SIAM J. IMAGING SCIENCES
Vol. 2, No. 1, pp. 118–139

Diffusion Interpretation of Nonlocal Neighborhood Filters for Signal Denoising*

Amit Singer[†], Yoel Shkolnisky[‡], and Boaz Nadler[§]

Abstract. Nonlocal neighborhood filters are modern and powerful techniques for image and signal denoising. In this paper, we give a probabilistic interpretation and analysis of the method viewed as a random walk on the patch space. We show that the method is intimately connected to the characteristics of diffusion processes, their escape times over potential barriers, and their spectral decomposition. In particular, the eigenstructure of the diffusion operator leads to novel insights on the performance and limitations of the denoising method, as well as a proposal for an improved filtering algorithm.

Key words. denoising, neighborhood filters, nonlocal means, Fokker–Planck equation, first passage time.

AMS subject classifications. 62H35, 82C31

DOI. 10.1137/070712146

1. Introduction.

Denoising of signals and images is a fundamental task in signal processing. Early approaches, such as Gaussian (Gabor) filters and anisotropic diffusion [22], denoise the value of a signal $y(x_1)$ at a point x_1 based only on the observed values $y(x_2)$ at neighboring points x_2 spatially close to x_1. To overcome the obvious shortcomings of this locality property, many authors proposed various global and multiscale denoising approaches. Among others, we mention minimization of global energy functionals such as the total-variation functional [23] and Fourier and wavelet denoising methods [12]. The recent paper [10] provides a review of many such methods.

Although quite sophisticated, these methods typically do not take into account an important feature of many signals and images, that of *repetitive behavior*, e.g., the fact that small patterns of the original noise-free signal may appear a large number of times at different spatial locations. For one-dimensional (1-D) signals this property holds for every periodic or nearly periodic function (such as repetitive neuronal spikes, heart beats, etc.) and for many telegraph type processes. Similarly, identical patches typically appear at many different and possibly spatially distant locations in two-dimensional (2-D) images. The fact that the same noise-free pattern appears multiple instances can obviously be utilized for improved denoising. Rather than averaging the value of a noisy signal at a point x based only on its few neighbor

[*] Received by the editors January 2, 2008; accepted for publication (in revised form) October 20, 2008; published electronically January 30, 2009. The work of the first and second authors was supported in part by DARPA DSO (Dr. Cindy Daniell PM) under AFOSR contract FA9550-07-C-0024 (Dr. Fariba Fahroo PM).
http://www.siam.org/journals/siims/2-1/71214.html

[†] Department of Mathematics and PACM, Princeton University, Fine Hall, Washington Road, Princeton, NJ 08544-1000 (amits@math.princeton.edu). The first author's research was supported by the Yale University-Weizmann Institute Joint Research Fund.

[‡] Department of Mathematics, Program in Applied Mathematics, Yale University, 10 Hillhouse Ave., P.O. Box 208283, New Haven, CT 06520-8283 (yoel.shkolnisky@yale.edu).

[§] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, 76100, Israel (boaz.nadler@weizmann.ac.il). This author's research was supported by the Israeli Science Foundation (grant 432/06) and by the Lord Sieff of Brimpton memorial fund.

values, one can identify other locations in the signal where a similar pattern appears and average all of these instances. This observation naturally leads to the development of various *nonlocal* (NL) denoising methods [30, 8, 10, 18, 14, 28, 7, 27, 2, 3]. An understanding of these methods is the focus of this paper.

The concept of NL averaging, via an NL neighborhood filter was introduced by Yaroslavsky [30]. For a continuous signal $y(x)$, the neighborhood filter is defined as

$$NF_h y(x_1) = \frac{1}{D(x_1)} \int K\left(\frac{y(x_1) - y(x_2)}{h}\right) y(x_2) dx_2, \tag{1.1}$$

where $K(z)$ is any smoothly decaying integrable kernel, such as the Gaussian, $K(z) = e^{-z^2/2}$, and $D(x_1) = \int K(\frac{y(x_1)-y(x_2)}{h})dx_2$ is a normalization factor. Note that one application of the neighborhood filter (1.1) averages the value of $y(x_1)$ according to points x_2 with similar y-values. These can be located far from the original point x_1.

Various authors [30, 26, 29, 4, 5] combined NL neighborhood filters with spatially local kernels, leading to methods which denoise the signal at x_1 by taking averages of values $y(x_2)$ for which both $y(x_2)$ is close to $y(x_1)$ and x_2 is close to x_1. The latter is also known as bilateral filtering. Recently, this idea was further extended to an NL-means neighborhood filter, where the similarity between locations x_1 and x_2 is measured not by their single y-values but rather by some local means [8, 10, 18, 27, 28, 2, 3]. For example, [10] proposed the following operator:

$$NL_h y(x_1) = \frac{1}{D(x_1)} \int K\left(\frac{G_a * [y(x_1 + .) - y(x_2 + .)]^2}{h}\right) y(x_2) dx_2, \tag{1.2}$$

where $G_a * [y(x_1 + .) - y(x_2 + .)]^2$ is the convolution of the squared difference of the shifted signals $y(x_1 + .) - y(x_2 + .)$ with a Gaussian kernel,

$$G_a * [y(x_1 + .) - y(x_2 + .)]^2 = \int G_a(t) [y(x_1 + t) - y(x_2 + t)]^2 dt. \tag{1.3}$$

In other words, the value $y(x_2)$ is used to denoise $y(x_1)$ if the local pattern near $y(x_2)$ is similar to the local pattern near $y(x_1)$. In [2, 3] these equations were derived via minimization of a joint entropy principle with $D(x_1)$ interpreted as a nonparametric density estimate. Alternative derivations were given in [17, 14, 7] using variational principles. The diffusion character of neighborhood filters as a local algorithm was analyzed in [9, 4].

Even though the algorithm is extremely simple, essentially described by (1.2) and (1.3), it is surprisingly superior to other methods, as demonstrated in [10, 7, 14, 3] through extensive experimentation. In the context of images, NL neighborhood filter methods are able to handle texture, edges, and high frequency signals all at once. When applied to images, these methods are able to separate the majority of the signal from the noise with the resulting residuals typically looking like pure noise and showing almost no texture or other structure [10, 8].

The purpose of this paper is to provide a probabilistic interpretation to these NL methods. For simplicity, we present our analysis for 1-D signals, although it can be easily extended to the case of 2-D images. The key observation in our analysis is that whereas standard Gabor-type filtering methods can be viewed as a diffusion on the x-axis, single-pixel based neighborhood filters perform a diffusion on the y-axis. Similarly, the NL-means algorithm that averages patches can be viewed as a random walk in the patch space [27]. This simple observation sheds light on many of the properties of these algorithms. Our main result is a probabilistic explanation of the behavior of both the neighborhood filter and the NL-means algorithm, including their rate of convergence, their blurring properties, and the advantages (but also limitations) of taking patches of values rather than a single value for computing a similarity metric between pixels. Furthermore, the relation between the averaging process and the eigenstructure of the related diffusion operator leads to a proposition of an improved filtering algorithm.

The paper is organized as follows. In section 2 we present neighborhood filters that are based on single-pixel values and their probabilistic interpretation as diffusion processes on the y-values of the signal. The denoising performed by this algorithm on constant functions contaminated with white noise and the resulting rate of convergence are considered in section 3. We show that the rate of convergence is intimately connected to a diffusion in a potential well whose parameters depend on the noise variance and on the width of the kernel. An analysis of the algorithm on more complicated stepwise functions is described in section 4. In denoising such functions, the key quantity that comes into play is the mean exit time of a diffusion process in a multiwell potential, from one y-value well to another. The advantages and limitations of taking patches rather than single y-values are considered in section 5. In section 6 we consider the neighborhood filter denoising method as a low pass filter and present possible improvements and modifications to the basic scheme that are supported by experimental results. Finally, section 7 is a summary.

2. Diffusion in the y-space.

To denoise a signal, the NL-means algorithm typically compares local patches of the signal (or image) values. We start by analyzing the neighborhood filter which compares the smallest possible neighborhood size, i.e., a single pixel. In later sections we will carry over the analysis to the more realistic case of larger neighborhoods containing several pixels.

Consider a continuous signal $y(x)$ sampled at N points x_i $(i = 1, \dots, N)$. We wish to denoise the sequence $y(x_i)$, $i = 1, \dots, N$. One iteration of the neighborhood filter averages nearby y-values to obtain a denoised version of the signal y_d given by

$$y_d(x_i) = \frac{1}{D(x_i)} \sum_{j=1}^{N} K_\varepsilon(y(x_i), y(x_j))\, y(x_j), \tag{2.1}$$

where K_ε is a positive definite kernel, e.g., the Gaussian

$$K_\varepsilon(y(x_i), y(x_j)) = \exp\left\{-(y(x_i) - y(x_j))^2/2\varepsilon\right\},$$

$D(x_i) = \sum_{j=1}^{N} K_\varepsilon(y(x_i), y(x_j))$ is a normalization factor, and $\sqrt{\varepsilon}$ is the width of the kernel. The kernel K_ε measures the similarity of its input. It is close to one whenever $y(x_1) \approx y(x_2)$ and is close to zero when $|y(x_1) - y(x_2)| \gg \sqrt{\varepsilon}$. Thus, points with y-values less than $\sqrt{\varepsilon}$ apart are averaged via (2.1), leading to a suppression of noise. Note that points x_1, x_2 with similar y-values are not necessarily spatially nearby. This can happen in several cases: the function may be discontinuous so nearby points have different y-values or two distant points may have the same y-value.

