Edge-Preserving Noise Removal, Part I: Second-Order Anisotropic Diffusion*

Seongjai Kim†

Abstract

The article is concerned with efficient edge-preserving numerical techniques for noise removal via anisotropic diffusion. New minimum-biased (MB) finite difference formulas are introduced to minimize diffusion on both the piecewise smooth portions and their boundaries. Locally one-dimensional time-stepping algorithms are analyzed and a formula for efficient timestep sizes is suggested, to remove the high-frequency components of the error (the noise) more efficiently and to minimize the torture of lower-frequency components of the image. An effective strategy is suggested for an automatic stopping of the diffusion process. It is numerically verified that the MB formulas eliminate, rather than diffuse, the noise of piecewise constant images. With such a property, the MB formulas are successfully applied to noise removal of general images. Numerical examples carried out with various images are presented to demonstrate superior properties of (a) the new MB formulas, (b) the formula for timestep sizes, and (c) the automatic stopping strategy of the diffusion process.

Key words. Noise removal, anisotropic diffusion, minimum-biased finite difference, locally one-dimensional time-stepping algorithm.

*Technical Report #2001-09, Department of Mathematics, University of Kentucky
†Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027 USA
Email: skim@ms.uky.edu. The work is supported in part by NSF grant DMS-0107210.
Contents

1. Introduction 3

2. Preliminaries 4
   2.1. Linearized time-stepping procedures .......................... 4
   2.2. Edge-stopping anisotropic diffusion equations ................. 5

3. Finite Difference Schemes 9

4. Locally One-Dimensional Methods 12
   4.1. The ADI method .............................................. 12
   4.2. The FS method .............................................. 13
   4.3. The AOS method ............................................. 14

5. Efficiency of LOD Methods 14

6. Numerical Experiments 18
   6.1. Piecewise constant images ................................... 19
   6.2. General images ............................................... 23

7. Discussion 28

8. Conclusions 30
1. Introduction

Anisotropic diffusion has been a popular tool for noise removal since the first elegant formulation by Perona and Malik [14]. A considerable amount of research has been carried out for the theoretical and computational understanding of the method and related methods for image enhancement [1, 2, 3, 11, 12, 15, 22]; a good reference to work on them is Sapiro [16].

Consider the following nonlinear anisotropic diffusion equation [1]

$$\frac{\partial u}{\partial t} = \kappa(u) |\nabla u|, \quad \kappa(u) = \nabla \cdot \left( \frac{\nabla u}{\|\nabla u\|} \right),$$  \hspace{1cm} (1.1)

where \((\mathbf{x}, t) \in \Omega \times J\), \(\Omega\) is a rectangular domain and \(J = (0, T)\) for some \(T > 0\), and \(\kappa(u)\) denotes the curvature. Here both \(|\cdot|\) and \(\|\cdot\|\) denote the magnitude (the same in definition), but we distinguish them to indicate separately. The function \(u\) is initialized by the original (noisy) image \(u(x, t = 0)\), normalized to have its values between zero and one, and the Neumann boundary condition is imposed. The above curvature-driven equation (1.1) can be viewed as a geometric version of the heat equation (or a nonlinear heat equation). Due to the diffusion-like term, large oscillations are smeared out quickly; it has been utilized for noise removal and image enhancement.

However, it should be noticed that the anisotropic diffusion is not only smearing out the noise but also blurring images, in particular, the interfaces (edges) of piecewise smooth parts (faces). In numerical approximation of (1.1), special cares have to be taken to minimize the diffusion across the edges. In particular, the gradient magnitude \(|\nabla u|\) must be carefully discretized to serve as an edge-stopper.

In this article, we study efficient edge-stopping numerical methods for noise removal, based on (1.1) and focusing on the following important issues:

- **Edge-stopping difference formulas**: It is hard to design robust edge-stopping formulas for general noisy images. However, we introduce finite difference (FD) formulas that can preserve edges of piecewise constant images precisely when no noise is added to the image. Utilizing such FD formulas, one may hope that the noise in general images can be removed with the edges better preserved.

- **Computational efficiency**: Efficiency is an important factor in designing numerical algorithms for noise removal, in particular, for stationary solutions and/or 3D images. Efficiency is viewed as an ability of the algorithm to eliminate the noise in a certain speed; it is often the case that more efficient algorithms can restore more accurate images.

- **Automatic stop of the diffusion process**: In practice, it is interesting to give a criterion for an automatic stop of the diffusion iteration.
The article is organized as follows. In §2, we present preliminaries on formal linearization procedures for (1.1) and edge-stopping anisotropic diffusion equations. §3 introduces new edge-stopping FD formulas that preserve the boundaries of piecewise constant images. In §4, we review three different diffusion iterations: the alternating direction implicit (ADI) method, the fractional step (FS) method, and the additive operator splitting (AOS) method. In §5, we discuss strategies for efficient timestep sizes. The timestep size turns out to affect both efficiency and accuracy of the restored images. In §6, we report some of numerical experiments on synthetic and Lena images. In the same section, a strategy for automatic stop of the diffusion iteration is suggested. §7 contains discussions on other efficient edge-stopping strategies for general images. The last section includes conclusions.

