Preprint
UCRL-JC-145574

PDE-based Non-linear Diffusion Techniques for Denoising Scientific and Industrial Images: an Empirical Study

S. K. Weeratunga and C. Kamath

This article was accepted in
Image Processing: Algorithms and Systems Conference, SPIE Electronic Imaging Symposium, San Jose, January 20-25, 2002

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

December 1, 2001

PDE-based non-linear diffusion techniques for denoising scientific and industrial images: an empirical study

Sisira K. Weeratunga and Chandrika Kamath

Center for Applied Scientific Computing, L-561, P.O. Box 808
Lawrence Livermore National Laboratory, Livermore, CA 94551

ABSTRACT

Removing noise from data is often the first step in data analysis. Denoising techniques should not only reduce the noise, but do so without blurring or changing the location of the edges. Many approaches have been proposed to accomplish this; in this paper, we focus on one such approach, namely the use of non-linear diffusion operators. This approach has been studied extensively from a theoretical viewpoint ever since the 1987 work of Perona and Malik showed that non-linear filters outperformed the more traditional linear Canny edge detector. We complement this theoretical work by investigating the performance of several isotropic diffusion operators on test images from scientific domains. We explore the effects of various parameters such as the choice of diffusivity function, explicit and implicit methods for the discretization of the PDE, and approaches for the spatial discretization of the non-linear operator etc. We also compare these schemes with simple spatial filters and the more complex wavelet-based shrinkage techniques. Our empirical results show that, with an appropriate choice of parameters, diffusion-based schemes can be as effective as competitive techniques.

Keywords: Denoising, non-linear isotropic diffusion, partial differential equations

1. INTRODUCTION

One of the first steps in data analysis is the identification of objects in the data. This is usually done through techniques such as edge detection, which use the local gradient of the data to find the edges. In data that is noisy, the measurement of this gradient can be suspect. To overcome this problem, the data is often de-noised, or smoothed, before the process of edge detection. Traditionally, de-noising of data has been done through the use of simple filters such as the mean filter or the Gaussian filter. While these filters smooth the noise in the data, they also smooth the edges, blurring them and changing their location. To address this problem, several alternative techniques have been proposed, including schemes based on wavelets[1, 2] and partial differential equations.[3–7]

In this paper, we focus on the use of non-linear diffusion techniques for denoising scientific images. These techniques, based on the use of partial differential equations, have been extensively studied since the early work of Perona and Malik in 1987.[3, 4] Much of this work has been of a theoretical nature, and there is little information to guide a practitioner on the choice of various parameters used in the implementation of these methods. We attempt to address this shortcoming by exploring the effects of several of these parameters using test images from scientific domains that have been corrupted with additive Gaussian noise. We also compare the performance of these techniques with the commonly used spatial filters, and the more complex wavelet-based thresholding techniques.

This paper is organized as follows. In Section 2, we introduce the idea behind the non-linear diffusion equation and describe the parameters that must be chosen in order to define the equation and solve it numerically. In Section 3, we describe our numerical experiments with diffusion-based denoising using a synthetic test image and two real images from scientific domains. We also compare and contrast these techniques with spatial filters and wavelet-based thresholding schemes. Finally, in Section 4, we conclude with a brief summary of our work and ideas for future research.

Further author information: (Send correspondence to C.K.)
C.K.: E-mail: kamath2@llnl.gov

2. DIFFUSION-BASED DENOISING

The idea behind the use of the diffusion equation in image processing arose from the use of the Gaussian filter in multi-scale image analysis. Convolving an image with a Gaussian filter K_σ:

$$K_\sigma(x,y) = \frac{1}{2\pi\sigma^2} exp\Big(-\frac{|x|^2+|y|^2}{2\sigma^2}\Big) \tag{1}$$

with standard deviation σ, is equivalent to the solution of the diffusion equation in two dimensions

$$\frac{\partial}{\partial t} I(x,y,t) = \nabla I(x,y,t) = \frac{\partial^2}{\partial x^2} I(x,y,t) + \frac{\partial^2}{\partial y^2} I(x,y,t) \tag{2}$$

where $I(x,y,t)$ is the two-dimensional image $I(x,y)$ at time $t = 0.5\sigma^2$, with initial conditions $I(x,y,0) = I_0(x,y)$, where I_0 is the original image. In a general form, this can be written as

$$\begin{aligned} \frac{\partial}{\partial t} I(x,y,t) &= \nabla \cdot \Big(c(x,y,t)\nabla I(x,y,t)\Big) \\ I(x,y,0) &= I_0(x,y) \end{aligned} \tag{3}$$

where $c(x,y,t)$ is the diffusion conductance or diffusivity of the equation, ∇ is the gradient operator, and $\nabla\cdot$ is the divergence operator. If c is a constant, independent of x, y, or t, it leads to a linear diffusion equation, with a homogeneous diffusivity. In this case, all locations in the image, including the edges are smoothed equally. This is, of course, undesirable, and a simple improvement would be to change c with the location x and y in the image, thus converting the equation into a linear diffusion equation with non-homogeneous diffusivity. If the function c were image dependent, then the linear diffusion equation becomes a non-linear diffusion equation. This approach was proposed by Perona and Malik in their 1987 work,[3,4] where they considered two values for the function c:

$$c(x,y,t) = 1/\Big(1 + \frac{|\nabla I|^2}{K^2}\Big) \tag{4}$$

and

$$c(x,y,t) = exp\Big(-|\nabla I|^2/2K^2\Big) \tag{5}$$

Here, c varies as a function of the image data, and is small where the gradient of the image is large, resulting in lower diffusivity near the edges. The conductance parameter K enables backward diffusion when it is smaller than the gradient ∇I, thus enhancing the edges.