The denoising iteration (2.1) can also be written as

$$y_1(x_i) = \frac{1}{D_0(x_i)} \sum_{j=1}^{N} K_\varepsilon(y_0(x_i), y_0(x_j))\, y_0(x_j), \tag{2.2}$$

where $y_0 = y$ and $D_0(x_i) = \sum_{j=1}^{N} K_\varepsilon(y_0(x_i), y_0(x_j))$. In many cases a single denoising step is not sufficient and a few iterations are needed. There are three ways of iterating (2.2). The first is to define the denoised signal y_n at stage n as

$$y_n(x_i) = \frac{1}{D_0(x_i)} \sum_{j=1}^{N} K_\varepsilon(y_0(x_i), y_0(x_j))\, y_{n-1}(x_j). \tag{2.3}$$

In this scheme, the kernel matrix and the normalization factors remain *fixed* during the iteration process and depend only on the initial signal y_0. The second way is to update the kernel matrix and normalization based on the most recent denoised signal leading to

$$y_n(x_i) = \frac{1}{D_{n-1}(x_i)} \sum_{j=1}^{N} K_\varepsilon(y_{n-1}(x_i), y_{n-1}(x_j))\, y_{n-1}(x_j), \tag{2.4}$$

where $D_{n-1}(x_i) = \sum_{j=1}^{N} K_\varepsilon(y_{n-1}(x_i), y_{n-1}(x_j))$. The third way [7, eq. (11)] is to update only the kernel and its normalization while keeping the signal fixed:

$$y_n(x_i) = \frac{1}{D_{n-1}(x_i)} \sum_{j=1}^{N} K_\varepsilon(y_{n-1}(x_i), y_{n-1}(x_j))\, y_0(x_j). \tag{2.5}$$

All methods have been considered and compared in the literature [7], and each has its own advantages and disadvantages. Iterative algorithms of both types (2.3) and (2.4) have been studied in [11] as part of the mean shift algorithm for clustering and mode seeking, rather than for denoising. In the context of the mean shift algorithm, updating the kernel matrix with each iteration (2.4) is referred to as the blurring process. In this paper we focus on the nonblurring (or stationary) procedure (2.3) with a fixed kernel matrix and analyze the properties of the resulting denoising algorithm. Later on, we comment on the relation between the nonstationary blurring process (2.4) and the stationary process (2.3) for the special case of Gaussian white noise.

The denoising iteration (2.1) can be written as the matrix-vector multiplication

$$\boldsymbol{y}_d = \mathbf{D}^{-1}\mathbf{W}\boldsymbol{y}, \tag{2.6}$$

where $\mathbf{W}$ is an N by N matrix given by

$$W_{ij} = K_\varepsilon(y(x_i), y(x_j)), \tag{2.7}$$

$\mathbf{D}$ is a diagonal matrix with

$$D_{ii} = \sum_{j=1}^{N} W_{ij},$$

and $\boldsymbol{y} = (y(x_1), \dots, y(x_N))$ is the signal. Introducing the averaging operator

$$\mathbf{A} = \mathbf{D}^{-1}\mathbf{W}, \tag{2.8}$$

the denoising iteration (2.6) is equivalent to

$$\boldsymbol{y}_d = \mathbf{A}\boldsymbol{y}. \tag{2.9}$$

The matrix $\mathbf{A}$ is a row-stochastic matrix, corresponding to a random walk on the values $y(x_j)$. We emphasize that the random walk is determined only by the y-values, while the x-values have no role. The probability of jumping from $y(x_i)$ to $y(x_j)$ depends only on the difference $y(x_i) - y(x_j)$. The matrix $\mathbf{A}$ is the transition probability matrix of the Markovian process Y_k:

$$A_{ij} = \Pr\{Y_{k+1} = y(x_j) \,|\, Y_k = y(x_i)\}. \tag{2.10}$$

The values $y(x_j)$ can be viewed as nodes of a weighted graph, where A_{ij} are the probabilities of jumping from node i to node j.

The probabilistic interpretation of a single denoising step now becomes clear:

$$\begin{aligned}(\mathbf{A}\boldsymbol{y})(x_i) &= \sum_{j=1}^{N} A_{ij} y(x_j) = \sum_{j=1}^{N} \Pr\{Y_{k+1} = y(x_j) \,|\, Y_k = y(x_i)\}\, y(x_j) \\ &= \mathbb{E}[Y_{k+1} \,|\, Y_k = y(x_i)].\end{aligned} \tag{2.11}$$

In other words, applying the matrix $\mathbf{A}$ to the signal $\boldsymbol{y}$ gives a vector whose ith entry is the expected value of the random walker starting at the ith node $y(x_i)$ after a single step. Similarly, performing k successive denoising iterations corresponds to the expected value after k steps.

The matrix $\mathbf{A}$ is conjugate to the positive definite matrix $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ via

$$\mathbf{A} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{1/2}. \tag{2.12}$$

This implies that $\mathbf{A}$ has a complete set of right eigenvectors $\{\boldsymbol{\psi}_j\}_{j=0}^{N-1}$ and positive eigenvalues

$$1 = \lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{N-1} > 0.$$

The largest eigenvalue is $\lambda_0 = 1$, corresponding to the trivial all-ones right eigenvector ($\mathbf{A}\mathbf{1} = \mathbf{1}$). We expand the signal vector $\boldsymbol{y}$ in the eigenbasis

$$y(x_i) = \sum_{j=0}^{N-1} b_j \psi_j(x_i), \tag{2.13}$$

where $b_j = \langle \boldsymbol{\phi}_j, \boldsymbol{y} \rangle$ and $\{\boldsymbol{\phi}_j\}_{j=0}^{N-1}$ are the left eigenvectors of $\mathbf{A}$.

Applying the matrix $\mathbf{A}$ to $\boldsymbol{y}$ results in

$$(\mathbf{A}\boldsymbol{y})(x_i) = \sum_{j=0}^{N-1} \lambda_j b_j \psi_j(x_i). \tag{2.14}$$

In practice, signals are denoised by repeated application of the denoising operator $\mathbf{A}$. Again, we emphasize that the denoising iterations are to be understood in the stationary "nonblurring" sense of (2.3). Applying the operator k times gives

$$(\mathbf{A}^k\boldsymbol{y})(x_i) = \sum_{j=0}^{N-1} \lambda_j^k b_j \psi_j(x_i), \tag{2.15}$$

which effectively discards the high modes with small eigenvalues.

In the limit of an infinite number of denoising iterations, the resulting denoised signal is a constant function with value b_0 which is the coefficient of the first eigenvector $\mathbf{1}$. This is also the weighted average of the original observed values $y(x_j)$ with respect to the steady state distribution $\boldsymbol{\phi}_0$. Thus, in order to denoise the signal, we need to apply a finite number of iterations, large enough to attenuate the noise, but not so large as to blur the signal. As mentioned in the introduction, neighborhood filters and NL-means can be viewed as gradient descent algorithms that denoise the signal by minimizing its entropy [3], which increases when noise is added to it. Although the process of minimizing the entropy affects both the clean signal and the added noise component, it was noted in [3] that the first few iterations of the NL-means algorithm reduce the noise while leaving the clean signal part almost unchanged. This behavior of the NL-means algorithm is the key to its success. In the next section we discuss this issue in detail.

3. Level sets, Gaussian noise, and the Hermite polynomials.

We start from the simplest possible example of neighborhood filter denoising applied to a signal $y(x)$ consisting of a constant function $v(x) = v_0$ corrupted by additive Gaussian white noise $n(x)$:

$$y(x) = v_0 + n(x). \tag{3.1}$$

In this case, the matrix $\mathbf{W}$ depends only on the Gaussian noise

$$W_{ij} = \exp\{-(n(x_i) - n(x_j))^2/2\varepsilon\}. \tag{3.2}$$

In the limit of a large number of samples $N \to \infty$ and kernel width $\varepsilon \to 0$ the averaging operator converges to the backward Fokker–Planck operator $\mathcal{L}$ (see [25, 20, 21, 6, 16] for more details):

$$\sum_{j=1}^{N} A_{ij} f(y(x_j)) = f(y(x_i)) + \frac{\varepsilon}{2}\mathcal{L}f(y(x_i)) + O(\varepsilon^2) \tag{3.3}$$

for any smooth function f defined on the data points $y(x_j)$. When using single pixels the resulting Fokker–Planck operator is a second order differential operator given by

$$\mathcal{L}f(y) = f''(y) - U'(y)f'(y), \tag{3.4}$$

where $U(y) = -2\log p(y)$ is the potential derived from $p(y)$, which is the density of the y-values. In our case $p(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-(y-v_0)^2/2\sigma^2\}$, where σ^2 is the variance of the white noise. Up to an additive constant, this results in a parabolic potential well given by $U(y) = (y-v_0)^2/\sigma^2$.

It can also be shown that the eigenvectors of $\mathbf{A}$ are discrete approximations of the eigenfunctions of the continuous Fokker–Planck operator. For the parabolic potential, the eigenfunctions $\psi_j(y)$ satisfy the second order differential equation

$$\psi_j''(y) - \frac{2(y-v_0)}{\sigma^2}\psi_j'(y) = -\mu_j\psi_j(y), \tag{3.5}$$

where μ_j are the corresponding eigenvalues. The eigenfunctions are the well-known Hermite polynomials $\psi_j(y) = H_j((y-v_0)/\sigma)$ with $\mu_j = 2j/\sigma^2$ [1]. The first few Hermite polynomials are $H_0(y) = 1$, $H_1(y) = y$, and $H_2(y) = y^2 - 1$.

We are now ready to understand the effect of applying the denoising operator $\mathbf{A}$ on the signal $\boldsymbol{y}$. For the moment, we assume that $N \gg 1$ and ε is sufficiently small so that the continuous approximation of the discrete operator $\mathbf{A}$ holds. Then, in light of (2.13) and the special form of the eigenfunctions, the expansion of $\boldsymbol{y}$ contains only the first two terms, namely, H_0 and H_1:

$$\boldsymbol{y} = v_0\boldsymbol{\psi}_0 + b_1\boldsymbol{\psi}_1 = v_0\mathbf{1} + (\boldsymbol{y} - v_0\mathbf{1}). \tag{3.6}$$

It follows from (2.14) that the y-values obtained after k denoising iterations are given by

$$\mathbf{A}^k\boldsymbol{y} = v_0\mathbf{1} + \lambda_1^k(\boldsymbol{y} - v_0\mathbf{1}), \tag{3.7}$$

where

$$\lambda_1 \approx \exp\{-\mu_1\varepsilon/2\} \approx 1 - \varepsilon/\sigma^2 \tag{3.8}$$

is the asymptotic noise reduction rate. That is, each iteration of the denoising operator shrinks the noisy values of $\boldsymbol{y}$ towards their mean v_0 at a constant rate λ_1. In particular, the noise remains Gaussian and white, only its standard deviation is decreased by the factor $\lambda_1 < 1$. A consequence of this constant shrinkage behavior is that in the Gaussian white noise case, constructing at each iteration a new kernel matrix (2.4) is equivalent to changing the kernel parameter ε in every iteration of the stationary procedure (2.3). Thus, constructing a new matrix at every iteration corresponds to changing the parameter ε of the previous iteration to ε/λ_1^2 in the current iteration.