2. Preliminaries

2.1. Linearized time-stepping procedures

As a time-stepping procedure of the numerical solution of (1.1), we can adopt either the incomplete (linearized) backward-Euler method

\[ \frac{u^n - u^{n-1}}{\Delta t^n} + A^{n-1/2} u^n = 0 \]  \hspace{1cm} (2.1)

or the incomplete Crank-Nicolson scheme

\[ \frac{u^n - u^{n-1}}{\Delta t^n} + \frac{1}{2} A^{n-1/2} (u^n + u^{n-1}) = 0, \] \hspace{1cm} (2.2)

where \( \Delta t^n = t^n - t^{n-1} \), \( A^{n-1/2} = A_1^{n-1/2} + A_2^{n-1/2} \), with

\[ A_{i/2}^{n-1/2} u^n = -|\nabla_h u^{n-1/2}| D_{x_i} \left( \frac{D{x_i} u^n}{\|\nabla_h u^{n-1/2}\|} \right), \quad \ell = 1, 2, \]

and \( \nabla_h = (D_{x_1}, D_{x_2})^T \) is a proper difference operator for the gradient. The nonlinear terms can be approximated by either the solution in the previous timestep

\[ u^{n-1/2} = u^{n-1} \] \hspace{1cm} (2.3)

or the extrapolation given as

\[ u^{n-1/2} = (1 + \gamma) u^n - \gamma u^{n-2}, \quad \gamma = \frac{\Delta t^n}{2 \Delta t^{n-1}}. \] \hspace{1cm} (2.4)

When \( \Delta t^n \) is much larger than \( \Delta t^{n-1} \), the extrapolation may not be stable; in practice, \( \gamma \) should be modified as, e.g.,

\[ \gamma = \min \left( \frac{1}{2}, \frac{\Delta t^n}{2 \Delta t^{n-1}} \right). \]
The algebraic system to be solved in each time level is of the form

\[(1 + \tau A^{n-1/2}) u^n = b, \quad (2.5)\]

where \(\tau\) is either \(\Delta t^n\) or \(\Delta t^n/2\) for (2.1) and (2.2), respectively, and \(A^{n-1/2}\) is a banded matrix corresponding to the five-point difference stencil. Since the image is mostly defined on a rectangular domain, the algebraic system can be solved efficiently by applying one of locally one-dimensional (LOD) perturbation methods to be considered in Sections 4 and 5.

2.2. Edge-stopping anisotropic diffusion equations

Various anisotropic/nonlinear diffusion equations incorporating edge-stopping functions have been suggested to stop or minimize the diffusion across the edges; see Sapiro [16, Ch.4] and references therein. The model studied by Perona and Malik [14] is

\[\frac{\partial u}{\partial t} - \nabla \cdot \left( g(|\nabla u|) \nabla u \right) = 0, \quad (2.6)\]

where \(g\) is a function, \(g(x) \geq 0\), having the property: \(g(x) \to 0\) as \(x \to 0\). For example, one may choose one of the following

\[g(x) = \frac{1}{1 + x^2/\sigma^2}, \quad g(x) = e^{-x^2/\sigma^2}, \quad \sigma > 0, \quad (2.7)\]

as suggested in [14]. The equation (2.6) has motivated a large number of researchers to study the mathematical properties of the kind, numerical schemes, and applications [1, 2, 14]. Note that

\[-\nabla \cdot \left( g(|\nabla u|) \nabla u \right) = -\nabla g(|\nabla u|) \cdot \nabla u - g(|\nabla u|) \Delta u \quad (2.8)\]

and \(g(|\nabla u|)\) has its minimum on the edges of the image. The first term in the right side of (2.8) shows a self-focusing characteristics (an inverse diffusion) on the edges, while the second term diffuse the least near the edges. This explains an edge-stopping property of (2.6). However, it should be noticed that the overall action is diffusion.

To minimize the diffusion across the edges, one may discretize the gradient in \(g(|\nabla u|)\) of (2.6) utilizing one-sided differences as follows [14]:

\[- \nabla \cdot \left( g(|\nabla u^{n-1/2}|) \nabla u^n \right)_{i,j} \approx \sum_{(\ell,m) \in N^{(4)}_{i,j}} g(|u^{n-1/2}_{i,j} - u^{n-1/2}_{\ell,m}|) (u^n_{i,j} - u^n_{\ell,m}), \quad (2.9)\]

where \(N^{(4)}_{i,j}\) denotes the four adjacent points of \((i,j)\) corresponding to the five-point difference stencil.
Figure 1: Performance of (2.6) with $g(x) = e^{-x^2/\sigma^2}$. (a) the original image in a 100 $\times$ 100 grid mesh, (b) a 10% noisy image, (c) the first iterate with $\sigma = 0.5$, and (d) the 50th iterate with $\sigma = 0.2$. 
Figure 2: Performance of (2.6) with $g(x) = e^{-x^2/\sigma^2}$ for the Lena image. (a) the original Lena image on a $512 \times 512$ mesh, (b) a $20\%$ noisy image, (c) $50$ ADI iterations with $\sigma = 0.05$ for the $20\%$ noisy Lena image, and (d) $72$ ADI iterations with $\sigma = 0.01$ for the original Lena image (no noise is added).
In Figure 1, we present the performance of (2.6) with \( g(x) = e^{-x^2/\sigma^2} \) approximated by (2.9), for the 10% noisy image. The original image contains a house, a square, thin curves, and thick curves (of thickness three). The values for the house, the square, and the curves are 0.33 (in average), 0.7, and 1.0, respectively. For the noise, 10% of the cell values are replaced by computer-generated random numbers (scaled to be between zero and one). For the iteration, the ADI method is utilized; see (4.2) below. When \( \sigma = 0.5 \), one ADI iteration has diffused pretty much of the noise and edges of small jumps. For \( \sigma = 0.2 \), the noise still remains in a considerable amount in 50 iterations, while the house (of which the edges have small jumps) is completely smeared out.

