)

NL-means [10] affinity:

$$g(F_f(x), F_f(y)) = e^{-(\|F_f(x) - F_f(y)\|_{2,a}/h)^2},$$

$$F_f(x)(u) = f(x) \in B_x,$$

where B_x is a patch centered at x ,

$$\Omega_w(x) = \Omega,$$
(2.19)

Note that (2.18) can be viewed as a special case of (2.19) by taking B_x to be a one pixel patch. One can also modify (2.19) to a semi-local version by restricting the neighborhood: $\Omega_w(x) = \{y \in \Omega : |y - x| \leq r\}$. Also, in a similar way to (2.16), a penalty can be introduced to account for pixels which are further from x. This can be done by setting g in (2.19) to

$$g(F_f(x), F_f(y)) = e^{-(\|F_f(x) - F_f(y)\|_{2,a}/h)^2} e^{-|x - y|^2/(2\sigma_d^2)}.$$

Some remarks: .

- The bilateral filter [58] affinity (gray level) is equivalent to (2.16). The SUSAN filter [53] has also a similar structure to (2.16) (in this case the power in the exponent of g can be larger than 2). See [5] and [23] for relations between the bilateral filter and anisotropic diffusions as well as to other methods which can be viewed as emerging from a Bayesian framework. The latter suggests that statistical image processing can also be accommodated to the above formalism.
- Each of the filters can be approximated (up to a normalization factor) by several iterations of the flow (2.4), when its corresponding affinity is used (e.g. (2.19) for NL-means).
- Certainly, many other denoising schemes (and affinities suggested for segmentation) can be written in the above form.

3. Discretization. The equations are best discretized using the data structure of a graph, where the pixels are the nodes and w(x, y) is represented by the graph weights. One should make the weights sparse enough, so the complexity of the algorithm would be linear. This constraint is usually not very limiting since in most cases if a large window is used many connections have very low weight values which can be ignored (set to zero). If there are many connections with high weight values we suggest to sample them (taking randomly only part of them). The iterative process can usually compensate for this sampling (a broad discussion on that topic is beyond the scope of this paper).

Let u_k denote the value of a pixel k in the image $(1 \le k \le N)$, w_{kl} is the sparsely discrete version of w(x, y). We use the neighbors set notation $l \in \mathcal{N}_k$ defined as $l \in \mathcal{N}_k := \{l : w_{kl} > 0\}$. The flow (2.4) is implemented by the explicit in time forward Euler approximation

$$u_{k}^{n+1} = u_{k}^{n} + \Delta t \sum_{l \in \mathcal{N}_{k}} w_{kl} (u_{l}^{n} - u_{k}^{n}), \qquad (3.1)$$

where $u_k^n = u_k(n\Delta t)$.

All the coefficients on the right side are nonnegative if

$$1 \ge \Delta t \sum_{l \in \mathcal{N}_k} w_{kl}. \tag{3.2}$$

This is the well known CFL restriction on the time step Δt . This leads to maximum norm stability, in fact a maximum principle, for this approximation to (2.4).

The approximation for (2.8) is

$$u_k^{n+1} = u_k^n + \Delta t \sum_{l \in \mathcal{N}_k} w_{kl} (u_l^n - u_k^n) + \lambda \Delta t (f_k - u_k^n)$$
(3.3)

and the analogous time step restriction is

$$1 \ge \Delta t (\sum_{l \in \mathcal{N}_k} w_{kl} + \lambda).$$

3.1. Computing weights for nonlocal means. We present two approximations of w(x, y) to a sparse discrete version w_{kl} . First, we present the semi-local one, which appears to be more useful for denoising. The second algorithm is fully nonlocal and is intended for nonlocal segmentation.

3.1.1. Semi-local version.

Algorithm. For each pixel k:

- 1. Compute the similarity of all the patches in the window [we used 5×5 patch B_x and 11×11 window Ω_w]. Construct \mathcal{N}_k by taking the *m* most similar and the four nearest neighbors of the pixel [we used m = 5].
- 2. Compute the weights w_{kl} , $l \in \mathcal{N}_k$ using (2.19) and set to zero all other connections $(w_{kl} = 0, l \notin \mathcal{N}_k)$.
- 3. Set $w_{lk} = w_{kl}, l \in \mathcal{N}_k$.

Some remarks. In Step 1 taking the four nearest neighbors ensures connectivity of w_{kl} (condition (2.12) is satisfied) and increases the regularity of the process. It may happen that l is not among the m closest nodes to k, but k is in the m closest nodes to l. In this case we add the connections w_{kl} and w_{lk} . We allow up to m additional such connections for each node. Thus, the maximum number of connections is 2m + 4. We have found that m can be very small and produce very good results. In fact enlarging m can decrease performance. This may be understood in a similar manner to the experimental results that increasing the size of the window decreases performance. We use m = 5. For example, the weights computed for the Cameraman image had on average 10.6 connections for each pixel.

CFL. As w(x, y) is bounded by 1, the CFL condition (3.2) can be achieved by setting $\Delta t = \frac{1}{2m+4}$. Usually when adding a fidelity term one need not change the size of the time step and still have a stable flow in practice (for $\lambda < 1$).

3.1.2. Computational complexity. The complexity of computing the weights using the semi-local algorithm is $N \times Window_{size} \times (Patch_{size} + \log m)$. As an example, for 11×11 window with a patch of 5×5 and m = 5 (we can approximate $\log m = \frac{m}{2}$) we need $121 \times (25 + 2.5) \approx 3300$ operations per pixel. Most of the computation time is in this part. For denoising, only a few tens of iterations are usually needed. In the Cameraman example we evolve about 30 iterations with 10.6 nodes per pixel (a total of 318 operations per pixel). Using iterations we actually gain much larger effective support at a very low computational cost. Compare this with a fixed support of a 41×41 window in the original algorithm, which has less support than 30 iterations of a 11×11 window. The computations are considerably larger: $1681 \times 25 = 42025$ operations per pixel. Moreover, in our approach the effective support is very selective (data driven), which we believe significantly contributes to the overall improved performance.

3.1.3. Fast approximation for the fully nonlocal version. This method uses similar ideas to the ones presented in [39]. It is simpler and faster but not as accurate. Still, reasonable denoising performance is achieved (better than the original fully nonlocal version).