Even when ε is not small, similar results hold in the case of a Gaussian white noise. When ε is not small, the discrete matrix $\mathbf{A}$ can be approximated in the limit $N \gg 1$ by an integral operator instead of the differential Fokker–Planck operator. It turns out that the kth eigenfunction of the integral operator is a polynomial of degree k. In particular, the second eigenfunction is linear as is the second Hermite polynomial; see [21, section 5.1], where it is also shown that a uniform approximation for the noise reduction rate that holds for all values of ε is given by

$$\lambda_1 \approx \frac{\sigma^2}{\sigma^2 + \varepsilon}. \tag{3.9}$$

Clearly, the approximations (3.9) and (3.8) agree in the limit of small ε.

The noise reduction rate λ_1 depends on the kernel parameter ε and on the variance of the noise σ^2, but is independent of the number of samples N. A similar result regarding the variance reduction of the noisy pixels in the Gaussian white noise case was obtained in [8, Theorem 5.3] using different probabilistic considerations.

The diffusion interpretation of the neighborhood filter is illustrated in Figures 1(a)–1(g). Figure 1(a) shows a white noise signal consisting of $N = 2000$ independently identically distributed samples from a standard Gaussian distribution ((3.1) with $v_0 = 0$ and $\sigma = 1$). The matrix $\mathbf{W}$ is formed using (3.2) and is normalized according to (2.8) to give the denoising operator $\mathbf{A}$. Figure 1(b) shows the denoised signal $\mathbf{A}\boldsymbol{y}$. The only significant difference between Figures 1(a) and 1(b) is the vertical scaling: the denoised signal takes on smaller values. This effect is clearly observed in Figure 1(c) that shows the denoised signal after 10 iterations. Figure 1(e) is a scatter plot of $(\mathbf{A}\boldsymbol{y})(x_i)$ against $y(x_i)$, from which it is evident that $(\mathbf{A}\boldsymbol{y})(x_i)$ is proportional to $y(x_i)$. Using a least squares linear fit, we find that $(\mathbf{A}\boldsymbol{y})(x) \approx 0.826y(x)$. We computed the first few eigenvectors and eigenvalues of $\mathbf{A}$. In particular, a scatter plot of $\psi_1(x_i)$ against $y(x_i)$ is given in Figure 1(f). The linear dependence of $\psi_1(x_i)$ in $y(x_i)$ is clear, in agreement with (3.6) (H_1, the first order Hermite polynomial, is linear). The corresponding computed eigenvalue $\lambda_1 = 0.824$ explains the slope in Figure 1(e). Moreover, the computed eigenvalue agrees with the approximation (3.9) $\lambda_1 \approx \frac{\sigma^2}{\sigma^2+\varepsilon} = \frac{1}{1.2} = 0.833$. We estimated the density of the 2000 y-values using kernel smoothing with 20 nearest neighbors. The empirical density $p(y)$, which approximates a standard Gaussian, is shown in Figure 1(d). The derived empirical potential $U(y) = -2\log p(y)$ is given in Figure 1(g). This potential explains why neighborhood filter denoising of the discrete white noise signal can be approximated by a continuum diffusion in a parabolic potential well.

Our analysis applies to models of noise other than the Gaussian. For noise with probability density $p(x)$ there corresponds a potential $U = -2\log p$. The asymptotic rate of convergence now depends on the smallest nontrivial eigenvalue of the Fokker–Planck operator (3.4) (with Neumann boundary conditions if p is compactly supported). For example, if the noise is uniformly distributed in the interval $[-a, a]$, then the eigenfunctions are $\psi_j(y) = \cos(j\pi y/a)$ $(j = 0, 1, 2, \dots)$ and $\mu_j = \frac{\pi^2 j^2}{a^2}$. In this case the effect of the neighborhood filter will not be a constant shrinkage of all noise values, because $\boldsymbol{y}$ is no longer an eigenfunction of $\mathbf{A}$.

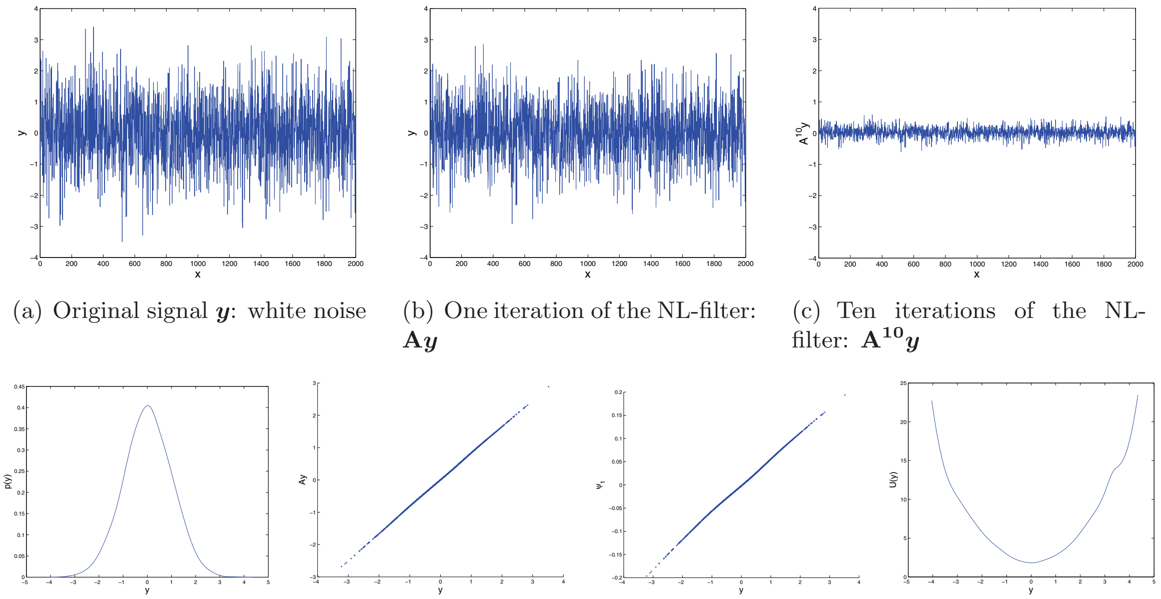
4. Step functions, edges, and the double well escape problem.

We now consider the neighborhood filter applied to the signal $y(x) = v(x)+n(x)$, where $v(x)$ is a piecewise constant step function which obtains one of two values $v(x) = v_0$ or $v(x) = v_1$ $(v_0 < v_1)$. Such functions are, for example, the output of a random telegraph process with only two states which jumps from one state to the other at random times, and therefore appear in many signal processing and communication applications.

In the absence of noise, the density of y-values of this function is the sum of two δ-functions. The additive noise leads to a density $p(y)$ which is a weighted sum of two Gaussians

$$p(y) = \frac{w_1}{\sqrt{2\pi\sigma^2}} \exp\{-(y-v_0)^2/2\sigma^2\} + \frac{w_2}{\sqrt{2\pi\sigma^2}} \exp\{-(y-v_1)^2/2\sigma^2\}, \tag{4.1}$$

where w_1 and w_2 are the frequencies of occurrence of the two states, satisfying $w_1 + w_2 = 1$.



(a) Original signal $\boldsymbol{y}$: white noise

(b) One iteration of the NL-filter: $\mathbf{A}\boldsymbol{y}$

(c) Ten iterations of the NL-filter: $\mathbf{A}^{\mathbf{10}}\boldsymbol{y}$

(d) Density: $p(y)$ vs. y

(e) $\mathbf{A}\boldsymbol{y}$ vs. $\boldsymbol{y}$

(f) $\boldsymbol{\psi}_1$ vs. $\boldsymbol{y}$

(g) Potential: $U(y)$ vs. y

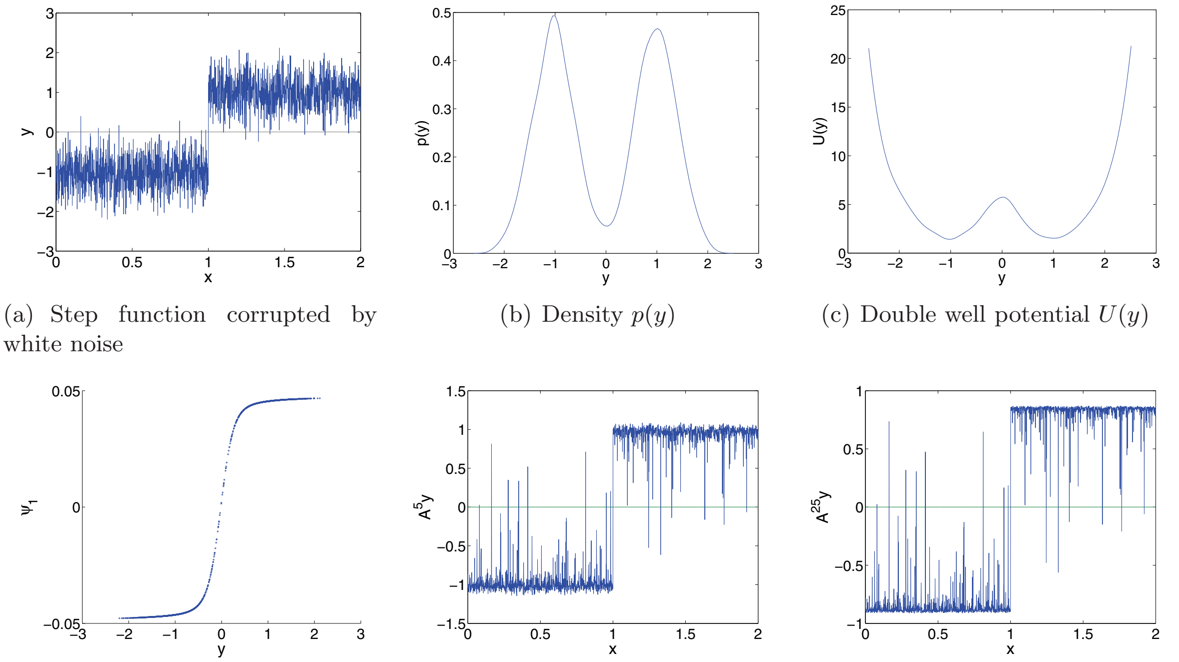
Figure 1. *The NL-filter applied to white noise. Parameters:* $N = 2000$, $\sigma = 1$, $\varepsilon = 0.2$.