The nonlinear diffusion (2.6) is applied to the Lena image, as shown in Figure 2. In 50 iterations with \( \sigma = 0.05 \) for the 20% noisy Lena image, flat portions of the image become flatter, while large jumps of both the edges and the noise are not smeared out. To check the flattening characteristics of (2.6), we apply 72 iterations with a sufficiently small parameter \( (\sigma = 0.01) \) to the original Lena image (of no noise). As one can see from Figure 2(d), the restored image shows flattened portions.

For the edge-stopping functions \( g \) in (2.7), the parameter \( \sigma \) can be better determined based on statistical properties of the images; its choice is often problematic. It should be sufficiently small to virtually stop diffusion across the edges, because otherwise edges of small jumps would smear out rapidly, as shown in Figures 1 and 2. Given \( \sigma \) properly chosen, the diffusion is ignorable at points where the gradient magnitude is large, e.g., the edges and the noise. However, the diffusion would be large where the gradient magnitude is small. As a consequence, the diffusion process makes relatively flat portions of the image flatter. Furthermore, since the noise is hardly smooth, its gradient magnitude is likely large; the noise removal process incorporating (2.6) can become slow and require a large number of iterations. In summary: the nonlinear diffusion (2.6) is hard to satisfy the following ideal properties except the last one:

\[
(P) \left\{ \begin{array}{l}
(a) \text{Fast reduction of the noise,} \\
(b) \text{Negligible diffusion at the faces, and} \\
(c) \text{Negligible diffusion at the edges.}
\end{array} \right.
\]

As an alternative, consider edge-stopping functions \( g = G \) having the properties:

\[
G(0) = 0, \quad \text{and } G(x) \to \infty \text{ as } x \to \infty. \quad (2.10)
\]

Then, the nonlinear diffusion (2.6) incorporating (2.10) would easily satisfy the first two properties in (P). For the third property, one may introduce edge-stopping FD formulas, which is one of the main concerns in this article. (See §3.) The basic idea is that we approximate the gradient magnitude \( |\nabla u| \) by one-sided FD schemes over the whole domain. Consider a piecewise constant areal image. Then the gradient mag-
nitude obtained from the one-sided differences becomes completely zero, if no noise is added. When an isolated grid value of \( u \) shows a large difference from neighboring points, the one-sided FD would be large at the point and therefore the isolated value must be quickly eliminated by the diffusion process. The above argument can be applied for various images, in practice.

The model in (1.1) is chosen to easily satisfy the properties in (P) when \( |\nabla u| \) is approximated by one-sided FD formulas, while \( \|\nabla u\| \) is discretized by the central FD scheme with its singularities regularized by a positive constant, that is,

\[
\|\nabla u\| \approx \sqrt{\text{max}(\varepsilon^2, u_x^2 + u_y^2)}, \quad \varepsilon > 0.
\]  

(2.11)

In this article, we set \( \varepsilon = 0.2 \) for all examples incorporating (1.1).

3. Finite Difference Schemes

We consider \( |\nabla u| \) in (1.1) as an edge-stopper (and also as a quick noise-remover). When the gradient is approximated by the central scheme, it would be large on the edges and therefore diffusion happens much on the edges. To overcome the difficulties, this section considers one-sided FD schemes for \( |\nabla u| \). Here the main purpose is to introduce FD schemes that preserve perfectly piecewise constant images when no noise is added.

We first present the central (second-order) FD scheme:

\[
(u_x)_{i,j} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2}, \quad (u_y)_{i,j} \approx \frac{u_{i,j+1} - u_{i,j-1}}{2},
\]  

(3.1)

where we have set \( \Delta x = \Delta y = 1 \). The central scheme (3.1) would be utilized for \( \|\nabla u\| \) in (1.1). However, it is not appropriate to approximate \( |\nabla u| \) on the edges. Now, we may consider the one-sided FD scheme defined as follows:

\[
| (u_x)_{i,j} | \approx \min\left( |u_{i+1,j} - u_{i,j}|, |u_{i,j} - u_{i-1,j}| \right),
\]

\[
| (u_y)_{i,j} | \approx \min\left( |u_{i,j+1} - u_{i,j}|, |u_{i,j} - u_{i,j-1}| \right).
\]  

(3.2)

The one-sided scheme may be better than the central scheme in edge-stopping for \( |\nabla u| \). All piecewise constant areal faces would be preserved except corners and vertical/horizontal tips of the edges. But it still can smear out actively some of main features in the texture (e.g., thin curves).