The main idea is that for the nonlocal version some fast global computations can be made that help remove many non relevant patches. in [39] various image features were used. As the similarity between patches is based on a distance, one can use a more formal approach of approximating nearest neighbors by similarity hashing, see e.g. [27]. See also a projection approach aimed directly at the fast computations of similarity patches in [30].

Algorithm.

- 1. Compute the mean and the standard deviation of all patches in the image. Create a two dimensional bin table such that all patches in a bin are within a specific range of mean and standard deviation from each other. Both types of bins are spaced in h/2 increments.
- 2. To construct the set \mathcal{N}_k : For each pixel k we consider the 9 bins around it $(3 \times 3 \text{ window in the table, this ensures that patches which are very similar are taken into account). Pick randomly <math>3m$ patches from these bins, check their similarity to the patch of pixel k and take the most similar m of them. Add to \mathcal{N}_k also the four nearest neighbors (to ensure connectivity).
- 3. Compute w_{kl} as in the local algorithm.



FIG. 3.1. MRI of a mouse brain (no synthetic noise added). Top left: input image. Filtered result (middle) and residual (right) are shown for three different methods, from top to bottom: Proposed nonlocal scale-space Eq. (2.4) with BCM weights (Eq. (2.19)). [8 iterations, h = 20]; ROF [51]; ISS based on ROF [13]. [we thank CCB UCLA and the Beckman Institute at Caltech for the image].

4. Denoising Experiments. Our nonlocal regularization framework, presented above, is quite general and can have many variations. There are two main goals in our experiments: first we would like to find a suitable setting which applies well to most image denoising tasks. Then we show that this method is superior than several other methods. We focus here only on the BCM weights, Eq. (2.19), which are more advanced and can serve well for many applications. We first summarize our main findings and then go into a more detailed discussion and explanation regarding the figures.

Our main conclusions can be summarized as follows:

- A semi-local search window $\Omega_w(x) = \{y \in \Omega : |y x| \le r\}$ performs better than a fully nonlocal one $\Omega_w(x) = \Omega$ (at least when no preprocessing is done to remove non-relevant regions in the search window, as in [39]).
- The steepest descent flow (nonlocal scale-space), Eq. (2.4), performs better than the variational minimization, Eq. (2.5), for the same regularizer J(u) and variance of residual.
- The proposed flow performs better (both visually and in terms of SNR) than the original NL-means filter [10] as well as several well-known local PDE-based regularizations [51, 2, 49, 13].

In Fig. 3.1 a noisy MRI image is processed. Both the nonlocal scale-space (NL-SS) and the inverse-scale-space (ISS) perform well visually and do not smooth out fine-scale details, as compared to ROF (second row). As the noise is part of the original image one cannot compute the SNR. The rest of the images were degraded synthetically by white Gaussian noise (in these cases we also measure and compare the SNR). The clean and noisy images are in Fig. 4.1. We first did an extensive comparison of the nonlocal variations (we show for comparison also the filtering using ROF and ISS). The results, in Figs. 4.2 and 4.3, clearly indicate that the semi-local versions are better than the fully nonlocal. Also, the best nonlocal algorithm is the nonlocal scale-space, Eq. (2.4). We currently cannot justify convincingly why the nonlocal variational denoising is inferior. A similar trend was found in experiments with other images. In Fig. 4.4 we compare more closely the NL-scale-space and the original NL-means. One can see in the residual part f - u, right side, that less edges and details of the sails are eroded. Also less texture is smoothed out in the sea part (bottom). Note that the comparison is for the same amount of filtering, that is the variance of the residual is the same and is equivalent to that of the noise. This is not the best result that can be achieved (for both cases), but it is chosen automatically and it gives rather good results. One can achieve in this example somewhat better SNR when the filtering is a little weaker, that is when $var(f-u) \approx 0.9\sigma^2$. This is true for both methods, in any case our proposed denoising performs consistently better for different residual variances (the residual variance can be considered as an alternative scale parameter, see [26] for a broader discussion along with some analysis of the SNR behavior).

In Fig. 4.5 we show the improved performance of the nonlocal (linear) scalespace, Eq. (2.4), over traditional local nonlinear scale-spaces [49, 2]. As can be seen in the residual part, thin lines are less eroded. Compared to [49] (bottom left), there is also no sporadic enhancement of points. In Fig. 4.6 the measures var(u(t))and var(f - u(t)) are shown as a function of time. It is shown empirically that var(f - u(t)) is monotonically increasing with time in general and the algorithm can be stopped according to the discrepancy principle: $var(f - u) = \sigma^2$. Fig. 4.7 depicts the interesting phenomenon that when the image is not very periodic, the optimal window size should be quite small (around 11×11) for both the original NL-means and the proposed iterative method. The trial was done using the following window sizes: 7, 9, 11, 13, 15, 21, 31, 41. In all cases we kept constant the variance of the residual: $\operatorname{var}(f - u) = \sigma^2$.



FIG. 4.1. Test images. Clean image g (left), noisy image f. Top: Cameraman $\sigma = 20$, SNR=9.89. Second row: Sailboat $\sigma = 20$, SNR=4.40. Bottom row: Zebra, $\sigma = 30$, SNR=4.19.

NL-means (original)

NL-SS (nonlocal)



NL-means (original, semi-local)



NL-SS (semi-local)



NL-var (semi-local)



ROF



ISS



FIG. 4.2. Cameraman image filtering result u. Top row: NL-means (nonlocal), SNR=14.59, by nonlocal scale-space (Eq. (2.4)), SNR=15.93. Second row: NL-means (11 × 11 window), SNR=16.43, proposed nonlocal scale-space (11 × 11 window), SNR=17.25. Third row: proposed nonlocal variational denoising (Eq. (2.7)), SNR=16.32, ROF [51], SNR=15.76. Bottom row: ISS (relaxed) based on ROF [13], SNR=16.42. For all methods $var(f - u) = \sigma^2$.



FIG. 4.3. Cameraman image, corresponding residual parts f - u.



FIG. 4.4. Top: original NL-means algorithm, SNR=11.62. Bottom: NL scale-space, SNR=12.71. For both methods the semi-local versions is used, window 11×11 , patch 5×5 , $var(f-u) = \sigma^2$.



FIG. 4.5. Comparison of NL-linear scale-space with two local nonlinear scale spaces. Top: NL-SS (semi-local), SNR=12.03. Middle row: TV-flow, SNR=10.36. Bottom row: Perona-Malik $(k_{pm} = \sigma\sqrt{2})$, SNR=10.37. For all methods $var(f - u) = \sigma^2$.