Note that if c is calculated based on the gradient of the original image I_0, it is independent of the time t, resulting in a linear non-homogeneous diffusion equation. In the work of Perona-Malik, the gradient in c is taken of the image at time t, making it a non-linear, non-homogeneous diffusion equation. While Perona-Malik, as well as several other authors, use the term "anisotropic" to refer to the case where the diffusivity is a scalar function varying with the location, we reserve this term for the case where the diffusivity is a tensor, varying with both the location and the direction. Instead, we use the term "non-homogeneous" to refer to the case where the scalar diffusivity varies with location. This terminology is consistent with that used in the partial differential equation literature.

As several authors have observed, the early work of Perona-Malik resulted in an ill-posed problem, where images close to each other could produce divergent solutions and very different edges. A common solution to this problem is to use a regularized (or smoothed) version of the image to calculate the gradient in Equations (4) and (5). In this approach, the gradient ∇I is replaced by

$$\nabla I_G = \nabla(G_{\sigma_r} * I(x,y,t)) \tag{6}$$

where G_{σ_r} is another Gaussian with standard deviation σ_r. In other words, the gradient is taken after the image at time t is smoothed by convolution with a Gaussian. σ_r can be considered as a regularization parameter which

describes a uniform smoothing used to calculate the gradient of the image at time t. This σ_r should not be confused with the σ of Equation (1) which characterizes the time parameter of the smoothing through diffusion.

Once the formulation of Perona-Malik has been converted into a well-posed problem, there are several parameters that must be chosen in the application of the diffusion equation to an image. These parameters can be broadly categorized into two types - those associated with the partial differential equation (3) itself, and those related to the numerical solution of the partial differential equation (PDE). We next discuss these parameters in more detail.

2.1. The diffusion equation

A key choice in the diffusion equation (3) is the choice of the diffusivity function, $c(x, y, t)$. Following the original two suggestions by Perona-Malik (Equations (4) and (5), and their regularized versions), several authors have suggested variations or enhancements. For example, Charbonnier et. al[8, 9] suggested using the diffusivity

$$c(x, y, t) = \left(1 + \frac{|\nabla I_G|^2}{K^2}\right)^{-1/2} \tag{7}$$

while Weickert[10] suggested the use of

$$c(x, y, t) = \begin{cases} 1 & \text{if } |\nabla I_G| = 0 \\ 1 - exp\left(-\frac{C_m}{(|\nabla I_G|^2/K^2)^m}\right) & \text{if } |\nabla I_G| > 0 \end{cases} \tag{8}$$

where the coefficient C_m is derived from m using

$$1 = exp(-C_m)(1 + 2C_m m). \tag{9}$$

For $m = 2,\ 3,$ and 4, $C_m = 2.33666,\ 2.9183,$ and 3.31488, respectively. This value of the diffusivity function is designed to preserve edges better, with the effect enhanced the most for larger values of m.

In addition to the form of the diffusivity, another choice is the value of σ_r used in the regularization of the image to calculate the gradient (see Equation 6). Although σ_r is often kept constant throughout the evolution of the PDE, this may not always be desirable. As the image is smoothed, and the unwanted intensity variations diminish faster than the signal, the gradient measurements become more reliable. To account for this, the parameter σ_r can be reduced as the equation evolves.[11]

Another parameter to be determined in the diffusivity is the conductance parameter K. This is typically done experimentally, with a single value of K used during the entire evolution of the PDE. Li and Chen[12] have suggested that this might not be appropriate. When the gradient magnitude in the image exceeds the value K, the corresponding edge is enhanced. The assumption made is that the gradient magnitude of the noise is typically smaller than that of the edges in the image. But, in some images, this may not be the case, as a result of which, the edges and the noise (with gradient magnitude greater than K) are both enhanced. Further, by fixing the value of K for the entire process, edges that lie beyond the range of edge enhancement can never be enhanced. Both these problems can be addressed by making K a function of the time t. If K is set high at the beginning, it can reduce the noise significantly, while enhancing edges with very high gradients. As the image is smoothed, the intensity variation of the noise reduces more than that of the signal. Since the strength of the noise has been reduced, the value of K can also be reduced without any adverse effects. This would allow edges with lower gradients to be enhanced, without correspondingly enhancing the noise. Thus, by changing K with time, edges with different gradients are enhanced at different times in the evolution of the PDE.

In addition to the above parameters, we also need to determine the time duration for the evolution of the PDE. If this time is very large, the entire image is smoothed. Typically, this time duration is determined experimentally, and varies with the image.

2.2. The solution of the diffusion equation

Once the diffusivity function ($c(x,y,t)$), the regularization parameter (σ_r), and the conductance parameter ($K(t)$) have been chosen, the diffusion equation must be solved on the discrete grid represented by the image. Here, one can borrow extensively from the work that has already been done in the area of the solution of partial differential equations through techniques such as finite difference and finite element methods.[13] In this section, we focus on the finite difference approach to the solution of a generic diffusion equation (3).