Assuming that $\sigma \ll v_1 - v_0$, the corresponding potential $U(y) = -2\log(p)$ has the form of a double well potential.

An example of a piecewise (random) function $y(x)$ corrupted by noise, together with the corresponding density and potential of y-values, is shown in Figures 2(a), 2(b), 2(c), respectively. The eigenvector $\boldsymbol{\psi_1}$ is plotted in Figure 2(d) and is seen to be approximately a step function with a sharp transition at $y = 0$. Thus, the NL-filter takes negative y-values to (approximately) -1 and positive y-values to (approximately) 1 as wanted.

In the limit $N \to \infty, \varepsilon \to 0$, the discrete random walk associated with the row-stochastic matrix $\mathbf{A}$ converges to a continuous-time diffusion process. Equation (2.11) relates the expected value of the discrete random walk with the denoising operator $\mathbf{A}$, while (3.3) shows that this operator is approximated by the backward Fokker–Planck operator which is the generator of the continuous-time limit process. A single denoising iteration corresponds to a time step $\Delta t = \varepsilon$ of the continuous process. The denoising scheme thus depends on the characteristics of a diffusion process in a double well potential, a subject which has been extensively studied in the literature [24, 13, 19]. We now briefly describe the main features of this diffusion process that are of interest to us. Consider a diffusion process that starts at a value $y = v_0 + \xi$ in the left well. Of course, the purpose of the neighborhood filter is to denoise the signal, that is, to approximate all values in the left well by v_0 and all values in the right well by v_1.

The diffusion process has a few characteristic times. The first is τ_R, the relaxation, or equilibration time, inside the left (or right) well, conditioned on the process not exiting this well. To properly denoise all values in a given well to their mean, it is therefore necessary to apply approximately $k_R = \tau_R/\varepsilon$ denoising steps, because the time step of each iteration is $\Delta t = \varepsilon$ as discussed above. The characteristic relaxation time inside the well centered at the



(a) Step function corrupted by white noise

(b) Density $p(y)$

(c) Double well potential $U(y)$

(d) ψ_1 against y

(e) After 5 iterations of the neighborhood filter

(f) After 25 iterations of the neighborhood filter

Figure 2. *The neighborhood filter applied to a step function corrupted by white noise. The step function is given by $v(x) = v_0 = -1$ for $0 < x < 1$ and $v(x) = v_1 = 1$ for $1 < x < 2$. Parameters: $N = 2000$, $\sigma = 0.4$, $\varepsilon = 0.1$.*

y-value v_0 depends to first order only on the curvature at the bottom of the well and is given by

$$\tau_R(v_0) = \frac{1}{U''(v_0)}. \tag{4.2}$$

For the density p given by (4.1), this gives $\tau_R = \sigma^2$, and hence the resulting number of iterations is $k_R = \frac{\tau_R}{\varepsilon} \approx \frac{1}{1-\lambda_1}$, where λ_1 is given by (3.8). At this number of denoising iterations we obtain that the noise has been shrunk by $\lambda_1^{k_R} \approx \exp\{\frac{\log\lambda_1}{1-\lambda_1}\} \approx e^{-1}$, for λ_1 close to 1.

The second characteristic time τ_{exit} is the mean first passage time (MFPT) to exit this well by crossing the barrier separating the wells. For a double well potential, the MFTP from v_0 to v_1 is approximately given by Kramers' law

$$\tau_{exit}(v_0 \to v_1) = \frac{2\pi}{\sqrt{|U''(v_0)U''(v_m)|}} e^{U(v_m)-U(v_0)}, \tag{4.3}$$

where v_m is the location of the peak of the potential U, between the two wells $v_0 < v_m < v_1$. Similarly, after an order of $k_{exit} = \tau_{exit}/\varepsilon$ denoising iterations, the resulting y-values become averages of v_0 and v_1, leading to a blurring of the resulting signal.

High quality denoising by the neighborhood filter is possible only if the two different level sets are well separated. This is translated into the mathematical condition $\tau_R \ll \tau_{exit}$ or, equivalently, $k_{exit} \gg k_R$. This analysis also provides a guideline as to the number of denoising steps needed to achieve satisfactory results. The number of steps k should be larger than k_R but significantly smaller than k_{exit}, as otherwise the resulting values are weighted averages of the two level sets. This is in full agreement with our intuition and numerical results: too few iterations do not sufficiently smooth out the noise, whereas too many iterations have the undesired property of averaging distinct level sets.

An inherent limitation of using a single-pixel neighborhood filter occurs in the presence of significant noise. Consider, for example, a step function

$$v(x) = \begin{cases} -1, & 0 < x < 1, \\ 1, & 1 < x < 2. \end{cases} \tag{4.4}$$

Regardless of the spatial location of x between 0 and 1, noisy y-values above 0 will be assigned to the right well and thus averaged to 1, whereas y-values below 0 will be assigned to the left well and averaged to -1. This causes misidentifications and incorrect denoising. For example, five iterations of the neighborhood filter to the noisy signal shown in Figure 2(a) result in the signal shown in Figure 2(e). Although most of the noise was filtered out, misidentifications are evident. Also, the level sets after 25 iterations (Figure 2(f)) are $y \approx \pm 0.8$ instead of $y = \pm 1$, due to the global averaging effect (the potential barrier in Figure 2(c) is not too high).

The probability of misidentifying a single observation is

$$\begin{aligned} \Pr\{y(x_i) > 0 \,|\, v(x_i) = -1\} &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^\infty \exp\{-(y+1)^2/2\sigma^2\}\, dy \\ &= \frac{1}{2} \operatorname{erfc}\left(\frac{1}{\sqrt{2}\,\sigma}\right). \end{aligned} \tag{4.5}$$

In our example of $\sigma = 0.4$ we expect 0.6% of the samples to be misidentified (or 6 out of 1000 in accordance with Figure 2(f)).

There are at least two different methods for overcoming misidentifications due to large noise. The first is spatial adaptations as discussed in [3, 5], while the second is increasing the patch size. In the next section we show how using patches of size two or larger significantly reduces the misidentification probability.

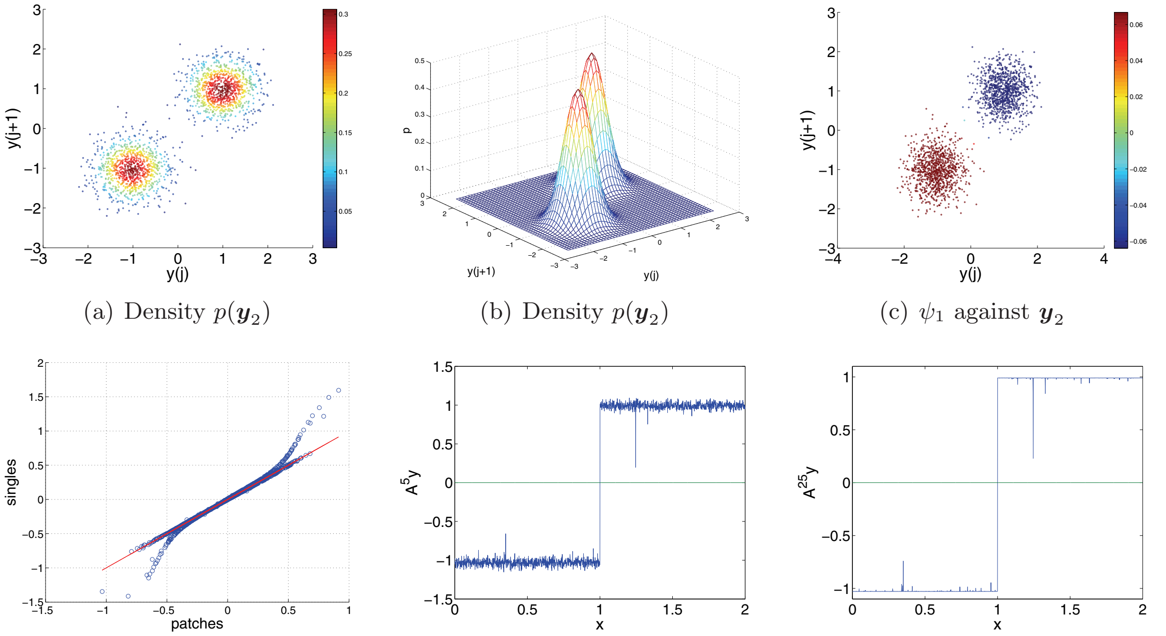
5. From pixels to patches: Diffusion in higher dimensions and clustering.

As seen in the previous section, denoising based on the similarity between single y-values may be insufficient, leading to misidentifications and blurring effects. Therefore, in practical applications a more advanced version based on the NL-filter is used. Instead of averaging points with similar y-values, points are averaged only if their entire local y-value patch is similar; see [10, 18, 28]. For simplicity, we analyze patches of size two,

$$\boldsymbol{y}_2(x_j) = (y(x_j), y(x_{j+1})).$$

The weights are only a function of the Euclidean distance between patches, for example,

$$W_{ij} = \exp\{-\|\boldsymbol{y}_2(x_i) - \boldsymbol{y}_2(x_j)\|^2/2\varepsilon\}. \tag{5.1}$$



(a) Density $p(\boldsymbol{y}_2)$

(b) Density $p(\boldsymbol{y}_2)$

(c) ψ_1 against $\boldsymbol{y}_2$

(d) Comparison of single vs. patch denoising

(e) After five iterations of the NL-filter

(f) After 25 iterations of the NL-filter

Figure 3. *The NL-filter with two-pixel patches applied to a step function corrupted by white noise, on the same data as in Figure 2(a). Parameters:* $N = 2000$, $\sigma = 0.4$, $\varepsilon = 0.1$.

The averaging operator $\mathbf{A}$ is defined as in the single-pixel case, $\mathbf{A} = \mathbf{D}^{-1}\mathbf{W}$. We demonstrate the advantages of using two-pixel patches over the single-pixel method. Keeping the diffusion interpretation in mind, using patches of size two corresponds to a 2-D diffusion process on the set of values $\boldsymbol{y}_2(x_i) = (y(x_i), y(x_{i+1}))$. That is, as $N \to \infty$, for ε sufficiently small, the eigenvectors of $\mathbf{A}$ can be approximated by those of a 2-D Fokker–Planck operator, analogous to the 1-D operator of (3.4).