FIG. 4.6. Variance of u and (f - u) as a function of time (iterations) of the nonlocal scale-space process (Cameraman image). Although monotonicity of var(f - u) is not guaranteed, in practice it is increasing with time for most input images, thus a discrepancy principle can be used as the stopping criterion $(var(f - u(t)) = \sigma^2)$.



FIG. 4.7. SNR as a function of the window size (for the Cameraman and Sailboat images). For both the original NL-means and the NL scale-space a rather small (semi-local) neighborhood is preferred for many natural images.

5. Supervised Segmentation Algorithm. In this section a nonlocal segmentation algorithm is outlined based on methods that are frequently used in the field of classification and machine learning. A generic two class kernel-based classification is: given n labelled points $(x_i, g_i) \in \mathbb{R}^d \times G$, $G = \{-1, +1\}$. Generalize the labels to the entire domain by

$$u(x) = \operatorname{sign}\left(\sum_{i=1}^{n} g_i K(x, x_i) + b\right),$$

where K(x, y) is a kernel (usually data driven).

A very simple choice of the kernel is a Gaussian function, or the Green's function of linear diffusion. Here we use the nonlocal family of Green's functions generated by L. Similar ideas using the graph-Laplacian were proposed e.g. in [52, 60, 29, 37, 8].

Our motivation and analysis is continuous rather than discrete. We show how some intuition can be gained by analyzing the weighted diffusion (heat) equation. The segmentation of a step signal is analyzed. We expect that farther analysis in this spirit of important special cases may give better understanding on the advantages and limitations of such methods. An interesting connection is shown between denoising and segmentation, where the same flow is used for both tasks, and only the initial conditions are different. We first present the algorithm.

Let f be the input image and w(x, y) the corresponding weights. Let Ω_0^O be an initial set which is part of the object to be segmented. Let Ω_0^B be an initial set which

is part of the background. Ω_0^O and Ω_0^B are disjoint $(\Omega_0^O \cap \Omega_0^B = \emptyset)$ and not necessarily connected. In our algorithm these regions are defined by the user, who marks them for a given image, specifying the object to be segmented. The supervised (but not interactive) algorithm is:

1. Initialize

$$u_0 := \begin{cases} 1, & x \in \Omega_0^0 \\ -1, & x \in \Omega_0^B \\ 0, & \text{otherwise} \end{cases}$$

2. Evolve for a duration T the flow

$$u_t(x) = Lu(x), \quad u_{t=0} = u_0(x).$$
 (5.1)

3. Define Ω^O , the set of nodes approximating the *Object*, by: $\Omega^O := \{x \in \Omega^O : u(x,T) > 0\}$, and the *Background* by the complement: $\Omega^B = \Omega - \Omega^O$.

Note that the only difference between (2.4) and (5.1) is in the initial conditions: both rely on the image f and the affinities to compute w(x, y), however (5.1) evolves the initial user marks and not the input image.

5.1. Multiple objects. For multiple objects we generalize the algorithm using a multichannel flow. Let $\Omega_0^1, \Omega_0^2 \dots \Omega_0^M$ be M disjoint sets of nodes which are part of M regions to be segmented (including the background). This data is defined by the user. The multiple objects segmentation algorithm is:

1. Initialize a M channel signal u^i , i = 1, ..., M as follows

$$u_0^i := \begin{cases} 1, & x \in \Omega_0^i \\ 0, & \text{otherwise} \end{cases}$$

2. Evolve for a duration T the flow

$$u_t^i = Lu^i(x), \quad u_{t=0}^i = u_0^i(x), \quad i = 1, \dots, M.$$
 (5.2)

- 3. Define Ω^i , the set approximating region *i*, by:
- $\Omega^i := \{ x \in \Omega : i = \arg \max_{i} \{ u^j(x, T) \} \}.$

Note that the above algorithm can be related also to image colorization, where a gray-scale image is colored by user-provided coloring examples (see an explanation of the problem and a highly effective solution in [63]). We recently became aware of the fact that similar ideas were mentioned by Szlam in [55].

5.2. Motivation and Analysis. Our motivation and analysis are both based on the (local) linear weighted diffusion equation:

$$u_t(x) = \operatorname{div}(c(x)\nabla u), \quad u|_{t=0} = u_0,$$
(5.3)

where c(x) is a spatially varying diffusion coefficient. The algorithm is then extended to the "nonlocal diffusion" flow, Eq. (5.1), where w(x, y) replaces the role of c(x).

5.2.1. Physical motivation: pool with a barrier. We would like to illustrate the segmentation problem by a simple physical problem which may give more intuition regarding the algorithm.

The model consists of a pool with water at a certain constant temperature T_0 . Inside the pool there is a thin barrier that separates the pool into two parts. The



FIG. 5.1. Physical motivation for the segmentation algorithm. Illustration of a pool with a barrier.

goal is to locate the barrier (see the illustration in Fig. 5.1). We are given that the conductance of heat of the barrier is much lower than that of water. A simple solution to the problem is to pour cold water on one side of the pool, the left for instance, and pour hot water on the other side, wait a few moments, and use the following decision rule: any region with temperature above T_0 is to the right of the barrier, any region with temperature below T_0 is to the left of the barrier.

To model this problem we use the diffusion (heat) equation with spatially varying diffusion (conductance) coefficient. For simplicity we assume a unit diffusion coefficient c(x, y) = 1 everywhere except at the barrier and $c(x, y) = \epsilon$ at the barrier. More details on the physical problem can be seen in [25].

5.2.2. Analytic solution. The general solution of Eq. (5.3) is of the form:

$$u(x,t) = \int_{\Omega} K(y,x;t)u_0(y)dy, \qquad (5.4)$$

where K(y, x; t) is a non-stationary Green's function.