Then, which schemes can preserve piecewise constant textures during the diffusion process? To answer the question, we begin with some observations. For simplicity, let the image contain two colors: black \((u = 1)\) and white \((u = 0)\). Then, the face in the image is composed of the grid cells that are of the same value and distributed nearby in the vertical, horizontal, or 45° directions. Any isolated cells can be viewed
as noise. It is easy to see that the isolated cells show high frequency components of the image. During the diffusion process, such isolated cells can be quickly diffused either to disappear or to form lower (thus, medium-high) frequency components of the noise particularly when the noise level is high. For the purpose of the development of FD schemes (in this section) and efficient diffusion procedures (to be considered in the next section), we may summarize the observations as follows:

(Obs1). The faces in the image are a combination of vertical, horizontal, and 45° line segments.

(Obs2). Noise is the error showing medium-high to high frequencies.

Based on (Obs1), the edge-preserving FD scheme at the point \((i, j)\) should be defined with a full consideration of the eight one-sided differences:

\[
\frac{|u_{\ell,m} - u_{i,j}|}{\sqrt{(\ell - i)^2 + (m - j)^2}}, \quad (\ell, m) \in [i-1, i+1] \times [j-1, j+1], \quad (\ell, m) \neq (i, j).
\]

We order the above differences as

\[D_{i,j,1} \leq D_{i,j,2} \leq \cdots \leq D_{i,j,8}.\]

Then, an edge-preserving scheme for the gradient magnitude can be defined as

\[
(|\nabla u|)_{i,j} \approx \sqrt{D_{i,j,1}^2 + D_{i,j,2}^2},
\]

(3.3)

In this article, we call (3.3) the minimum-biased finite difference (MB-FD) scheme. MB-FD is not an accurate approximation of the gradient; however, it preserves all piecewise constant faces (except the end points of thin curves). When all piecewise constant features must be perfectly preserved, one may adopt

\[
(|\nabla u|)_{i,j} \approx D_{i,j,1},
\]

(3.4)

which we call the minimum slope (Min-Slope) scheme. Min-Slope holds very tightly piecewise constant features of the image. It shows excellent properties in noise removal for piecewise constant images, but it must be applied with a caution for general images. See §6.

Remark. The combined magnitude of MB-FD/Min-Slope for \(|\nabla u|\) and the central FD scheme for \(||\nabla u||^{-1}\) (regularized as in (2.11)) is not larger than unity at most points of piecewise smooth images; it can be larger than one particularly where the image contains high frequency components such as noise. For piecewise constant images of a moderate noise level, MB-FD/Min-Slope would return zero for \(|\nabla u|\) at most grid points, while they are large at points of the noise. On the other hand, the
Figure 3: Images in one ADI iteration for various FD schemes for the piecewise constant image in Figure 1. (a) the central scheme, (b) the one-sided scheme, (c) MB-FD, and (d) Min-Slope.
central scheme applied to $\|\nabla u\|$ has relatively larger values at the edges. Thus the overall effects of such combinations are (a) a quick reduction of the noise and (b) an efficient edge-stopping.

Figure 3 shows restored images with various FD schemes applied to $|\nabla u|$ of (1.1), for the piecewise constant image in Figure 1. One iteration of ADI (4.2) is carried out with $\Delta t = \sqrt{2} \approx 1.414$. The one-sided scheme preserves the areal faces, but it does not show any ability to keep thin features as expected. MB-FD has removed parts of thin curves due to effects of the noise, but it eliminates the noise almost completely in one iteration and restores areal features of the image quite well. The Min-Slope preserves areal features and most parts of thin curves excellently, but it leaves parts of the noise in a noticeable amount yet. It should be noticed that the noise is diffusing with the central scheme, while the other schemes eliminate it! It is easy to see that in a single iteration with MB-FD and Min-Slope, an isolated value located at an interior point of a piecewise constant portion of the image is replaced by the average of neighboring values with no diffusion, which is a perfect elimination of the noise.

In summary: For piecewise constant images, MB-FD and Min-Slope have shown (a) the preservation of both areal and thin features and (b) a quick elimination, rather than diffusion, of the noise. Such advantageous properties can minimize image blur during the noise removal for general images. That is, the anisotropic diffusion (1.1) incorporating MB-FD and Min-Slope can better satisfy all the properties in (P) of §2.2, which will be numerically verified in §6.

4. Locally One-Dimensional Methods

Images are mostly given in a rectangular mesh; locally one-dimensional (LOD) methods can be applied, with a great efficiency, to noise removal via anisotropic diffusion. In this section, we review LOD computational algorithms such as alternating direction implicit (ADI) method [4, 6, 13], the fractional step (FS) method [7, 10, 20, 21], and the additive operator splitting (AOS) method [19]. These methods are efficient; in each half of the calculation in a time level, the matrix to be inverted is tridiagonal, so that they require $O(N := n_x n_y n_t)$ flops in total, where $n_x$, $n_y$, and $n_t$ respectively denote the number of grid points in $x$-, $y$-, and $t$-directions.