Consider, for example, the step function defined in (4.4). In the absence of noise the planar density of patches of size two $p(\boldsymbol{y}_2)$ is a sum of two δ-functions concentrated at $(-1,-1)$ and $(1,1)$. Similar to the 1-D case, corrupting the signal by noise changes $p(\boldsymbol{y}_2)$ into a mixture of 2-D Gaussians with identical covariance matrices equal to $\sigma^2\mathbf{I}$ (where $\mathbf{I}$ is the 2 by 2 identity matrix):

$$(5.2) \qquad p(\boldsymbol{y}_2) = \frac{1}{4\pi\sigma^2}\exp\left\{\frac{-\|\boldsymbol{y}_2 + (1,1)\|^2}{2\sigma^2}\right\} + \frac{1}{4\pi\sigma^2}\exp\left\{\frac{-\|\boldsymbol{y}_2 - (1,1)\|^2}{2\sigma^2}\right\}.$$

Extracting two-pixel patches from the noisy signal in Figure 2(a) results in two Gaussian clouds in the plane that are shown in Figure 3(a), where points are colored by their estimated empirical density. A surface plot of the analytical density (see (5.2)) is given in Figure 3(b).

There are two differences between the 1-D neighborhood filter and its 2-D NL-means version. First, observe that the distance between the Gaussian centers $(-1,-1)$ and $(1,1)$ is $2\sqrt{2}$, which is a factor of $\sqrt{2}$ larger than the distance of the 1-D Gaussians centered at

Table 1

The first few eigenvalues of $\mathbf{A}$ *constructed from single pixels in the step function case. Parameters:* $N = 2000$, $\sigma = 0.4$, $\varepsilon = 0.1$.

i	0	1	2	3	4	5
$\lambda_i^{(1)}$	1.0	0.993	0.631	0.598	0.412	0.344

Table 2

The first few eigenvalues of $\mathbf{A}$ *constructed from two-pixel patches in the step function case. Parameters:* $N = 2000$, $\sigma = 0.4$, $\varepsilon = 0.1$.

i	0	1	2	3	4	5	6	7	8	9
$\lambda_i^{(2)}$	1.0	0.9999	0.629	0.628	0.621	0.608	0.419	0.410	0.394	0.381

± 1. This dramatically reduces the misidentification probability, because for a patch to be misclassified, the independent noises of *both* coordinates need to be significantly large. This misidentification probability for a patch of size 2 is (compare with (4.5))

$$\Pr\{y(x_i) + y(x_{i+1}) > 0 \,|\, v(x_i) = -1,\, v(x_{i+1}) = -1\} = \frac{1}{2}\operatorname{erfc}\left(\frac{1}{\sigma}\right), \tag{5.3}$$

because the classification of a patch $(y(x_i), y(x_{i+1}))$ is determined by whether it is above or below the diagonal $y(x_i) + y(x_{i+1}) = 0$ (see Figure 3(c)). For $\sigma = 0.4$ we expect only 0.02% of the samples to be misidentified (or 0.2 out of 1000 on average). In this case the misidentification rate is 30 times smaller than that of the single-pixel neighborhood filter.

The second difference has to do with the exit and relaxation times of the 2-D diffusion process. The relaxation time does not change significantly, because it depends only algebraically on the Hessian of the potential at the bottom of the well. In contrast, the exit time depends *exponentially* on the barrier height, which increases by a factor of 2 compared to the 1-D potential barrier (the factor of 2 is explained by the fact that the distance between the Gaussian centers is larger by a factor of $\sqrt{2}$ and that the potential is quadratic). Therefore, the 2-D exit time is much larger than its 1-D counterpart. This means that many more NL averaging iterations may be performed without blurring different level sets.

These two differences are best illustrated by comparing five or 25 iterations of an NL-filter of two-pixel patches (Figures 3(e) and 3(f)) and five or 25 iterations of an NL-filter with a single pixel (Figures 2(e) and 2(f)) for the same level of noise $\sigma = 0.4$. Not only have the misidentifications completely disappeared, but also after 25 iterations, the level sets still remain at $y = \pm 1$ for the two-pixel NL-filter, whereas 25 iterations with the single-pixel NL-filter blur the level sets to $y = \pm 0.8$.

In Tables 1 and 2 we present the empirical eigenvalues $\lambda_i^{(1)}$ and $\lambda_i^{(2)}$ of the denoising matrices $\mathbf{A}$ corresponding to a single-pixel and to two-pixel neighborhood, respectively. First, $1 - \lambda_1^{(2)} \ll 1 - \lambda_1^{(1)}$ since the exit time from one well to the other is significantly larger in the two-pixel case. Next, we observe a twofold numerical multiplicity of the eigenvalues $\lambda_2^{(1)} \approx \lambda_3^{(1)}$ (0.631 and 0.598) corresponding to the noise reduction rate of (3.9), $\lambda = \frac{\sigma^2}{\sigma^2+\varepsilon} = 0.6154$ in the two wells. In the two-pixel case, the numerical multiplicity becomes fourfold: $\lambda_2^{(2)} \approx \lambda_3^{(2)} \approx \lambda_4^{(2)} \approx \lambda_5^{(2)}$ (0.629, 0.628, 0.621, 0.608). The reason is that for each 2-D well there correspond two equal eigenvalues due to separation of variables in the Fokker–Planck operator [21]. This analysis also shows that the noise reduction rate is the same whether one takes a single-pixel or a two-pixel patch. This is also seen in Figure 3(d), where the error after one denoising iteration $y_d - v$ is shown for the single-pixel vs. the two-pixel neighborhood filter. Note that for correctly identified points, the error is the same, as the red curve is the line $y = x$. Therefore, the number of iterations needed for convergence to the denoised values is roughly the same for both type of patches. The key advantage of enlarging the patch size is an increase of the distance between cluster centers, which translates into fewer misidentifications and reduction of the blurring.

When the signal has many interlacing white-black-white-black pixel values (e.g., like a texture in an image or high frequency telegraph signal), we should consider not only the completely white $(1, 1)$ patches and completely black $(-1, -1)$ patches, but also mixed white-black $(1, -1)$ and black-white $(-1, 1)$ patches. The mixed configurations produce two additional clusters (centered at $(1, -1)$ and $(-1, 1)$), and the probability of confusing these configurations with a totally white (or a totally black) patch is equal to that of the single-pixel neighborhood filter. In this case, of course, it is beneficial to apply NL-means denoising with a larger patch size.

One may conclude that it is favorable to use arbitrarily large patches. In practice, however, the signal is sampled only at a finite number of points; therefore, increasing the patch size eventually leads to a poor estimate of the high dimensional density. In other words, the number of sample points required for the diffusion approximation to hold depends exponentially on the intrinsic dimensionality of the patch space. The fact that in practice the method is successfully applied on natural images with relatively large patch sizes (of size 7×7 or 9×9) attests to the intrinsic low dimensionality of the patch space of natural images.

6. $2\mathbf{A} - \mathbf{A}^2$ filters. The analysis of the previous sections emphasized the view of the operator $\mathbf{A}$ as a diffusion operator in the continuum case, or a Markov matrix in the discrete one. In this section we revisit the eigenvector expansion of $\mathbf{A}$, whereby according to (2.14) denoising is achieved by expanding the signal in these eigenvectors and suppressing the contribution of the eigenvectors with small eigenvalues.

Consider, for example, a random telegraph signal corrupted by noise, where the noise-free signal randomly jumps between l distinct levels of y-values (the case $l = 2$ was considered in section 4). The resulting potential of y-values contains l wells. Consequently, the l largest eigenvalues of $\mathbf{A}$ are all close to 1, and their corresponding eigenvectors are piecewise constant inside each of these wells with sharp transitions between them [19]. The remaining eigenvalues and their corresponding eigenvectors capture the relaxation times in each of the individual wells. The noise-free signal can thus be approximately expressed as a linear combination of $\boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{l-1}$ that correspond to the l features of the signal.

In this setting, we would like the denoising operator to suppress the contribution of the noise, as captured in the remaining eigenvectors $\boldsymbol{\psi}_l, \boldsymbol{\psi}_{l+1}, \ldots$. However, iterative application of the denoising operator also suppresses the coefficients of the first l eigenvectors, since each application of $\mathbf{A}$ multiplies each $\boldsymbol{\psi}_j$ by a factor λ_j, which is strictly smaller than 1 for $j > 0$. In other words, not only are the modes corresponding to the noise suppressed, but also the true features of the signal are suppressed, though at a smaller rate. As demonstrated in the

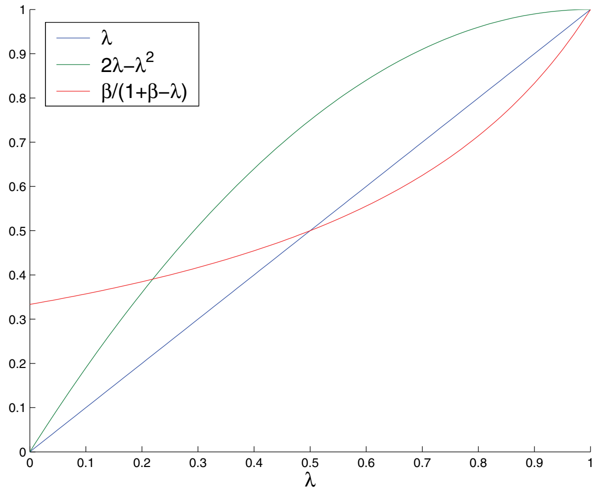


Figure 4. *Low pass filter responses:* λ, $2\lambda - \lambda^2$, *and* $\frac{\beta}{1+\beta-\lambda}$ *(shown with* $\beta = 0.5$*).*

previous sections, this leads to undesirable blurring of the distinct y-levels. An interesting question then is whether it is possible to construct an improved denoising operator from the original operator $\mathbf{A}$.