The finite difference approach essentially considers the discretized version of the image $I(x,y)$ to correspond to the intensity at the pixels (i,j), at locations (x_i, y_j), where $i = 1,\ldots,N$, and $j = 1,\ldots,M$ (see Figure 1). The distance between the centers of the pixels, referred to as the grid spacing, is h. The same inter-pixel distance h is used along the x and y dimensions. The PDE evolves with a time step of Δt, and the pixel (i,j) at the discrete time $t_k = k * \Delta t$ is referred to as $I(i,j,k)$.

There are several ways in the diffusion equation (3) can be discretized using the finite difference approach.[7,14] Using central differencing in the spatial domain, the one-dimension version of Equation (3) can be written as:

$$\frac{I(i,k+1) - I(i,k)}{\Delta t} = \frac{c(i+1,k) + c(i,k)}{2h^2}\Big(I(i+1,k) - I(i,k)\Big) + \frac{c(i-1,k) + c(i,k)}{2h^2}\Big(I(i-1,k) - I(i,k)\Big) \quad (10)$$

for the pixels in the interior of the image. For the boundary pixels, reflecting boundary conditions are used to obtain the values of the pixels not defined in the above equation. This is the explicit discretization of the PDE, where the image at time step $k+1$ is written explicitly in terms of the image at time step k. The right-hand side of the above equation can be written in terms of a sparse-matrix-vector multiply[7]

$$\frac{I(k+1) - I(k)}{\Delta t} = A(I(k))\ I(k) \quad (11)$$

where $I(k)$ defines the image at time step k. The elements of the matrix $A(I(k))$ are defined as:

$$A(I(k)) = [a_{ij}(I(k))] = \begin{cases} \frac{c(j,k)+c(i,k)}{2h^2} & \text{for } j = i-1 \text{ and } j = i+1 \\ -\frac{c(i-1,k)+c(i,k)}{2h^2} - \frac{c(i+1,k)+c(i,k)}{2h^2} & \text{for } j = i \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

This is a tridiagonal matrix, with nonzeros along the main diagonal ($j = i$), the first super-diagonal ($j = i+1$) and the first sub-diagonal ($j = i-1$), where i is the row number and j the column number in the matrix. As a result of this sparsity, the matrix-vector multiply requires few operations and the explicit approach is computationally very efficient. However, it requires small time steps for stability. As a result, more time steps are required to reach a particular time t_k.

To work around this stability constraint, a typical approach is to consider the semi-implicit version of the discretization

$$\frac{I(k+1) - I(k)}{\Delta t} = A(I(k))\ I(k+1) \quad (13)$$

where the right hand side of Equation (11) has been replaced by the image at time step $(k+1)$. This can be written as

$$\Big\{I - \Delta t\ A(I(k))\Big\}\ I(k+1) = I(k). \quad (14)$$

This requires the solution of a linear system of equations, which in the case of the one-dimensional problem requires the solution of a tridiagonal system of equations. This can be done efficiently with the Thomas' algorithm.[15]

These explicit and semi-implicit discretizations can be easily extended to the case of two-dimensional data, giving rise to the equations

$$I(k+1) = I(k) + \Delta t\ \Big\{A_x(I(k)) + A_y(I(k))\Big\} I(k) \quad (15)$$

and

$$\left\{I - \Delta t \left(A_x(I(k)) + A_y(I(k))\right)\right\} I(k+1) = I(k). \tag{16}$$

for the explicit and semi-implicit schemes, where the matrices A_x and A_y represent the matrices A along the x and y axes, respectively. However, in two- and higher dimensions, the solution of the system of equations given in (16) is not as simple as it was in the case of the one-dimensional problem. This is because, in the one-dimensional problem, the numbering of the pixels from $i = 1, \ldots, N$, enabled neighboring pixels to be adjacent in the matrix, resulting in a matrix with a small bandwidth of three. In the two dimensional case, the pixels in the image are ordered using natural ordering, with the pixels in one row of the image following the pixels in the previous row. Referring to Figure 2, this means that the east and west neighbors of a pixel are adjacent in the matrix, but the north and south neighbors are N pixels away, where N is the number of pixels along the x axis. The resulting matrix is penta-diagonal as shown in Figure 3, with a half-bandwidth of N.

There are several ways in which these complex systems of equations can be solved.[15] Since these systems have to be solved at each time step in the solution of the diffusion equation, we are interested in techniques that converge fast, are computationally efficient, and have relatively low memory requirements.

At this point, it would be appropriate to point out that there are actually two PDEs that are being solved in the solution of Equation (3). The first is the diffusion equation itself. However, in addition, in the diffusivity function, we also need to calculate the gradient of the regularized image. This regularized image is obtained by convolving the image with a Gaussian, a task that can also be viewed as the solution of another "inner" diffusion equation of the form in Equation (2), where the diffusivity is a constant equal to unity. The same techniques described above can also be used to solve this inner diffusion equation that arises in the calculation of the diffusivity at each step of the outer diffusion equation.

2.3. Solving the linear system of equations

We next consider techniques that can be used to solve the linear system of equations described in Equation (16). In the following, we use $I_k = I(k)$.