4.1. The ADI method

The ADI method was first introduced in three papers [4, 6, 13] by Douglas, Peaceman, and Rachford, to solve the heat equation. Consider the following ADI method
for solving (2.2):
\[
\begin{align*}
\frac{u^n - u^{n-1}}{\Delta t} + \frac{1}{2} A_1 u^* + \frac{1}{2} A_2 u^{n-1} &= 0, \\
\frac{u^* - u^{n-1}}{\Delta t} + \frac{1}{2} A_1 u^* + \frac{1}{2} A_2 u^n &= 0,
\end{align*}
\] (4.1)

where \( u^* \) is an intermediate solution. Here we have dropped the superscripts \( n \) on \( \Delta t \) and \( (n - 1/2) \) on \( A_\ell, \ell = 1, 2 \), for a simpler presentation. By eliminating \( u^* \), one can rewrite (4.1) as
\[
\frac{u^n - u^{n-1}}{\Delta t} + \frac{1}{2} A (u^n + u^{n-1}) + \frac{\Delta t}{4} A_1 A_2 (u^n - u^{n-1}) = 0.
\]

Since \( (u^n - u^{n-1}) = \mathcal{O}(\Delta t) \), the above equation is a perturbation of the Crank-Nicolson difference equation (2.2), by a splitting error of \( \mathcal{O}(\Delta t^2) \), which is the same order as the temporal truncation error already introduced during the discretization. Algorithm (4.1) can be efficiently implemented in the following form
\[
(\text{ADI}) \left\{ \begin{array}{l}
(1 + \frac{\Delta t}{2} A_1) u^* = (1 - \frac{\Delta t}{2} A_1 - \Delta t A_2) u^{n-1}, \\
(1 + \frac{\Delta t}{2} A_2) u^n = u^* + \frac{\Delta t}{2} A_2 u^{n-1}.
\end{array} \right.
\] (4.2)

ADI can also be derived as a perturbation of the backward-Euler method (2.1), for which the splitting error is again \( \mathcal{O}(\Delta t^2) \).

4.2. The FS method

The fractional step (FS) procedures, introduced by the Russian mathematicians D'yanov, Marchuk, and Yanenko [7, 10, 20, 21], can be formulated as a perturbation of either the backward-Euler method (2.1) or the Crank-Nicolson method (2.2), with a splitting error of \( \mathcal{O}(\Delta t) \) for both cases. We present the FS method derived from the backward-Euler method:
\[
\begin{align*}
\frac{u^* - u^{n-1}}{\Delta t} + A_1 u^* &= 0, \\
\frac{u^n - u^*}{\Delta t} + A_2 u^n &= 0,
\end{align*}
\] (4.3)

which is equivalent to solving
\[
\frac{u^n - u^{n-1}}{\Delta t} + A u^n + \Delta t A_1 A_2 u^n = 0.
\]

For an implementation purpose, we rewrite the algorithm (4.3) as
\[
(\text{FS}) \left\{ \begin{array}{l}
(1 + \Delta t A_1) u^* = u^{n-1}, \\
(1 + \Delta t A_2) u^n = u^*.
\end{array} \right.
\] (4.4)

This algorithm requires the least operations among all LOD methods considered in this article.
4.3. The AOS method

Recently, the additive operative splitting (AOS) method was suggested as an efficient numerical method in image processing by Weickert and his colleagues [19]:

\[
\frac{u^n - u^{n-1}}{\Delta t} + 2A_1 u^* = 0, \\
\frac{u^{**} - u^{n-1}}{\Delta t} + 2A_2 u^{**} = 0, \\
u^n = (u^* + u^{**})/2. 
\] (4.5)

When \( \Delta t \) is sufficiently small, the above algorithm solves

\[
\frac{u^n - u^{n-1}}{\Delta t} + A u^{n-1} - 2\Delta t (A_1 A_1 + A_2 A_2) u^{n-1} + \mathcal{O}(\Delta t^2) \\
= \frac{u^n - u^{n-1}}{\Delta t} + A u^n + A (u^n - u^{n-1}) \\
+ 2\Delta t (A_1 A_2 + A_2 A_1) u^{n-1} + \mathcal{O}(\Delta t^2) = 0.
\]

Thus it is an \( \mathcal{O}(\Delta t) \)-perturbation of the backward-Euler scheme (2.1). It can be implemented in the following form

\[
(AOS) \begin{cases} 
(1 + 2\Delta t A_1) u^* = u^{n-1}, \\
(1 + 2\Delta t A_2) u^{**} = u^{n-1}, \\
u^n = (u^* + u^{**})/2. 
\end{cases} 
\] (4.6)

Note that the \( x \)- and \( y \)-sweeps are independent on each other. When the parallelization efficiency matters, it can be advantageous over other LOD methods, as claimed in [19].

Remark. FS and AOS are easier to implement and a few per cent faster than ADI. But they are first-order accurate and introduce a large error unless the temporal step size is sufficiently small. One might claim that in noise removal, accuracy of iterates (intermediate solutions) is not important. However, in the view point that the noise is an error to be eliminated, efficiency and the quality of the restored images may not be satisfactory unless the numerical error is controlled reasonably well. As shown in §5, ADI is much easier to control the rate of error reduction than FS and AOS. ADI turns out to restore images with the best clarity and efficiency (3-5 time faster than FS and AOS); see Figure 5 and Table 1 below. See [5] for various discussions and strategies for a virtual elimination of the splitting error of the ADI and FS methods applied to the heat equation and the second-order wave equation.