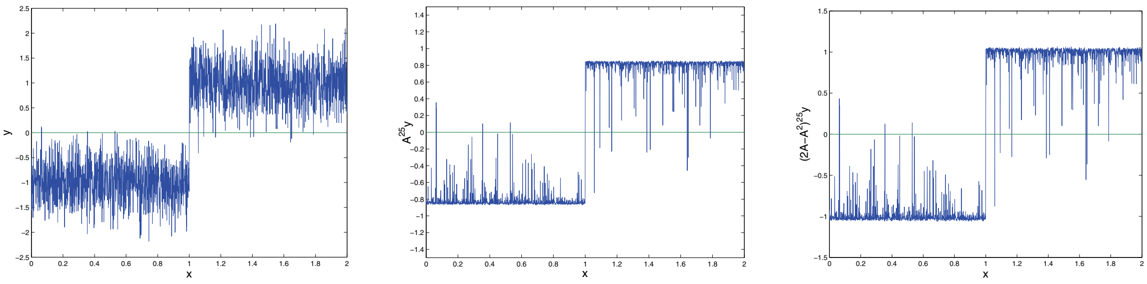
Consider, for example, the operator $\mathbf{A}_2 = 2\mathbf{A} - \mathbf{A}^2$, suggested to us by Ronald Coifman. This operator has the same eigenvectors as $\mathbf{A}$, but its eigenvalues are $2\lambda_j - \lambda_j^2 = 1 - (1 - \lambda_j)^2$. On the interval $[0,1]$ this gives an inverted parabola starting at zero at $\lambda = 0$ with a maximal value of 1 at $\lambda = 1$ (see Figure 4). Therefore, the new operator $\mathbf{A}_2$ has much smaller suppression of the large eigenvalues of $\mathbf{A}$, while still significantly suppressing its eigenvalues that are far away from $\lambda = 1$. As such, it is possible to apply the denoising operator $\mathbf{A}_2$ many more iterations without blurring the distinct y-levels. We illustrate this property in Figures 5(a)–5(c), where from left to right we show a noisy signal with $l = 2$ distinct values, 25 iterations of the operator $\mathbf{A}$, and 25 iterations of $\mathbf{A}_2$, respectively. Note that $\mathbf{A}_2$ is able to perform denoising with far less blurring as compared to $\mathbf{A}$.

In general, from k denoising iterations of $\mathbf{A}$, one can generate any polynomial $P_k(\mathbf{A})$ of degree k. In particular, this polynomial can be chosen to approximate the desired suppression of eigenvalues, such as a low pass filter at some cutoff value. For example, the following recursive scheme was considered by [28]: $P_k(\mathbf{A}) = (1+\beta)^{-1}(P_{k-1}(\mathbf{A}) + \beta I)$, which converges to $\lim_{k\to\infty} P_k(\lambda) = \frac{\beta}{1+\beta-\lambda}$.

Finally, we remark that the operator $\mathbf{A}_2$ has an interesting probabilistic interpretation in itself. Our analysis shows that the operator $\mathbf{A}$ is approximately equal to $\mathbf{I} + \mathcal{L}$, where $\mathcal{L}$ is the backward Fokker–Planck diffusion operator. Therefore,

$$\mathbf{A}_2 = 2\mathbf{A} - \mathbf{A}^2 = \mathbf{I} - (\mathbf{I} - \mathbf{A})^{\mathbf{2}} \approx (\mathbf{I} - \mathcal{L})(\mathbf{I} + \mathcal{L}).$$

This means that denoising using $\mathbf{A}_2$ amounts to running forward the heat equation for time ε (the denoising step—averaging out the noise), followed by running it backward for the same time (sharpening). This method takes a noisy signal, smoothes it by the y-axis diffusion



(a) Step function corrupted by white noise

(b) After 25 iterations of $\mathbf{A}$

(c) After 25 iterations of $2\mathbf{A} - \mathbf{A}^2$

Figure 5. *Comparing a single-pixel filter* $\mathbf{A}$ *with* $2\mathbf{A} - \mathbf{A}^2$. *Parameters:* $N = 2000$, $\sigma = 0.4$, $\varepsilon = 0.1$.

equation, and then tries to recover the original signal by running the diffusion backward in time. We remark that diffusion backward in time has also appeared in other contexts in signal denoising [15].

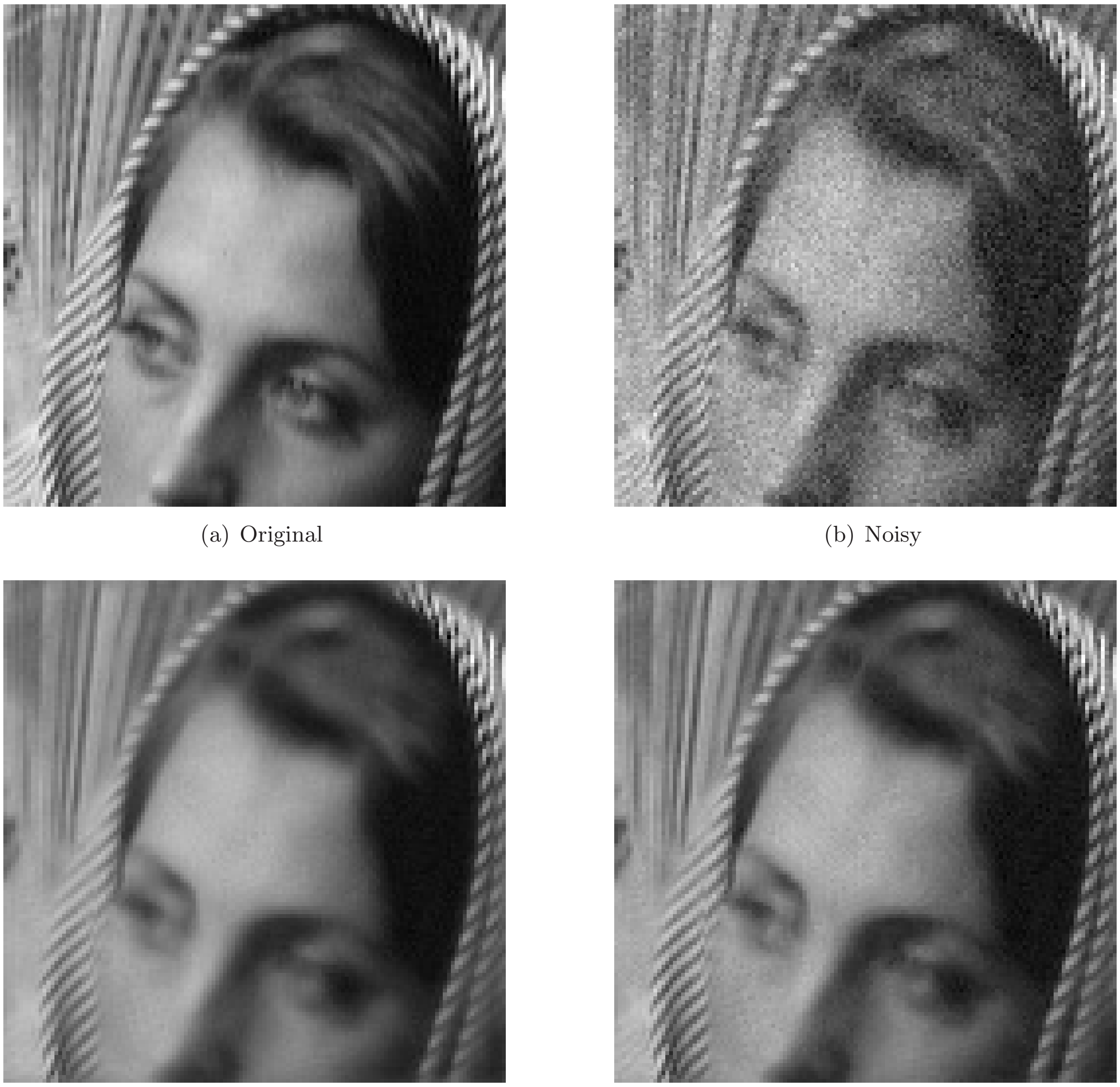
The analysis of the filter $A_2 = 2A - A^2$ is demonstrated in Figures 6–9. The figures are generated as follows. From each of the images in Figures 6(a) and 8(a), we take a subimage of size 128×128 and normalize it to have zero mean and unit standard deviation. We denote the 128×128 subimages by $I^{(1)}$ and $I^{(2)}$, respectively. We then add to the images zero-mean additive Gaussian white noise to achieve a prescribed signal-to-noise ratio (SNR). The SNR is defined as the ratio between the standard deviation of the clean image and the standard deviation of the noise. We use an SNR of 4 for Figure 6(a) and an SNR of 3.5 for Figure 8(a). The resulting noisy images are shown in Figures 6(b) and 8(b) and are denoted $\tilde{I}^{(1)}$ and $\tilde{I}^{(2)}$, respectively. From each noisy image $\tilde{I}^{(i)}$ we construct an operator $A^{(i)}$. To that end, we attach to each pixel j in the image a feature vector P_j, given by its 7×7 patch around that pixel, which we treat as a 49-dimensional vector. We then find for each feature vector P_j its 150 nearest neighbors, denoted $\mathcal{N}_j$, and construct the graph $W^{(i)}$, represented as an $N^2 \times N^2$ sparse matrix, $N = 128$, by

$$W^{(i)}_{jk} = e^{-\|P(j)-P(k)\|^2/\varepsilon^2}, \qquad j = 1, \ldots, N, \quad k \in \mathcal{N}_j,$$

with $\varepsilon = 10\sigma$ (see [10]), where σ is the standard deviation of the noise. We furthermore discard all entries in $W^{(i)}$ that are smaller than 10^{-8}. The number of nearest neighbors and the value of σ are chosen empirically such that the resulting graph is connected but still sparse enough to allow for efficient processing. Finally, we obtain $A^{(i)}$ by normalizing $W^{(i)}$ to be row-stochastic as in (2.8).

We reshape each $\tilde{I}^{(i)}$ in Figures 6(b) and 8(b) as a vector of length N^2 and apply on it its corresponding operators $A^{(i)}$ and $A^{(i)}_2 = 2A^{(i)} - \left(A^{(i)}\right)^2$. We then reshape the resulting vector back into an $N \times N$ image and denote the result of the application of $A^{(i)}$ and $A^{(i)}_2$ by $I^{(i)}_A$ and $I^{(i)}_{A_2}$, respectively. The outcome is shown in Figures 6(c) and 6(d) for the Barbara image, and in Figures 8(c) and 8(d) for the Lena image.