Let us first write a solution for the one dimensional case where c(x) is piecewise constant. The case can be viewed as a conductance problem for *n*-layer composite slabs with a constant conductivity in each layer. See [41] Ch. 9 for a detailed formal solution of the general problem. In our case we assume perfect thermal conductivity with continuity of both the temperature and flux on the boundaries of the inner layers. Neumann boundary conditions are assumed for the two outer layers. We denote by $x_k, 0 \le k \le n$, the n + 1 boundaries of the *n* layers, where $x_k < x_{k+1}$ and x_0, x_n coincide with the left and right boundaries of the domain, respectively. u_{0k} is the value of u_0 in layer k: $u_{0k}(x) := u_0(x), x \in [x_{k-1}, x_k]$, and similarly for $u_k(x, t), c_k$. The solution for each layer $u_k(x, t), 1 \le k \le n$, is:

$$u_k(x,t) = \sum_{i=1}^{\infty} \frac{\psi_k(\mu_i, x)}{N_i} e^{-\mu_i^2 t} \tilde{u_{0i}},$$
(5.5)

where N_i is a normalization factor

$$N_i := \sum_{k=1}^n \int_{x_{k-1}}^{x_k} \psi_k^2(\mu_i, x) dx,$$

 \tilde{u}_{0i} is defined as

$$\tilde{u}_{0_i} := \sum_{k=1}^n \int_{x_{k-1}}^{x_k} \psi_k(\mu_i, x) u_{0_k}(x) dx,$$

and $\mu_i, \psi_k(\mu_i, x)$ are the eigenvalues and eigenfunctions, respectively, solving

$$\frac{d^2\psi_k(\mu_i, x)}{(dx)^2} + \frac{\mu_i}{c_k}\psi_k(\mu_i, x) = 0.$$

5.2.3. Step edge - an approximate solution. In order to gain some more intuition on the behavior of the solution, we would like to analyze a simple but important case of the segmentation of a step signal.

Our setting is as follows. Let f(x) be a one dimensional step signal: f(x) := 1 if x > 0, 0 otherwise. Let $c(x) = e^{-(|\nabla f|/k)^2}$. In this case we get

$$c(x) = \begin{cases} 0 & x = 0, \\ 1 & \text{otherwise.} \end{cases}$$

This trivially decouples the domain into two separate regions with no connection between them and our algorithm will give a perfect segmentation. However this result is not very helpful, since in a discrete setting c(x) will not attain zero at x = 0. Therefore we would like to analyze the discrete setting with a spatial resolution h. We can assume the conductivity is defined between two consecutive grid points and estimate $c_{i+1/2} := e^{-(|f_{i+1}-f_i|/(kh))^2}$. To gain symmetry around zero, we shift the problem by $\frac{1}{2}$ to have:

$$c_i = \begin{cases} \epsilon & i = 0, \\ 1 & \text{otherwise,} \end{cases}$$

where $\epsilon = e^{-(1/(kh))^2}$ (we assume $kh \gg 1$). We can model it continuously by the following piecewise constant conductivity:

$$c(x) = \begin{cases} \epsilon & x \in \left(-\frac{h}{2}, \frac{h}{2}\right), \\ 1 & \text{otherwise.} \end{cases}$$

Clearly, we can obtain an analytic solution of the problem using (5.5). However, we prefer to present an alternative way in this special case, which may better describe the characteristics of the solution. For simplicity we assume an unbounded domain $x \in \mathbb{R}$. We add the superscripts $-, +, \epsilon$ to denote the negative, positive and center regions, respectively, such that:

$$u^{-} := u(x), x \le -\frac{h}{2}; u^{+} := u(x), x \ge \frac{h}{2}; u^{\epsilon} := u(x), x \in (-\frac{h}{2}, \frac{h}{2}),$$

and similarly for the flux $\mathbf{J}(x) := c(x)u_x$ in each region:

$$\mathbf{J}^{-}(x) := c^{-}(x)u_{x}^{-} = u_{x}^{-}; \ \mathbf{J}^{+}(x) := c^{+}(x)u_{x}^{+} = u_{x}^{+}; \ \mathbf{J}^{\epsilon}(x) := c^{\epsilon}(x)u_{x}^{\epsilon} = \epsilon u_{x}^{\epsilon}.$$

We use the hypothesis of continuity in the temperature and flux at the boundary of each layer:

$$\begin{aligned} u^+(\frac{h}{2}) &= u^{\epsilon}(\frac{h}{2}), \quad u^-(-\frac{h}{2}) = u^{\epsilon}(-\frac{h}{2}), \\ \mathbf{J}^+(\frac{h}{2}) &= \mathbf{J}^{\epsilon}(\frac{h}{2}), \quad \mathbf{J}^-(-\frac{h}{2}) = \mathbf{J}^{\epsilon}(-\frac{h}{2}). \end{aligned}$$

Moreover, we assume h is small (a thin slab) so that the gradient of u^{ϵ} can be approximated by a linear function and the flux by a constant: $\mathbf{J}^{\epsilon}(-\frac{h}{2}) = \mathbf{J}^{\epsilon}(\frac{h}{2})$. We can thus obtain the following simplified boundary conditions relating u^+ to u^- :

22

Let us examine the case of a Green's function at the positive side, that is $u_0 = \delta(x - x_0)$, $x_0 > \frac{h}{2}$. As the problem is symmetric, a similar analysis will hold for the negative part. We can have a good approximation of the solution for u^+ using the method of images. For the case of two media where $c(x) = 1, x \ge \frac{h}{2}; \epsilon, x < \frac{h}{2}$ the solution for u^+ can be computed by assuming a homogeneous medium and adding a mirror image source $A\delta(x + x_0 - h)$ where $A = \frac{1 - \sqrt{\epsilon}}{1 + \sqrt{\epsilon}}$. The solution for $x < \frac{h}{2}$ is in the order of $1 - A = \frac{2\sqrt{\epsilon}}{1 + \sqrt{\epsilon}} \ll 1$. With a third medium at $x < -\frac{h}{2}$, as in our case, there will be feedbacks in the order of 1 - A which are small, compared with u^+ , as $A \gg 1 - A$ for small ϵ . u^- is approximated by using (5.6) and the relation $u^+(\frac{h}{2}) \gg u^-(-\frac{h}{2})$ for ϵ small enough. The problem simplifies to a semi-infinite slab at $x \in (-\infty, -\frac{h}{2})$ given the following flux at the boundary:

$$\mathbf{J_0}(t) = \mathbf{J}^-(-\frac{h}{2}, t) = \frac{\epsilon}{h}u^+(\frac{h}{2}, t).$$

A solution is obtained using Duhamel's theorem (see e.g. [16], Ch. 2), which for a semi-infinite slab originating at x = 0 yields

$$u(x) = 2 \int_0^t \mathbf{J}_{\mathbf{0}}(t-\tau) g_{\sigma(\tau)}(x) d\tau.$$

We can thus have the following approximate solution of the Green's function for $x_0 > \frac{h}{2}$:

$$\begin{aligned} u^{+}(x,t) &\approx g_{\sigma(t)}(x-x_{0}) + Ag_{\sigma(t)}(x+x_{0}-h) \\ u^{-}(x,t) &\approx \frac{2\epsilon(1+A)}{h} \int_{0}^{t} g_{\sigma(t-\tau)}(x_{0}-\frac{h}{2})g_{\sigma(\tau)}(x+\frac{h}{2})d\tau, \end{aligned}$$
(5.7)

where $g_{\sigma(t)}$ is a one dimensional Gaussian: $g_{\sigma(t)}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^2/(2\sigma^2)}$ and $\sigma(t) = \sqrt{2t}$.