- Alternating Direction Implicit (ADI) method: In this approach, Equation (16) is solved by alternatively treating one of the dimensions as implicit and the other explicit. So, at each time step, two tridiagonal systems are solved:

$$\left\{I - \Delta t\ A_x(I_k)\right\} I_{k*} = \left\{I + \Delta t\ A_y(I_k)\right\} I_k \tag{17}$$

$$\left\{I - \Delta t\ A_y(I_{k*})\right\} I_{k+1} = \left\{I + \Delta t\ A_x(I_{k*})\right\} I_{k*} \tag{18}$$

  First, we solve for the image at time-step $k+1/2$, assuming the $x-$dimension as implicit. Then, the solution at time-step $k+1$ is obtained assuming the $y-$dimension is implicit and using the image obtained at time-step $k+1/2$. This scheme is second order accurate in time. However, the operator is non-symmetric and different results can be obtained depending on the order in which the directions are treated as implicit.[7, 14]

- Additive Operator Splitting (AOS) method: In this method, Equation (16), which can be re-written as:

$$I_{k+1} = \left\{I - \Delta t \left(A_x(I_k) + A_y(I_k)\right)\right\}^{-1} I_k \tag{19}$$

  is solved using the approximation

$$I_{k+1} = \frac{1}{2}\left\{\left(I - 2\Delta t A_x(I_k)\right)^{-1} + \left(I - 2\Delta t A_y(I_k)\right)^{-1}\right\} I_k \tag{20}$$

  This again reduces to the solution of two tridiagonal systems. Note that this is a symmetric scheme, where interchanging the dimensions x and y does not change the solution. This operator is first order accurate in time.

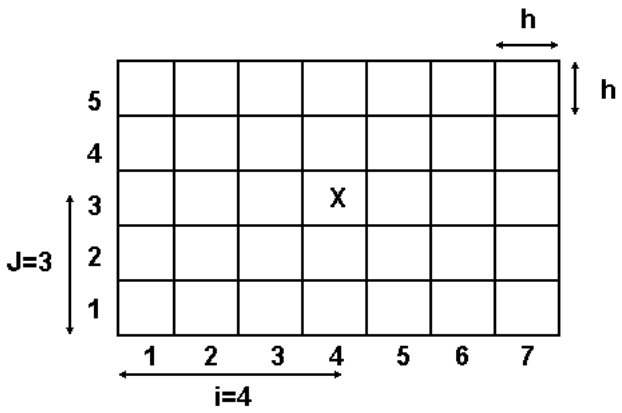


Figure 1: Discretization of an image, of dimension (7,5), where the pixel marked with an X is at location (4,3).

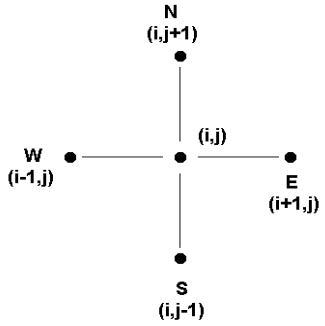


Figure 2: Neighboring points for a pixel at location (i,j) for a two-dimensional problem

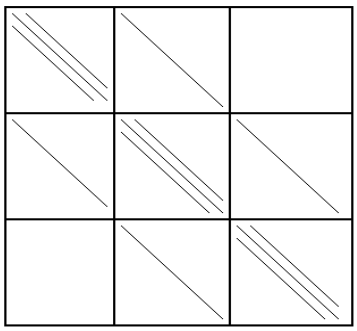


Figure 3: Pentadiagonal matrix that must be solved in the semi-implicit case for a two-dimensional problem

- Approximate Factorization Implicit (AFI) method: In this approach, we solve the system

$$I_{k+1} = \frac{1}{2}\Big\{\Big(I - \Delta t A_x(I_k)\Big)^{-1}\Big(I - \Delta t A_y(I_k)\Big)^{-1} + \Big(I - \Delta t A_y(I_k)\Big)^{-1}\Big(I - \Delta t A_x(I_k)\Big)^{-1}\Big\} I_k \quad (21)$$

  This scheme is first-order accurate in time, and preserves symmetry. It involves the solution of four tridiagonal systems.

All the schemes above, as well as the explicit scheme, are discrete approximations to the continuous diffusion equation. They vary in their computational complexity, convergence rate, and memory requirements.

2.4. Spatial discretization of the diffusivity

In discretizing the diffusion equation, values of the image pixels are taken as the values at the nodes of the discretized grid (see Figure 2). However, in calculating the diffusivity, we need to evaluate it at locations $(i + 1/2, j)$, $(i - 1/2, j)$, $(i, j - 1/2)$, and $(i, j + 1/2)$. One approach is to use the node-centered evaluation of the diffusivity by taking the edge-centered values as the arithmetic average of the two neighboring node values. For example,

$$c(i + 1/2, j) = 1/2(c(i, j) + c(i + 1, j)). \quad (22)$$

In contrast, the edge-centered diffusivity can be evaluated directly by using the edge-centered gradient of the image. In this approach, the edge-centered gradients are computed based on the central difference approximation using only the image data at the two nodes of the edge. This is the approach used by Perona-Malik in their original paper.[4]

3. EXPERIMENTAL RESULTS

In this section, we describe the results of our experiments using non-linear isotropic diffusion to denoise various scientific images. Using images which have been corrupted by additive Gaussian noise, we explore the effectiveness of denoising as we change the various parameters described above.