5. Efficiency of LOD Methods

It is easy to see that the practical efficiency of noise removal via the anisotropic diffusion (1.1) strongly depends on the choices of temporal step size \( \Delta t^n \). In this
section, we present strategies to choose $\Delta t^n$ for an efficient computation.

We begin with the ADI method (4.2), which equivalently reads

\[
\begin{align*}
\left(1 + \frac{\Delta t}{2} A_1 \right) u^* &= \left(1 - \frac{\Delta t}{2} A_2 \right) u^{n-1}, \\
\left(1 + \frac{\Delta t}{2} A_2 \right) u^n &= \left(1 - \frac{\Delta t}{2} A_1 \right) u^*.
\end{align*}
\]

(5.1)

When (5.1) is viewed as an algebraic system, the number "1" must be considered as the identity matrix. Let $u_h$ be the stationary solution of (5.1) and define the error in the $n$th iterate as

\[
E^n = u_h - u^n.
\]

Then it follows from (5.1) that

\[
E^n = T_{ADI} E^{n-1},
\]

(5.2)

where

\[
T_{ADI} = \left(1 + \frac{\Delta t}{2} A_2 \right)^{-1} \left(1 + \frac{\Delta t}{2} A_1 \right)^{-1} \left(1 - \frac{\Delta t}{2} A_1 \right) \left(1 - \frac{\Delta t}{2} A_2 \right).
\]

The convergence of (5.1) is equivalent to that the spectral radius of $T_{ADI}$ is less than unity, i.e., $\rho(T_{ADI}) < 1$, which can be easily proved when $A_1, \ell = 1, 2$, are positive indefinite. Here our purpose is to choose $\Delta t$ such that the algorithm can efficiently reduce the error; particularly, its high-frequency components. Note that the noise to be removed through the nonlinear diffusion is most likely high-frequency components of the error.

When the image is locally linear, the action of (1.1) on the image is the same as that of the heat equation except at the edges. For an easier analysis, we begin with the algorithm for the heat equation. Then $A_\ell, \ell = 1, 2$, denote FD approximations of $-\partial^2 / \partial x^2$ and $-\partial^2 / \partial y^2$, respectively. Let the number of grid points in the $x$-direction is the same as that in the $y$-direction, i.e., $m := n_x = n_y$. Then $A_1 = P A_2 P^T$, for a permutation matrix $P$; let their eigenvalues be $\sigma(A_1) = \sigma(A_2) = \{\lambda_1 \leq \cdots \leq \lambda_m\}$:

\[
A_1 \mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad i = 1, \cdots, m,
\]

where $\mathbf{v}_i$ are the corresponding eigenvectors of $A_1$. Then the eigenvalues of $T_{ADI}$ reads

\[
\sigma(T_{ADI}) = \left\{ \frac{1 - \frac{\Delta t}{2} \lambda_i}{1 + \frac{\Delta t}{2} \lambda_i} : \frac{1 - \frac{\Delta t}{2} \lambda_j}{1 + \frac{\Delta t}{2} \lambda_j} \right\} : i, j = 1, \cdots, m.
\]

Since we are interested in reducing high frequency components (corresponding to large eigenvalues of $A_1$) of the error, we may choose $\Delta t$ as the solution of the min-max problem

\[
\min_{\Delta t} M(\Delta t), \quad (5.3)
\]
where
\[ M(\Delta t) := \max_{i,j=i_0,\ldots,m} \left| \frac{1 - \frac{\Delta t}{2} \lambda_i}{1 + \frac{\Delta t}{2} \lambda_i} \cdot \frac{1 - \frac{\Delta t}{2} \lambda_j}{1 + \frac{\Delta t}{2} \lambda_j} \right|, \]
for some \( i_0 \gg 1 \).

For example, to reduce the very high-frequency components of the error efficiently, one may choose \( \Delta t = 2/\lambda_m \). In the case, the highest-frequency component of the error disappears in one iteration and adjacent components would be reduced significantly. However, in the practical computation, we do not know the eigenvalues of the anisotropic diffusion matrices, \( A_1 \) and \( A_2 \), and finding their large eigenvalues would be more expensive than carrying out the iteration with a rough parameter. Furthermore, the noise often corresponds to not only the highest-frequency but also medium-high frequencies; see (Obs2) in §3. Also, it should be noticed that the real computation power of the ADI method (4.2) for stationary solutions comes from a cyclic application of the parameters \( \Delta t \); see [18] for optimum ADI parameters for the cycle length of the form \( 2^k, k \geq 0 \).

Then, how can we find a set of ADI parameters for an efficient reduction of medium-high to high frequency components of the error for the anisotropic diffusion? When the ADI method is implemented by setting \( \Delta x = \Delta y = 1 \), the smallest and largest eigenvalues of \( A_1 \) are in \( O(h^2) \) and \( O(1) \), respectively, where \( h = 1/m \) and \( m \) is the number of grid points in the \( x \)-direction. The upper bound of the eigenvalues of \( A_1 \) can be found from the inequality

\[ \rho(A_1) \leq \|A_1\|_{\infty} = \max_{1 \leq i \leq m} \sum_{j=1}^{m} |a_{ij}|, \]

where \( A_1 = (a_{ij}) \) and \( \cdot \|_{\infty} \) denotes the matrix \( \infty \)-norm.