In Fig. 5.2 the approximate solution of (5.7) is shown and compared to a simulated solution. A good agreement is shown between both solutions. In Fig. 5.3 some steps in the evolution are shown. It is clear that for small ϵ the segmentation is very robust and does not rely on the specific stopping time or specific location of the marks (we remind that for a wrong segmentation to happen the magnitude of u^- of one mark should be larger than the magnitude of u^+ of the other mark). A specific example of evolving both marks is depicted in Fig. 5.4. The case of a two dimensional step edge is shown in Fig. 5.5. Here the image is quite noisy and ϵ is not very small. Still we get a clear distinction between the sides. The case of a noisy circle, with points inside and outside the circle, is shown in Fig. 5.6. An example of the nonlocal flow, where the weights are based on the (fully nonlocal) BCM affinity (2.19), is given in Fig. 5.7. One can clearly see the nonlocal support of the Green's function after only 30 iterations and that the dominant regions have similar feature as the initial condition.

5.2.4. Remark: curve length and noise sensitivity. The method does not take into account curve length, as done in most of the variational segmentation algorithms ([43, 31, 17, 18] and their extensions). Therefore it is less robust against noise, however it may segment better highly non-convex shapes (with long boundaries) and does not tend to smooth corners. If curve shortening is desired for certain applications, it can be gained implicitly by denoising the input image f beforehand, using [51] for instance. Post-processing is also possible by applying curve-smoothing methods (such as mean-curvature-flow) to the segmentation result.



FIG. 5.2. Comparing the approximate analytic solution, Eq. (5.7), to simulation results. From left: the simulated and analytic solutions superimposed, enlargement of the negative side, difference between the solutions: $E = u_{simulated} - u_{analytic}$: positive side, negative side (right). $[x_0 = 5.5, t = 100, k = 3, h = 1, \epsilon = e^{-(1/(kh))^2}]$.



FIG. 5.3. Example of the evolution in time of the approximate analytic solution, Eq. (5.7), at times t = 10, 100, 400 (from left to right, respectively). [$x_0 = 14.5$, other parameters as above].

6. Segmentation Experiments. Below are some experiments testing the supervised segmentation algorithm. First, we make some experiments which are local and approximate the above discussion and analysis regarding linear weighted diffusion (Eq. (5.3)). In this case an edge which induces low diffusivity can indeed isolate regions only if it has considerable support. Edges with small support which are due to noise or sporadic small outliers have little effect. Outliers take their values according to the dominant value in the region which surrounds them (as some "labeling information" does penetrate even with low diffusivity). This most often makes the correct assignment of the regions to Object and Background. Next, we show a more advanced segmentation using the BCM weights. We also use here the fully nonlocal version, demonstrating that in this case the labelling information is not constrained by the spatial location of the initial marks. Certainly, the degree of localization of the weights and thus the entire process can be fully controlled by the user and will vary according to the specific application.

In Figs. 5.8 - 5.12 the experiments are done with the simple affinity (2.15), so that $w(x, y) = e^{-(|f(x)-f(y)|/h)^2}$ is evaluated only at the 4 nearest neighbors. In Fig. 5.8 two regions with an edgy boundary and outliers are separated. This corresponds to the pool with a barrier problem discussed above. In Fig. 5.9 we show that "leaks in the barrier", represented by local blurs near the shared boundary of the two regions, are handled well.

In Fig. 5.10 a polygon is segmented, keeping the corners. In Fig. 5.11 we show that the algorithm is quite robust to the user data (marks), as long as they are correct.

In Fig. 5.12 a horse silhouette image is processed with additive white noise and some spots both inside and outside the object. The segmentation process can perform with this moderate levels of noise, corners and fine details are correctly segmented.

In Fig. 5.13 we show a typical case where the algorithm can fail. The second problem is illustrated in Fig. 5.13, the background marks are very uneven and do not



FIG. 5.4. Example of the segmentation algorithm using the approximate analytic solution. The user marks an object point at x = 16 and a background point at x = -10. The corresponding initial condition is $u_0(x) = -\delta(x+10) + \delta(x-16)$. The evolution is for the times t = 10, 100, 400 (from left to right, respectively). [k = 3, h = 1].



FIG. 5.5. Example of a Green's function, the input image is a noisy step. From left: input f, light point marks the location of the function at t = 0 (superimposed on f with a low contrast), Green's function at t = 40.

surround the object. This does not mean a failure in all cases, but the results then depend much more strongly on the specific parameters (h, stopping time) noise level etc. We interpret this result as though the marks of the object are "closer" to the right part of the background than the background mark on the left. For a fully nonlocal algorithm, this type of failure will not happen as only features similarities determine the evolution and not spatial distance. A different type of failure would happen if the marks are not given correctly. See our report [25] for an example.

Figs. 5.14 - 5.15 show experiments with NL-means metric. We use the fully nonlocal scheme here. In this case a single segment can contain many objects which are very far apart, but have the same features. The image contains three main types of objects, in Fig. 5.14 the dark part of the cells is not marked as either object or background. The ambiguity is reflected by the fact that those regions stay gray (second row), meaning the value is very close to zero. Apparently the values are slightly positive and the thresholding operation at the end classifies them as part of the object. If the dark regions are to be marked as background, as done in Fig. 5.15, the final classification changes accordingly. Note that very small and local marks are needed for the algorithm to give a good classification of the different objects.

7. Discussion and Conclusion. It was shown in this paper how various denoising methods can be realized using a nonlocal weighted quadratic functional. Specifically, the nonlocal means algorithm [10] can be generalized in various manners (as a scale-space, convex variational problem or as inverse scale space [13]). Our current experiments indicate that for images the nonlocal (forward) scale-space approach performs best. It is also the fastest to compute. The computation complexity is similar to the original algorithm (for the same window and patch sizes), as most of the pro-



FIG. 5.6. Example of a Green's function, the input image is a noisy circle. Top, from left: input f, location of the function at t = 0 (inside the circle), Green's function at t = 40. Bottom, from left: location of the function at t = 0 (outside the circle), Green's function at t = 40.

cessing time is devoted to computing the weights.