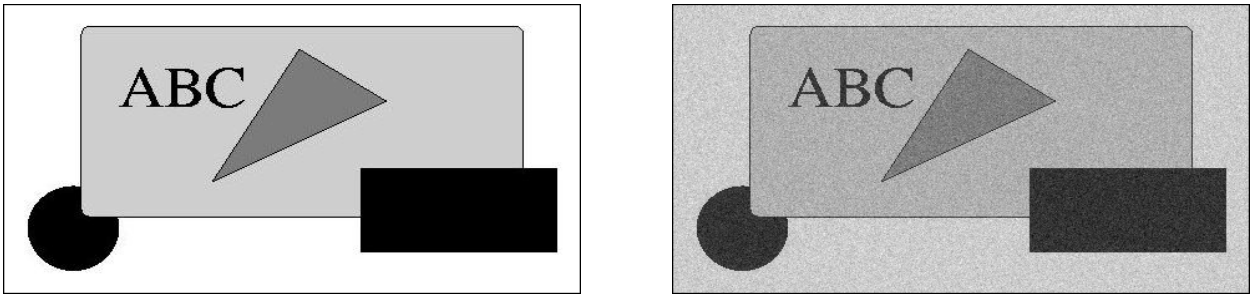
3.1. Experimental images

In order to evaluate the different diffusion-based denoising methods, we have chosen three test images (see Figure 4). The first is a synthetic image chosen the enable experimentation with different parameters. This is a rectangular image, and unlike most images that have dimensions that are power of 2, its size is 462 by 249. The other two images are from the scientific domain - one is an image of a lung taken from http://www-itg.lbl.gov/ImgLib.project/homepage.html and the other is an image of a galaxy from http://astro.u-strasbg.fr/ fmurtagh/multires/quick.html. In order to evaluate the effects of applying diffusion-based denoising, we have added Gaussian noise, with standard deviation of 20, to each of these images, as shown in Figure 4. Once the diffusion process is applied to the noisy image, its effectiveness in removing the noise is obtained by calculating the mean-squared error(MSE) relative to the original image as follows:

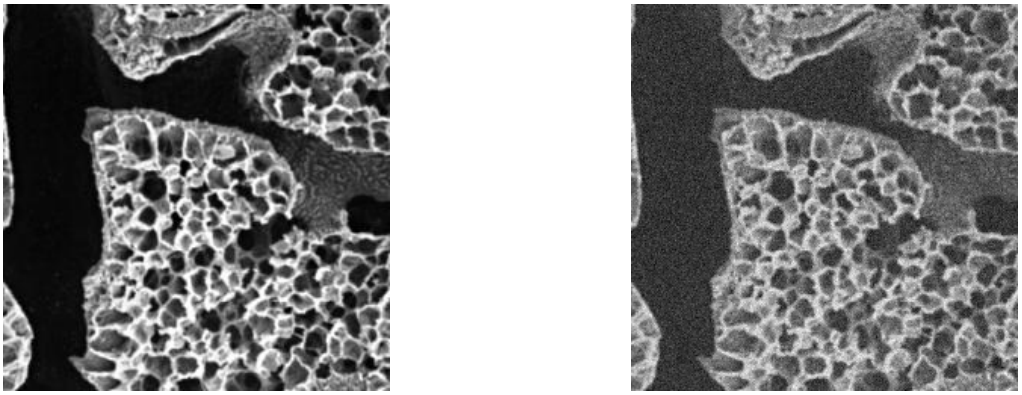
$$MSE = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(I_{orig}(i, j) - I_{denoised}(i, j))^2, \quad (23)$$

where I_{orig} indicates the original image and $I_{denoised}$ is the image after denoising. The lower this number, the better the effect of the denoising. Note that this measure does not necessarily imply that an image with a lower MSE is also more visually pleasing. We could also have used other measures for evaluating the performance of the denoising techniques such as the one norm, or the normalized MSE.

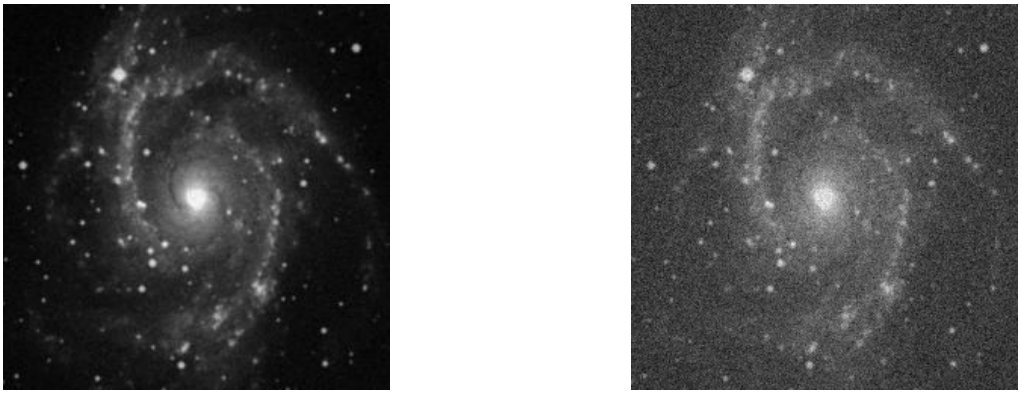
We first present the results of denoising with the synthetic test image. The MSE of the noisy image in Figure 4(b) is 401.16. Table 1 contains the mean-squared error (MSE) of the denoised image for variations in the conductance K and the type of spatial discretization (edge-centerd vs. node-centered). The results



(a) (b)



(c) (d)



(e) (f)

Figure 4. Original images ((a), (c), and (e)) and the corresponding images with additive Gaussian noise of standard deviation 20 ((b), (d), and (f))

Table 1. Mean-squared error for the synthetic test image in Figure 4(b): Effect of varying the conductance K for different diffusivities and spatial discretizations. See the text for the other parameters used in denoising. The MSE of the noisy image is 401.16.