From various experiments, we have found that a single \( (k = 0) \) or double \( (k = 1) \) parameters are efficient enough and easy to implement for ADI. We will try to seek the parameters that can rapidly reduce the error components in the following interval of frequencies

\[ I_{\text{noise}} := (\alpha_0, \beta_0), \quad (5.4) \]

where
\[ \alpha_0 \in [\alpha_{0s}, \alpha_{0s}^*] := \left[ (\pi h/2) \|A_1\|_{\infty}, (\pi h/2)^{1/2} \|A_1\|_{\infty} \right], \]
\[ \beta_0 = \|A_1\|_{\infty}. \]

Here the interval for \( \alpha_0 \) is not only from experiments but from theory of optimal ADI parameters. When the heat equation is to be solved, the minimum and maximum eigenvalues of \( A_1 \) are approximately \( \pi^2 h^2 \) and \( 4 \), respectively. For the anisotropic diffusion problem, we can consider \( \|A_1\|_{\infty} \) as a scaling factor and we may assume that \( \lambda_1 = \pi^2 h^2 \|A_1\|_{\infty}/4 \) and \( \lambda_m = \|A_1\|_{\infty} \), the smallest and largest eigenvalues of \( A_1 \). Note that \( \alpha_{0s} = (\lambda_1 \lambda_m)^{1/2} \), the geometric average of \( \lambda_1 \) and \( \lambda_m \), and \( \alpha_{0s}^* = (\lambda_1 \lambda_m^3)^{1/4} \).
The quantity $\alpha_{0s}$ is the optimal choice for a single ADI parameter when all frequency components of the error are to be reduced [18]. Thus $\alpha_{0s}$ can be viewed as an effective average of eigenvalues of $A_1$. The upper bound of $\alpha_0$, $\alpha_{0s}$, is the geometric average of $\alpha_{0s}$ and $\lambda_m$ and therefore it is on the three-fourth geometric position of the eigenvalues of $A_1$. Therefore $I_{\text{noise}}$ would contain eigenvalues that are medium-large to large.

Depending on the noise level incorporated in the image, one can determine $\alpha_0$ as a geometric interpolation of $\alpha_{0s}$ and $\alpha_{0s}$. For example,

$$
\alpha_0 = (\alpha_{0s})\delta(\alpha_{0s})^{1-\delta}, \quad \delta \in [0, 1],
$$

(5.5)

where $\delta$ is the noise level. It is often the case that the noise itself has high frequency components only. However, when the noise level is high, the noise turns to become lower frequencies during the diffusion iteration. Noise removal can be carried out most efficiently (and accurately) when the algorithm parameters are chosen focusing on not only the original high frequency components of the noise but also the diffused noise. For the cases the noise level is not known, one may choose a small number for the noise level $\delta$ (zero or near zero), which is equivalent to focusing on higher frequency components of the error. Most small numbers set for $\delta$ have worked well for various real images.

Since $A_1$ is tridiagonal and made from a (nonnegative) diffusion operator, one can easily get

$$
\beta_0 = \|A_1\|_\infty = 2 \max_{1 \leq i \leq m} \{a_{ii}\}.
$$

(5.6)

Following Wachspress [18], we first find a single frequency-parameter $\xi$ as the geometric average of $\alpha_0$ and $\beta_0$, i.e., $\xi = (\alpha_0\beta_0)^{1/2}$, and therefore the ADI parameter is given as

$$
\Delta t_{\text{ADI}} = 2(\alpha_0\beta_0)^{-1/2}.
$$

(5.7)

For the cyclic parameters of length two, two frequency-parameters $\xi_1$ and $\xi_2$ ($\xi_1 > \xi_2$) can be determined as the solutions of the quadratic equation

$$
\sqrt{\alpha_1\beta_1} = \frac{1}{2}(\xi + \frac{\alpha_0\beta_0}{\xi}),
$$

(5.8)

where

$$
\alpha_1 = \sqrt{\alpha_0\beta_0}, \quad \beta_1 = \frac{\alpha_0 + \beta_0}{2}.
$$

Then the cyclic ADI parameters of length two read

$$
(\Delta t)_{\text{ADI},k} = 2\xi_k^{-1}, \quad k = 1, 2.
$$

(5.9)

For example, when $m = 300$, $\|A_1\|_\infty = 4$, and $\delta = 0.33$ (33% error), we obtain the single and double parameters as

$$
(\Delta t)_{\text{ADI}} \approx 2.87; \quad (\Delta t)_{\text{ADI},1} \approx 0.92 \quad \text{and} \quad (\Delta t)_{\text{ADI},2} \approx 8.94.
$$
The maximum $M(\Delta t)$ in (5.3), for $\lambda_i, \lambda_j \in I_{\text{noise}}$, becomes 0.49 and 0.26 (in average) for single and double parameters, respectively, and therefore the expected iterations turn out to be 10 and 6 for the reduction of all frequency components of the error in $I_{\text{noise}}$ by a factor of $10^{-3}$. (ADI is so fast!) In practice, the iteration with the tolerance of $10^{-3}$ takes about twice the expected, because the components of the error outside $I_{\text{noise}}$ are also being reduced with a slower speed. One may try larger cyclic lengths to further improve efficiency. Here we consider only up to two, for simplicity, and because the single parameter is fast enough for noise removal of most real images.