The method consistently outperforms the original nonlocal means algorithm. At this point we cannot claim to fully understand the reasons for that. Three major differences between the methods can account for the improved performance: we use a different normalization which keeps symmetric similarities and does not tend to blur rare and singular regions; we use very sparse weights (about 10 on average for each pixel), which are chosen selectively; the process is iterative - allowing more complex interaction between the pixels and extended support of the averaging, which is datadriven (and not predetermined). As was shown here experimentally (see Fig. 4.7), simply extending the support by using a larger window (or the entire image, in the extreme case) usually degrades the performance, unless the image is highly periodic.

The current framework is spatially varying but still linear. Several generalizations to a nonlinear framework are possible. One alternative is to use a L^1 -based functional (as suggested in Appendix C), which still retains a convex framework. Another possibility is to follow the (local) nonlinear diffusion mechanism [49] and update the weights at each iteration. Thus w(x, y) would depend on the evolved image u and not on the input image f. This can increase the computational complexity and also may raise some stability issues. A nonlinear framework may add robustness when the weights which are calculated according to the input image are not very reliable (due to high noise) and some strong connections contain outliers. One may use the proposed convex regularizer for applications other than denoising or segmentation. For example, a nonlocal generalization of variational inpainting [4] can be suggested.

Additionally, it was demonstrated how nonlocal supervised segmentation can be performed, following ideas of classification by kernel based methods. The main idea is to extend the user marks ("labels") of object and background to the entire image by using the image adaptive Green's function as the kernel. This translates to simply diffusing the marks (and thresholding at the end).



FIG. 5.7. Example of Green's function based on NL-means. Top: input image. Second row: location of two functions at t = 0. Third row: Green's functions after 30 iterations. Fourth row: mesh plots.

G. GILBOA AND S. OSHER



FIG. 5.8. Edgy step with outliers.



FIG. 5.9. Edgy step with two blurry regions on the edge.

Object

Background

Input image





FIG. 5.10. Polygon with outliers.



FIG. 5.11. Polygon as previous figure. Different marks given by the user (for the same objective).



FIG. 5.12. Horse silhouette, with spots in both object and background, white Gaussian noise is added ($\sigma = 15$). 4-neighbor scheme. In the second and third rows, the advancement of the information with the iterations is illustrated (depicting values above 0.001 in white and below -0.001 in black).



FIG. 5.13. Failure example: Marks of the background are too sparse and uneven.



FIG. 5.14. Segmenting with the nonlocal flow, based on NL-means. One background mark is given. u is shown in the range [-0.001, 0.001], values over / under are saturated to white / black, respectively. Object is defined by u > 0.



FIG. 5.15. Two background marks are given.

Acknowledgements. We would like to thank Jean-Michel Morel (ENS-Cachan), Antoni Buades (U. de les Illes Balears) and Arthur Szlam (UCLA) for useful and stimulating discussions.

Appendix A. Fourier Analysis. It is instructive to compare equation (2.4) with standard linear parabolic equations. For example, suppose w(x, y) = w(|x - y|), (which is definitely not recommended for practical calculations). For simplicity, we let Ω be all of \mathbb{R}^N . We now have

$$u_t = -\int (u(x) - u(y))w(|x - y|)dy.$$

Let $\hat{w}(\xi) = \int_{\mathbb{R}} e^{-(ix\cdot\xi)} w(|x|) dx$ be the Fourier transform of w. The evolution equation becomes

$$\hat{u}_t = \hat{u}(\xi) [\hat{w}(\xi) - \hat{w}(0)].$$

Because w(x) = w(|x|), we have

$$\hat{w}(\xi) - \hat{w}(0) = -\int \frac{(\xi - x)^2}{2} w(|x|) dx + O(|\xi|^4)$$

 So

$$L(u) \approx \frac{1}{2} \sum_{i,j=1}^{N} a_{ij} \frac{\partial^2}{\partial x_i \partial x_j}(u)$$

with $a_{ij} = \frac{1}{2} \int_{\mathbb{R}^N} x_i x_j w(|x|) dx$. This is a symmetric parabolic equation. For example if

$$w(x,y) = \frac{\delta(x-y-h) + \delta(x-y+h)}{2h^2} = \frac{\delta(|x-y|-h)}{2h^2}, \in \mathbb{R}^1$$

then $\hat{w}(\xi) = \cos \frac{(h\xi)}{h^2}$ and $\hat{w}(\xi) - \hat{w}(0) = -\frac{\xi^2}{2} + O(h^2\xi^4)$ so $L(u) \approx \frac{1}{2} \frac{\partial^2}{\partial x^2}$. Smoothing out this w by, for example, using

$$\hat{w}_{\sigma}(\xi) = \cos(h\xi) \frac{e^{-\sigma h^2 |\xi|^2}}{h^2}$$

(which means convolving w with a Gaussian of variance σh^2) leads us to $\hat{w}_{\sigma}(\xi) - \hat{w}_{\sigma}(i) = \xi^2 \left(-\frac{1}{2} - \sigma\right)$ so $L(u) \approx \left(\frac{1}{2} + \sigma\right) \frac{\partial^2}{\partial x^2}$.

A.1. Link with more general parabolic equations. Consider the parabolic equation

$$u_t = \sum_{j=1}^N \frac{\partial}{\partial x_i} (a_{ij}(x)u_{x_j}) = Mu$$

to be solved in $\Omega \subset \mathbb{R}^n$, a_{ij} is a positive definite symmetric matrix, with $\frac{\partial u}{\partial n} = 0$ on $\partial \Omega$. Then

$$< -Mu(x), u(x) > = \int_{\mathbb{R}^N} \sum_{i,j=1}^N a_{ij}(x) u_{x_i} u_{x_j}.$$

Compare this with

$$< -Lu(x), u(x) > = \frac{1}{2} \int \int (u(x) - u(y))^2 w(x, y) dx dy$$
$$\approx \frac{1}{2} \int \sum_{i,j=1}^N b_{ij}(x) u_{x_i} u_{x_j}$$

where

$$b_{ij} = b_{ij}(x) = \int_{\Omega} w(x,y)(y_i - x_i)(y_j - x_j)dy.$$

So we can expect to associate L with M up to quadratic terms by defining

$$a_{ij} = \frac{1}{2} \int_{\Omega} w(x,y)(y_i - x_i)(y_j - x_j)dy.$$

Appendix B. Bregman Iteration and Inverse Scale Space.