	Diffusivity functions					
Discretization, K	Perona-Malik (1)	Perona-Malik (2)	Charbonnier	Weickert (m=2)	Weickert (m=3)	Weickert (m=4)
Node-centered, K=2.0	142.22	180.81	95.86	163.46	201.35	223.58
Node-centered, K=5.0	77.46	68.95	137.50	62.42	63.70	66.28
Node-centered, K=10.0	114.22	106.22	232.66	124.72	104.34	91.88
Node-centered, K=15.0	177.54	178.85	294.06	229.48	220.38	213.43
Edge-centered, K=2.0	124.59	150.47	106.28	136.89	160.89	175.02
Edge-centered, K=5.0	65.34	58.43	104.70	52.05	55.14	58.61
Edge-centered, K=10.0	99.90	80.20	178.57	90.16	61.50	52.73
Edge-centered, K=15.0	164.53	156.56	235.30	175.78	158.62	140.69

Table 2. Mean-squared error for the synthetic test image in Figure 4(b): Effect of varying the time-advancement schemes for different diffusivity functions and discretizations, with K=5.0. See the text for the other parameters used in denoising. The MSE of the noisy image is 401.16.

	Diffusivity functions					
Discretization, time advancement method	Perona-Malik (1)	Perona-Malik (2)	Charbonnier	Weickert (m=2)	Weickert (m=3)	Weickert (m=4)
Node-centered, explicit	77.46	68.95	137.50	62.42	63.70	66.28
Node-centered, ADI	79.76	71.20	127.29	63.64	65.06	67.79
Node-centered, AOS	81.29	72.86	123.12	64.99	66.60	69.55
Node-centered, AFI	99.34	72.37	214.58	59.11	58.62	63.97
Edge-centered, explicit	65.34	58.43	104.70	52.05	55.14	58.61
Edge-centered, ADI	66.28	59.19	98.99	52.31	55.12	58.35
Edge-centered, AOS	67.90	60.89	96.03	53.73	56.91	60.59
Edge-centered, AFI	70.37	44.97	165.45	41.41	41.87	43.34

are presented for six different diffusivity functions $c(x, y, t)$ described in Equations (4), (5), (7), and (8), with $m = 2, 3$, and 4. The time step-size for advancement of the PDE is set equal to 0.1 and the results are evaluated after 10 time steps. The explicit method is used for advancing from one time step to another. For the regularization in the calculation of the diffusivity, we set $\sigma_r = 1.0$. As mentioned, this regularization is also implemented by solving the diffusion equation (3), with explicit time advancement, with 10 time steps. The step-size can be calculated as we know the value of σ_r.

We make several observations based on this table. First, the edge-centered discretization is better than the node-centered discretization at reducing the noise in almost all the cases, regardless of the value of the conductance K and the diffusivity function. Second, the denoising, as expected, tends to be sensitive to the value of the conductance K. For this image, among the values of K considered, the best results are obtained for $K = 5.0$. However, given a value of K, and the discretization, it is much harder to select the best diffusivity function among those tested, as no function consistently gives the best performance.

Next, in Table 2, we consider the effects of the four time advancement schemes - explicit, ADI, AOS, and AFI, for the node-centered and edge-centered discretizations. In these experiments, we kept the value of the conductance K fixed at 5.0. All other parameters were as before. Again, we observe that the edge-centered discretization works better than the node-centered discretization. The performance of the time advancement methods varies with the diffusivity function, though the AFI scheme tends to perform well with the Weickert functions. Also, the Charbonnier function is not as good at reducing noise as the other functions.

In figure 5, panels (a) and (b), we show the effect of denoising the synthetic test image. Panel (a) is

Table 3. Mean-squared error for de-noising the lung image in Figure 4(d): Effect of varying the conductance K for different discretizations and diffusivities. See the text for the other parameters used in denoising. The MSE of the noisy image is 397.2.

	Diffusivity functions					
Discretization, K	Perona-Malik (1)	Perona-Malik (2)	Charbonnier	Weickert (m=2)	Weickert (m=3)	Weickert (m=4)
Node-centered, K=10.0	158.43	191.96	184.27	188.71	205.98	214.48
Node-centered, K=15.0	161.40	179.19	236.45	192.28	198.77	204.17
Edge-centered, K=10.0	140.10	149.53	163.76	154.81	158.73	162.01
Edge-centered, K=15.0	163.34	167.59	204.12	189.22	179.51	175.51

Table 4. Mean-squared error for de-noising the galaxy image in Figure 4(f): Effect of varying the conductance K for different discretizations and diffusivities. See the text for the other parameters used in denoising. The MSE of the noisy image is 400.25.