The residual errors for the Barbara image, given by the norms of the difference between



(a) Original

(b) Noisy

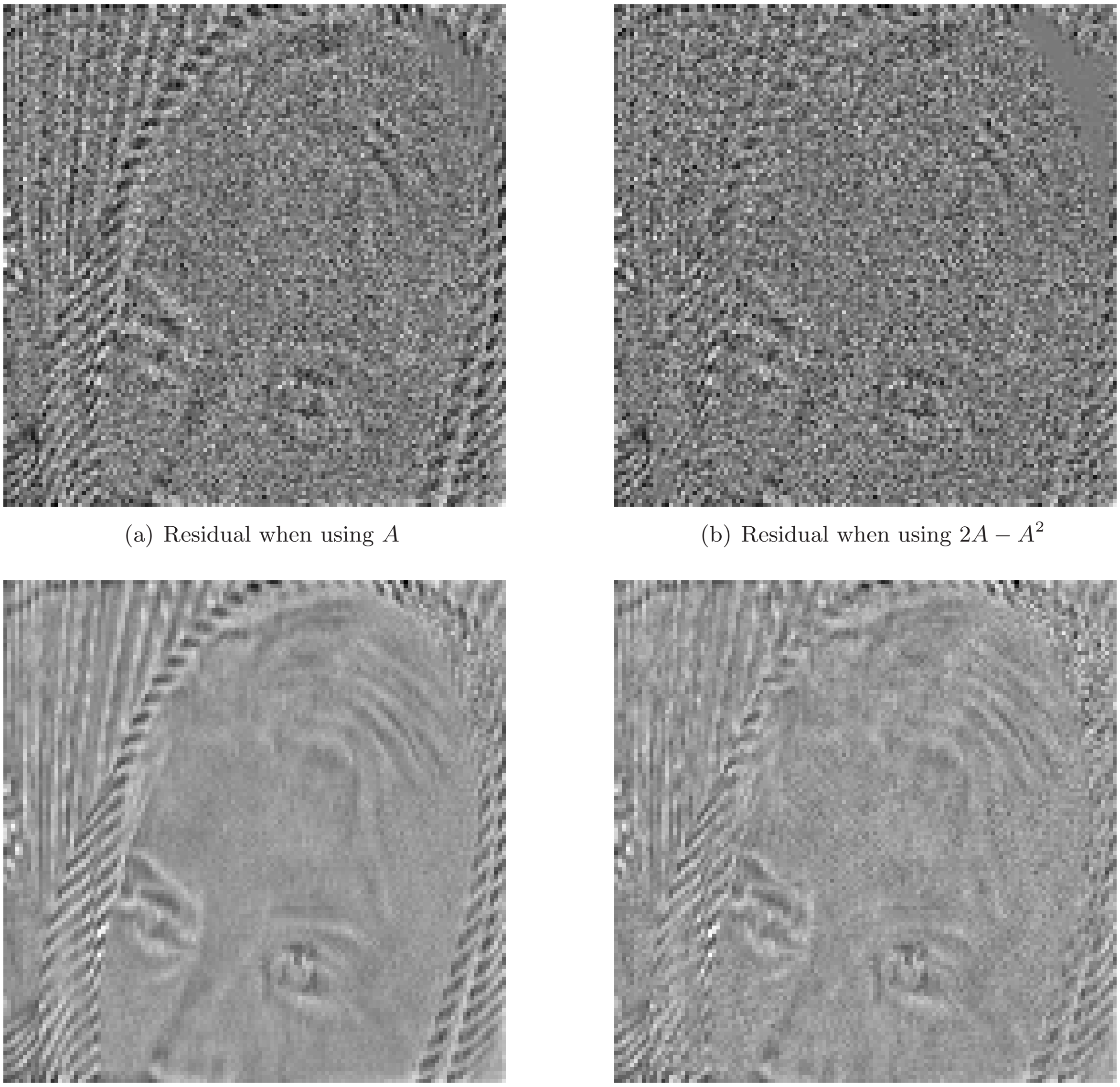
(c) Denoised using the operator A

(d) Denoised using the operator $2A - A^2$

Figure 6. *Demonstration of the operator $A_2 = 2A - A^2$ for the Barbara image.*

the original image and its denoised versions, are

$$\|I^{(1)} - I_A^{(1)}\| = 0.192111,$$
$$\|I^{(1)} - I_{A_2}^{(1)}\| = 0.149881,$$



(a) Residual when using A

(b) Residual when using $2A - A^2$

(c) Lost features when using A

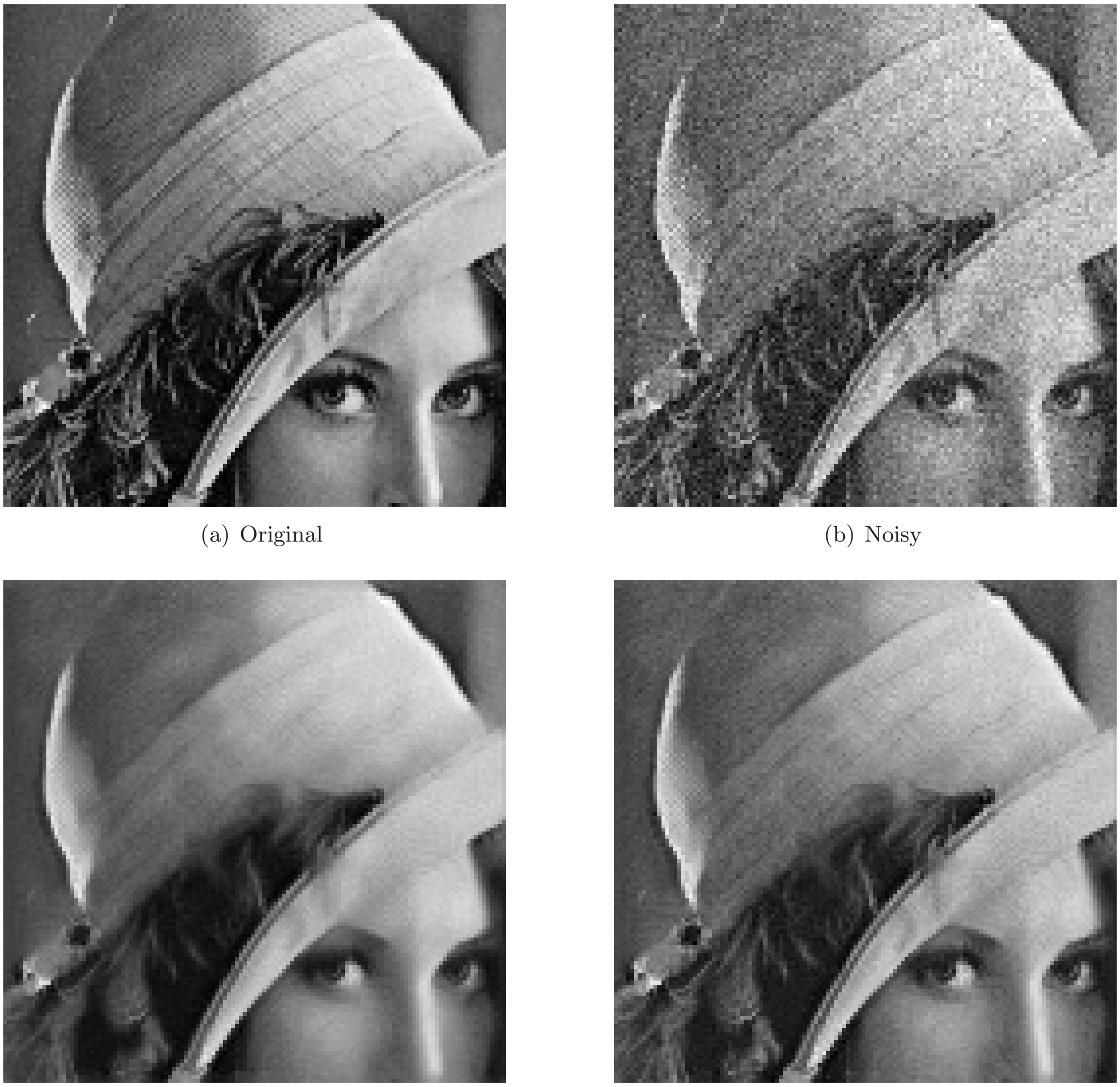
(d) Lost features when using $2A - A^2$

Figure 7. *Demonstration of the operator $A_2 = 2A - A^2$ for the Barbara image (continued).*

and for the Lena image

$$\|I^{(2)} - I_A^{(2)}\| = 0.235994,$$
$$\|I^{(2)} - I_{A_2}^{(2)}\| = 0.198785.$$

That is, the residual errors of the operators $A_2^{(i)}$ are smaller. Note, however, that these numbers do not reveal the true effect of the operators, as demonstrated in Figures 7 and 9.