In [13],[15] we developed an inverse scale space (ISS) and relaxed inverse scale space approach to denoising. It was based on the continuous limit of Bregman iterations devised in [45], not only for denoising but for blind deconvolution [40] and other reconstruction tasks [61],[7] as well as being a useful theoretical device to obtain sharp estimates in standard reconstruction methods [14]. Briefly, to reconstruct an image from a given data f we begin with the variational problem

$$u_1 = \arg\min_u (J(u) + \lambda H(u, f))$$

(see [45] for more details). This leads to the sequence

$$u_{k} = \arg\min_{u} (J(u) - J(u_{k-1}) - \langle u - u_{k-1}, p_{k-1} \rangle + \lambda H(u, f))$$

 $k = 1, 2, \ldots$, with $u_0 = 0$ and $J(0) = p_0 = 0$, for $\lambda > 0$, where $p_{k-1} = p(u_{k-1})$ and p(u) is an element of the subgradient of J(u). Under reasonable hypotheses it follows that $u_k \to \tilde{u}$ the minimizer of H(u, f) monotonically, but, more importantly, the Bregman distance between u_k and g, g a "denoised version of \tilde{u} ", which means $J(g) < \infty$, decreases until u_k gets too close to \tilde{u} . See [45] for the precise results.

The Bregman distance is defined by

$$D(g, u) = J(g) - J(u) - \langle g - u, p(u) \rangle$$

For J(u) defined by (2.1), we have

$$\begin{array}{lll} D(g,u) & = & \frac{1}{4} \int_{\Omega \times \Omega} ((g(x) - u(x)) - (g(y) - u(y)))^2 w(x,y) dx dy \\ & = & J(g-u). \end{array}$$

The inverse scale space equation associated with (2.5) is

$$-Lu_t = (f - u)$$

or

$$u_t = (-L)^{-1}(f - u)$$

with u(0) = 0 and the normalization $\int u = \int f = 0$ is required.

Solving this involves inverting L and evaluating (L^{-1}) at every time step, which is computationally nontrivial.

Instead we use the relaxed inverse scale space approximation [13]:

$$\frac{\partial u}{\partial t} = L(u) + \lambda(f - u + v)$$
$$\frac{\partial v}{\partial t} = \alpha(f - u)$$

for $\lambda, \alpha > 0, \ \alpha \leq \frac{\lambda}{4}$ with initial data u(0) = v(0) = 0.

Appendix C. L^1 functional. We may also consider the following functional which can be interpreted as a weighted non-local total-variation:

$$J_1(u) := \frac{1}{2} \int_{\Omega \times \Omega} |u(x) - u(y)| w(x, y) dx dy,$$
 (C.1)

with the the corresponding steepest descent:

$$u_t(x) = -J_1'(u)(x) = \int_{\Omega} \operatorname{sign}(u(y) - u(x))w(x, y)dy.$$
(C.2)

The non-local analogue of this framework to ROF [51] is then

$$E_1(u, f) = J_1(u) + \frac{\lambda}{2} ||f - u||_2^2.$$
(C.3)

For more details on extending the nonlocal framework from a linear to a convex one see [24].

Appendix D. Iterating NL-means. In [12] and also in [55] the NL-means algorithm is applied iteratively. We can formulate this as:

$$u^{k+1}(x) = \int_{\Omega} \frac{u^k(y)w(x,y)}{\int_{\Omega} w(x,\hat{y})d\hat{y}} dy,$$

where $w(x,y) = e^{-d_a(u(x),u(y))/h^2}$ and $d_a(\cdot, \cdot)$ is defined in (1.3). This can be understood as an analogous to Jacobi's iterative method for solving

$$Lu(x) = \int_{\Omega} (u(y) - u(x))w(x, y)dy = 0,$$

where we initialize with f. Naturally, the final solution as $k \to \infty$ is not of interest (as it is simply a constant) and one iterates only a limited number of times. This type of iterations induce large "time steps" and the accurate amount of filtering may be somewhat hard to control.

Our approach in contrast is a gradient descent one (for minimizing J(u)), which can be written iteratively as:

$$u^{k+1}(x) = u^k(x) + \Delta t \int_{\Omega} (u^k(y) - u^k(x))w(x,y)dy$$

with f as initial condition.

34

REFERENCES

- R. Adams and L. Bischof. Seeded region growing. IEEE Trans. Pattern Anal. Mach. Intell., 16(6):641–647, 1994.
- [2] F. Andreu, C. Ballester, V. Caselles, and J. M. Mazn. Minimizing total variation flow. Differential and Integral Equations, 14(3):321–360, 2001.
- [3] G. Aubert and P. Kornprobst. Mathematical Problems in Image Processing, volume 147 of Applied Mathematical Sciences. Springer-Verlag, 2002.
- [4] C. Ballester, M. Bertalmo, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions On Image Processing*, 10(8):1200– 1211, 2001.
- [5] D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847, 2002.
- [6] M. Belkin and P. Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. In COLT, pages 486–500, 2005.
- [7] B. Berkels, M. Burger, M. Droske, O. Nemitz, and M. Rumpf. Contour extraction based on anisotropic image classification, 2006. UCLA CAM Report 06-42.
- [8] Y. Boykov and M-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *International Conference on Computer Vision*, volume I, pages 105–112, 2001.
- Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222– 1239, 2001.
- [10] A. Buades, B. Coll, and J-M. Morel. On image denoising methods. SIAM Multiscale Modeling and Simulation, 4(2):490–530, 2005.
- [11] A. Buades, B. Coll, and J-M Morel. Neighborhood filters and PDE's. Numerische Mathematik, 105(10):1–34, 2006.
- [12] A. Buades and J-M. Morel. 2006. Private communication.
- [13] M. Burger, G. Gilboa, S. Osher, and J. Xu. Nonlinear inverse scale space methods. Comm. in Math. Sci., 4(1):179–212, 2006.
- [14] M. Burger and S. Osher. Convergence rates of convex variational regularization. Inverse Problems, 20(5):1411–1421, 2004.
- [15] M. Burger, S. Osher, J. Xu, and G. Gilboa. Nonlinear inverse scale space methods for image restoration. In VLSM '05, volume 3752 of Lecture Notes in Computer Science, pages 25–36, 2005.
- [16] H.S. Carslaw and J.C. Jaeger. Conduction of heat in solids (2nd ed.). Oxford University Press, London, 1959.
- [17] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. International Journal of Computer Vision, 22(1):61–79, 1997.
- [18] T. Chan and L. Vese. Active contours without edges. IEEE Trans. Image Processing, 10(2):266– 277, 2001.
- [19] T.F. Chan and J. Shen. Image Processing and Analysis. SIAM, 2005.
- [20] F. Chung. Spectral Graph Theory. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.
- [21] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker. Geometric diffusion as a tool for harmonic analysis and structure definition of data, part i: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, 2005.
- [22] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In ICCV (2), pages 1033–1038, 1999.
- [23] M. Elad. On the bilateral filter and ways to improve it. IEEE Transactions On Image Processing, 11(10):1141–1151, 2002.
- [24] G. Gilboa, J. Darbon, S. Osher, and T.F. Chan. Nonlocal convex functionals for image regularization, 2006. UCLA CAM Report 06-57.
- [25] G. Gilboa and S. Osher. Nonlocal linear image regularization and supervised segmentation, 2006. UCLA CAM Report 06-47.
- [26] G. Gilboa, N. Sochen, and Y.Y. Zeevi. Estimation of optimal PDE-based denoising in the SNR sense. *IEEE Trans. on Image Processing*, 15(8):2269–2280, 2006.