	Diffusivity functions					
Discretization, K	Perona-Malik (1)	Perona-Malik (2)	Charbonnier	Weickert (m=2)	Weickert (m=3)	Weickert (m=4)
Node-centered, K=10.0	40.61	41.75	42.77	41.74	43.31	44.60
Node-centered, K=15.0	41.14	42.54	46.51	45.79	45.51	45.47
Edge-centered, K=10.0	40.03	39.67	41.40	39.76	39.52	39.62
Edge-centered, K=15.0	41.46	42.55	44.23	45.93	45.00	44.37

the best result obtained so far, with the AFI time advancement scheme, edge-centered discretization, and the diffusivity function proposed by Weickert, with $m = 2$. It gives a denoised image with $MSE = 41.41$. For comparison, panel (b) is the denoised image taken with the same parameters, but with the second Perona-Malik filter (Equation (5)) and explicit time advancement. This has an $MSE = 58.43$.

In Tables 3 and 4, we present the corresponding results for the two scientific images, the lung image and the galaxy image. Based on our experiences with the synthetic image, we focused our experiments on the explicit time advancement method, varying the conductance K, the spatial discretization method (edge-centered vs. node-centered) and the diffusivity function. All the other parameters were set as before. As before, we observe that the edge-centered discretization tens to be better than the node-centered discretization and that there is no consistent winner among the diffusivity functions we tried. The best result for the lung image was obtained with the Perona-Malik (1) function, $K = 10.0$ and edge-centered discretization, with an $MSE = 140.10$. For the galaxy image, the best results were obtained with the Weickert ($m = 3$) function, edge-centered discretization, with $k = 10.0$, giving an $MSE = 39.52$. The corresponding images are given in Figure 5, panels (c) and (f).

For comparison, for the lung and galaxy images, we also applied our wavelet de-noising software.[1] We used the Symmlet12 wavelet, with three levels of wavelet decomposition, and periodic boundary conditions. The best results were obtained using the SURE thresholding, applied at a pyramid level, resulting in $MSE = 158.43$ and $MSE = 52.33$ for the lung and the galaxy images, respectively. The corresponding images are given in Figure 5, panels (d) and (g). Panels (e) and (h) are the result of denoising using a combination of simple filters (MMSE followed by a Gaussian, both with a filter size of 3). We observe that the visual quality of the images denoised using isotropic diffusion, wavelet-based thresholding, and spatial filters is very similar. However, an appropriate choice of parameters for the diffusion method results in images with slightly lower MSE than the best options with the other methods.

4. CONCLUSIONS AND FUTURE WORK

Our empirical study with non-linear isotropic diffusion-based denoisingwith scientific images results in several observations. First, these techniques can result in a lower MSE when compared with the best methods based on wavelet thresholding and spatial filters. Like the wavelet-based methods, they too have several parameters