(a) Original

(b) Noisy

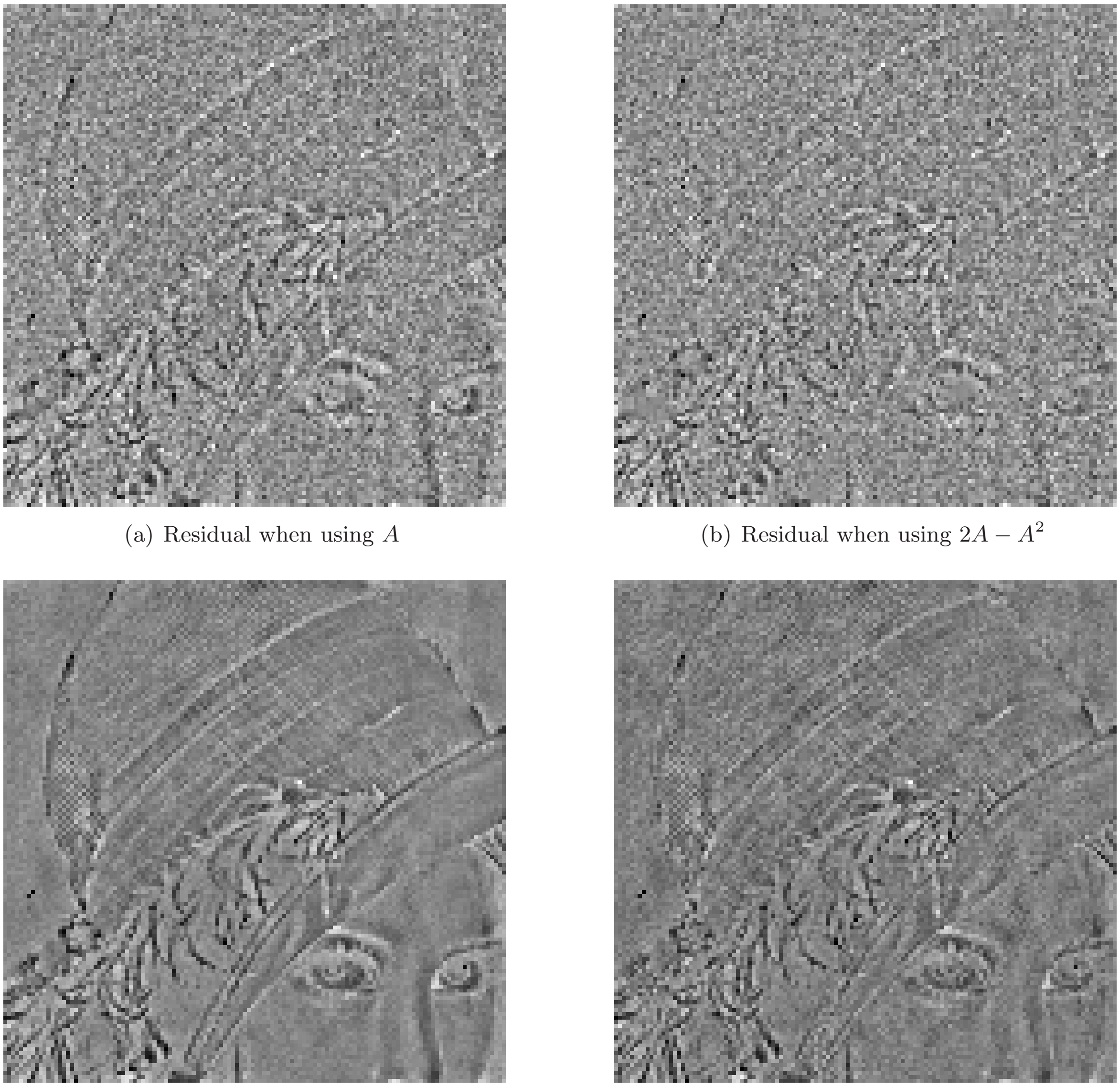
(c) Denoised using the operator A

(d) Denoised using the operator $2A - A^2$

Figure 8. *Demonstration of the operator $A_2 = 2A - A^2$ for the Lena image.*

In Figures 7(a) and 7(b) we show the difference $\tilde{I}^{(1)} - I_A^{(1)}$ and $\tilde{I}^{(1)} - I_{A_2}^{(1)}$, respectively, that is, the difference between the noisy image in Figure 6(b) and its denoised versions from Figures 6(c) and 6(d), respectively. For an ideal denoising scheme, this difference should look like noise. It is apparent that the residual for the operator $A_2^{(1)}$ in Figure 7(b) exhibits fewer features than the residual for the operator $A^{(1)}$ in Figure 7(a). Figures 9(a) and 9(b) show this difference as well as a similar behavior for the Lena image.

In Figures 7(c) and 7(d) we show $I^{(1)} - I_A^{(1)}$ and $I^{(1)} - I_{A_2}^{(1)}$, respectively, that is, the



(a) Residual when using A

(b) Residual when using $2A - A^2$

(c) Lost features when using A

(d) Lost features when using $2A - A^2$

Figure 9. *Demonstration of the operator $A_2 = 2A - A^2$ for the Lena image (continued).*

difference between the original Barbara image and the outcomes of the operators A and A_2. This provides a qualitative measure for the sharpening effect described above (and thus for the “feature loss” incurred by each operator). In Figures 9(c) and 9(d) we show the same difference for the Lena image. It is apparent that the residual when using the operator A pronounces more features than the residual of the operator A_2, supporting the analysis given above.

7. Summary.

This paper is devoted to the analysis of NL neighborhood filters, by interpreting the explicit formula for neighborhood filters as the transition operator associated to a random walk in the space of patches. The study of the spectrum of this matrix allows us to infer some properties of NL neighborhood filters, particularly when they are iterated (which corresponds to longer times in the random walk). We rely on earlier results that show that the transition matrix approximates the backward Fokker–Planck diffusion operator, and that the matrix eigenvectors approximate the operator eigenfunctions.

The benefits of this interpretation become evident in sections 4 and 5, where we study a step function denoised with patch sizes of one pixel and two pixels. In both cases, the filter can be interpreted as a random process in a double well potential. This interpretation enables the prediction and understanding of the behavior of the filter in these settings. Relevant parameters are studied, such as the number of iterations needed for convergence. The blurring effect corresponding to a high number of iterations is explained, as well as the benefits of increasing the size of the patch from one to two samples. Finally, in section 6 we propose a new operator, with a slower decay of the eigenvalues, thus attenuating the blurring effect. Although the analysis is restricted to somewhat simple cases, the interpretation proposed in this paper provides insight into the understanding of NL-filters.

Acknowledgments. We thank Ronald Coifman and Arthur Szlam for stimulating discussions. We also thank the anonymous referees for their valuable criticism, which greatly improved the exposition of this paper. Part of this work was conducted while the authors visited the Institute for Pure and Applied Mathematics (IPAM), for which we thank IPAM, Mark Green, and Peter Jones.

REFERENCES

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1972.

[2] S. P. Awate, *Adaptive Nonparametric Markov Models and Information-Theoretic Methods for Image Restoration and Segmentation*, Ph.D. dissertation, School of Computing, University of Utah, Salt Lake City, UT, 2006.

[3] S. P. Awate and R. T. Whitaker, *Unsupervised, information-theoretic, adaptive image filtering for image restoration*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 364–376.

[4] D. Barash, *A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation*, IEEE Trans. Pattern Anal. Mach. Intell., 24 (2002), pp. 844–847.

[5] D. Barash and D. Comaniciu, *A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift*, Image Vision Comput., 22 (2004), pp. 73–81.

[6] M. Belkin and P. Niyogi, *Towards a theoretical foundation for Laplacian-based manifold methods*, in Proceedings of the 18th Conference on Learning Theory (COLT), 2005, pp. 486–500.

[7] T. Brox and D. Cremers, *Iterated nonlocal means for texture restoration*, in Scale Space and Variational Methods in Computer Vision, Lecture Notes in Comput. Sci. 4485, F. Sgallari, A. Murli, and N. Paragios, eds., Springer-Verlag, Berlin, Heidelberg, 2007, pp. 13–24.

[8] A. Buades, *Image and Movie Denoising by Nonlocal Means*, Ph.D. dissertation, Universitat de les Illes Balears, Palma de Mallorca, Spain, 2006.

[9] A. Buades, B. Coll, and J. M. Morel, *Neighborhood filters and PDE's*, Numer. Math., 105 (2006), pp. 1–34.

[10] A. Buades, B. Coll, and J. M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., 4 (2005), pp. 490–530.

[11] Y. Cheng, *Mean shift, mode seeking, and clustering*, IEEE Trans. Pattern Anal. Mach. Intell., 17 (1995), pp. 790–799.

[12] R. R. Coifman and D. Donoho, *Translation-invariant de-noising*, in Wavelets and Statistics, Springer-Verlag, New York, 1995, pp. 125–150.

[13] G. W. Gardiner, *Handbook of Stochastic Processes for Physics, Chemistry and the Natural Sciences*, 2nd ed., Springer-Verlag, Berlin, 1985.

[14] G. Gilboa and S. Osher, *Nonlocal linear image regularization and supervised segmentation*, Multiscale Model. Simul., 6 (2007), pp. 595–630.

[15] G. Gilboa, N. Sochen, and Y. Y. Zeevi, *Forward-and-backward diffusion processes for adaptive image enhancement and denoising*, IEEE Trans. Image Process., 11 (2002), pp. 689–703.

[16] M. Hein, J. Audibert, and U. von Luxburg, *From graphs to manifolds—weak and strong pointwise consistency of graph Laplacians*, in Proceedings of the 18th Conference on Learning Theory (COLT), 2005, pp. 470–485.

[17] S. Kindermann, S. Osher, and P. W. Jones, *Deblurring and denoising of images by nonlocal functionals*, Multiscale Model. Simul., 4 (2005), pp. 1091–1115.

[18] M. Mahmoudi and G. Sapiro, *Fast image and video denoising via nonlocal means of similar neighborhoods*, IEEE Signal Process. Lett., 12 (2005), pp. 839–842.

[19] B. J. Matkowsky and Z. Schuss, *Eigenvalues of the Fokker–Planck operator and the approach to equilibrium for diffusions in potential fields*, SIAM J. Appl. Math., 40 (1981), pp. 242–254.

[20] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, *Diffusion maps, spectral clustering and eigenfunctions of Fokker–Planck operators*, in Advances in Neural Information Processing Systems, Vol. 18, Y. Weiss, B. Schölkopf, and J. Platt, eds., MIT Press, Cambridge, MA, 2006, pp. 955–962.

[21] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, *Diffusion maps, spectral clustering and reaction coordinates of dynamical systems*, Appl. Comput. Harmon. Anal., 21 (2006), pp. 113–127.

[22] P. Perona and J. Malik, *Scale space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell., 12 (1990), pp. 629–639.

[23] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[24] Z. Schuss, *Theory and Applications of Stochastic Differential Equations*, Wiley, New York, 1980.

[25] A. Singer, *From graph to manifold Laplacian: The convergence rate*, Appl. Comput. Harmon. Anal., 21 (2006), pp. 128–134.

[26] S. M. Smith and J. M. Brady, *Susan—A new approach to low level image processing*, Int. J. Comput. Vis., 23 (1997), pp. 45–78.

[27] A. D. Szlam, *Non-stationary Analysis on Datasets and Applications*, Ph.D. dissertation, Yale University, New Haven, CT, 2006.

[28] A. D. Szlam, M. Maggioni, and R. R. Coifman, *Regularization on graphs with function-adapted diffusion processes*, J. Mach. Learn. Res., 9 (2008), pp. 1711–1739.

[29] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images*, in Proceedings of the Sixth International Conference on Computer Vision (ICCV), IEEE Computer Society, Washington, D.C., 1988, pp. 839–846.

[30] L. P. Yaroslavsky, *Digital Picture Processing—An Introduction*, Springer-Verlag, Berlin, 1985.