- [27] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In 25th International Conference on Very Large Data Bases (VLDB), 1999.
- [28] L. Grady. Random walks for image segmentation. to appear in IEEE Trans. on Pattern Analysis and Machine Intelligence, 2006.
- [29] L. Grady and G. Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In ECCV, Workshop on Computer Vision Approaches to Medical Image Analysis, pages 230–245, 2004.
- [30] Y. Hel-Or and H. Hel-Or. Real-time pattern matching using projection kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(9):1430–1445, 2005.
- [31] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. International Journal of Computer Vision, 1(4):321–331, 1987.
- [32] C. Kervrann and J. Boulanger. Unsupervised patch-based image regularization and representation. In Proc. European Conf. Comp. Vision (ECCV'06), Graz, Austria, 2006.
- [33] R. Kimmel, R. Malladi, and N. Sochen. Images as embedding maps and minimal surfaces: Movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 39(2):111–129, 2000.
- [34] R. Kimmel, N. Sochen, and R. Malladi. From high energy physics to low level vision. In First International Conference on Scale-Space Theory in Computer Vision, Springer-Verlag, LNCS 1252, pages 236–247, 1997.
- [35] S. Kindermann, S. Osher, and P. Jones. Deblurring and denoising of images by nonlocal functionals. SIAM Multiscale Modeling and Simulation, 4(4):1091 – 1115, 2005.
- [36] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Mach. Intell., 26(2):147–159, 2004.
- [37] R.I. Kondor and J.D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In ICML, pages 315–322, 2002.
- [38] Y. Li, J. Sun, C-K. Tang, and H-Y. Shum. Lazy snapping. ACM Trans. Graph., 23(3):303–308, 2004.
- [39] M. Mahmoudi and G. Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters*, 12(12):839–842, 2005.
- [40] A. Marquina. Inverse scale space methods for blind dconvolution, 2006. UCLA CAM Report 06-36.
- [41] M.D. Mikhailov and M.N. Ozisik. Unified analysis and solutions of heat and mass diffusion. John Wiley and Sons, 1983.
- [42] B. Mohar. The Laplacian spectrum of graphs. In Y. Alavi, G. Chartrand, O. R. Oellermann, A. J. Schwenk (Eds.), Graph Theory, Combinatorics, and Applications, Wiley, volume 2, pages 871–898, 1991.
- [43] D. Mumford and J. Shah. Optimal approximations by piece-wise smooth functions and assosiated variational problems. Comm. Pure and Appl. Math., 42:577–685, 1989.
- [44] B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis. Diffusion maps, spectral clustering, and the reaction coordinates of dynamical systems. Report, Math. Dept. Yale, Nov. 2004. To appear in Journal of Applied and Computational Harmonic Analysis.
- [45] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation based image restoration. SIAM Journal on Multiscale Modeling and Simulation, 4:460–489, 2005.
- [46] S. Osher and N. Paragios (Eds.). Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer-Verlag, 2003.
- [47] S. Osher and R. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Springer, 2002.
- [48] P. Perona and W.T. Freeman. A factorization approach to grouping. In ECCV, pages 655–670, 1998.
- [49] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. PAMI, 12(7):629–639, 1990.
- [50] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309–314, 2004.
- [51] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [52] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

- [53] S.M. Smith and J.M. Brady. SUSAN a new approach to low level image processing. International Journal of Computer Vision, 23(1):45–78, 1997.
- [54] N. Sochen, R. Kimmel, and R. Malladi. A general framework for low level vision. IEEE Transactions on Image Processing, 7:310–318, 1998.
- [55] A.D. Szlam. 2006. Private communication.
- [56] A.D. Szlam. Non-stationary analysis of datasets and applications, 2006. PhD. Thesis, Yale.
- [57] A.D. Szlam, M. Maggioni, Jr. J.C. Bremer, and R.R. Coifman. Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions. In SPIE, 2005.
- [58] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In ICCV '98, pages 839–846, 1998.
- [59] J. Wang, P. Bhat, R.A. Colburn, M. Agrawala, and M.F. Cohen. Interactive video cutout. ACM Trans. Graph., 24(3):585–594, 2005.
- [60] Y. Weiss. Segmentation using eigenvectors: A unifying view. In International Conference on Computer Vision, pages 975–982, 1999.
- [61] J. Xu and S. Osher. Iterative regularization and nonlinear inverse scale space applied to wavelet based denoising, 2006. UCLA CAM Report 06-11.
- [62] L.P. Yaroslavsky. Digital Picture Processing, an Introduction. Springer-Verlag, Berlin, 1985.
- [63] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. IEEE Transactions on Image Processing, 15(5):1120–1129, 2006.
- [64] S.X. Yu and J. Shi. Segmentation given partial grouping constraints. IEEE Trans. Pattern Anal. Mach. Intell., 26(2):173–183, 2004.
- [65] S.C. Zhu and A.L. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9):884– 900, 1996.