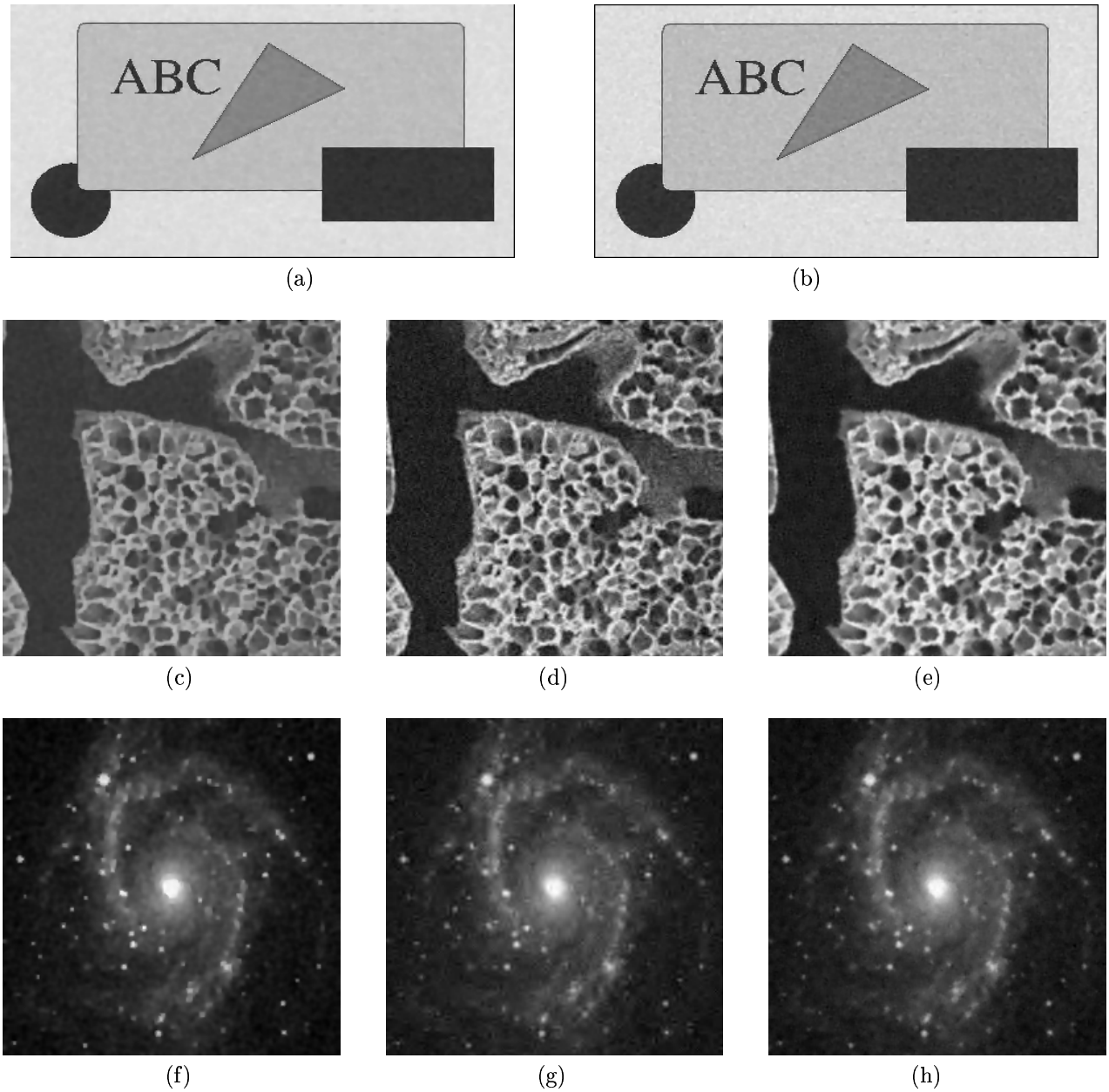


Figure 5. Denoised images: (a) Synthetic image, MSE 41.41, with Weickert diffusivity $m = 2$, AFI time advancement; (b) Synthetic image, MSE 52.05, with Weikert diffusivity, $m = 2$, explicit time advancement; (c) Lung image: best result with diffusion-based denoising, MSE=140.10; (d) Lung image: best result with wavelet-based denoising, MSE=158.43; (e) Lung image: best result with MMSE(3) filter followed by Gaussian (3) filter, MSE = 193.00; (f) Galaxy image: best result with diffusion-based denoising, MSE=39.52; (g) Galaxy image: best result with wavelet-based denoising, MSE=52.33; (h) Galaxy image: best result with MMSE(3) filter followed by Gaussian (3) filter, MSE = 43.84

that must be appropriately chosen to give good performance. For the images we tried, the edge-centered discretization for the diffusivity was consistently better than the node-centered one. However, it was difficult to find a diffusivity function and time advancement method that consistently gave better results. In addition, the conductance parameter K in the diffusivity is image dependent and must be chosen carefully. Like the spatial filters, the diffusion-based denoising techniques do not require that the image size be a power of two, a constraint that is imposed by the non-decimated implementations of the wavelet-based methods.

Our initial study has demonstrated that diffusion-based techniques can be effective at denoising. There are several ways in which we plan to extend this work. First, we would like to further study the effect of parameters such as the time duration for which the PDE is allowed to evolve, the effect of making both σ_r and K vary with time, a finer control over the regularization of the gradient in the diffusivity, etc. In addition, we would like to explore the use of anisotropic diffusion, especially for the case where the image has fine structure.

5. ACKNOWLEDGEMENTS

UCRL-JC-145574 - This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

REFERENCES

1. I. Fodor and C. Kamath, "On denoising images using wavelet-based statistical techniques," Technical report UCRL-JC-142357, Lawrence Livermore National Laboratory, 2001. http://www.llnl.gov/casc/sapphire/.
2. I. K. Fodor and C. Kamath, "Denoising through wavelet shrinkage: An empirical study," Technical report, UCRL-JC-144258, Lawrence Livermore National Laboratory, 2001. Submitted for publication, July 2001.
3. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," in *Proceedings, IEEE Computer Society workshop on Computer Vision*, pp. 16–27, 1987.
4. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(7), pp. 629–639, 1990.
5. R. Malladi and J. Sethian, "A unified approach to noise removal, image enhancement, and shape recovery," *IEEE Trans. Image Processing* **5**, pp. 1154–1168, 1996.
6. M. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Processing* **7**, 1998.
7. J. Weickert, B. ter Haar Romeny, and M. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Trans. Image Processing* **7**, 1998.
8. P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proceedings, IEEE International Conference on Image Processing*, **2**, pp. 168–172, 1994.
9. L. Blanc-Feraud, P. Charbonnier, G. Aubert, and M. Barlaud, "Non-linear image processing: modeling and fast algorithm for regularization with edge detection," in *Proceedings, IEEE International Conference on Image Processing*, **1**, pp. 474–477, 1995.
10. J. Weickert, *Anisotropic diffusion in Image Processing*, B.G. Teubner, 1998.
11. R. Whitaker and S. M. Pizer, "A multi-scale approach to non-uniform diffusion," *CVGIP:Image Understanding* **57**, pp. 99–110, 1993.
12. X. Li and T. Chen, "Nonlinear diffusion with multiple edginess thresholds," *Pattern Recognition* **27**(8), pp. 1029–1037, 1994.
13. V. Vemuri and W. J. Karplus, *Digital Computer Treatment of Partial Differential Equations*, Prentice Hall, 1981.
14. D. Barash and R. Kimmel, "An accurate operator splitting scheme for nonlinear diffusion filtering," technical report, hpl-2000-48 (r.1), Hewlett-Packard Laboratories, Israel, 2000.
15. L. Lapidus and G. F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, John Wiley, 1982.

University of California
Lawrence Livermore National Laboratory
Technical Information Department
Livermore, CA 94551