For FS and AOS, one might want to try the same analysis as for the ADI method. Note that for the heat equation,

\[ \sigma(T_{FS}) = \left\{ \frac{1}{(1 + \Delta t \lambda_i)(1 + \Delta t \lambda_j)} : i, j = 1, \cdots, m \right\}, \]

\[ \sigma(T_{AOS}) = \left\{ \frac{1 + \Delta t \lambda_i + \Delta t \lambda_j}{(1 + 2\Delta t \lambda_i)(1 + 2\Delta t \lambda_j)} : i, j = 1, \cdots, m \right\}. \]

To make $M(\Delta t)$ in the corresponding min-max problem (for eigenvalues of $A_1$ in $I_{\text{noise}}$) as small as possible, one has to choose $\Delta t$ as large as possible. That is, the solution of the min-max problem is trivial: $\Delta t = \infty$. But a large $\Delta t$ cannot be an optimal choice, because it reduces the error focusing on low-frequency components. Then, how can we find a set of optimal parameters for FS and AOS? The only answer seems to be “from experiments”. It has been found that cyclic parameters hardly improve efficiency of FS and AOS. It has been numerically verified that the choices

\[ (\Delta t)_{FS} = (\Delta t)_{AOS} = 4(a_0\beta_0)^{-1/2} \quad (5.10) \]

give a reasonable convergence speed.

Remark. Since the matrices $A_1$ and $A_2$ depend on the solution, they vary as time marches and the optimal parameters (timestep sizes) should be reset either in each iteration or once for a few iterations. In practice, $\beta_0 = \|A_1^{n-1/2}\|_\infty$ decreases slowly as $n$ grows. However, most trials of resetting $a_0$ and $\beta_0$ have failed to show an apparent improvement of the convergence speed. It is recommended not to reset them later the first iteration. With a fixed single parameter, ADI takes only 3-8 iterations to reduce the noise of real images, for a reasonable tolerance, as numerically verified in the next section.

6. Numerical Experiments

An effective texture-preserving noise removal algorithm seems strongly dependent on the numerical techniques for the edge-stopper $|\nabla u|$. We have implemented the numerical methods presented in this article, for the noise removal with the anisotropic
diffusion (1.1). All the computations are carried out on a 400MHz Laptop having a Linux operating system.

When the term \( u^{n-1/2} \) is obtained utilizing (2.4), the convergence is improved by 5-20\% for MB-FD and Min-Slope. However, for the extrapolation, the central and one-sided FD schemes require a smaller timestep to guarantee convergence. Since they diffuse the noise (and features of the image) extensively, the difference between successive iterates can be relatively large and therefore the extrapolation can overshoot the iteration process to diverge unless the timestep and the noise level are sufficiently small. In this article, we have utilized and will use (2.3), i.e., \( u^{n-1/2} = u^{n-1} \), for all examples.

6.1. Piecewise constant images

Various numerical methods have been tested for noise removal. In this subsection, we will show their performances for piecewise constant images, focusing on ADI (4.2) incorporating Min-Slope (3.4) and the double parameters (5.9). The noisy images are obtained from original images of no noise, by replacing a certain per cent of cell values by computer-generated random numbers (scaled to be between zero and one). The noise removal process (the iteration) stops when

\[
\| u^n - u^{n-1} \|_\infty := \max_{i,j} |u^n_{ij} - u^{n-1}_{ij}| < 10^{-3}.
\]

To check the quality of restored images, the difference is measured in \( \ell^1 \)-norm:

\[
\ell^1\text{-Err} := \| u^{\text{orig}} - u^n \|_1 = \frac{1}{m^2} \sum_{i,j=1}^m |u_{ij}^{\text{orig}} - u_{ij}^n|,
\]

where \( u^{\text{orig}} \) denotes the original image of no noise. For all experiments in this subsection, ADI utilizes the double parameters (5.9) and FS and AOS use the single parameter given in (5.10).

In Figure 4, we present the noisy and denoised images by ADI incorporating Min-Slope (3.4). The images are in a 500 \times 500 grid mesh and are added 50\% (left) and 80\% (right) noise; ADI takes respectively 18 and 30 iterations for a tolerance of \( 10^{-3} \). For the 50\% noise, the Min-Slope restores the image quite accurately (\( \ell^1\text{-Err}=0.0059 \)). Every corner of the background house and letters are clearly recovered. When the noise level is 80\%, the noisy image is hardly readable; the restored image shows clear features of the image (\( \ell^1\text{-Err}=0.029 \)). The figure provides an example of efficient noise elimination (rather than diffusion) for Min-Slope applied to piecewise constant images.

When the \( \ell^1 \)-error is not larger than 0.03, the restored image turns out to reveal the main features well.
Figure 4: Noisy and restored images by ADI incorporating Min-Slope. The images are in a $500 \times 500$ grid mesh and are added 50\% (left) and 80\% (right) noise.
Figure 5: Noisy and restored images by ADI incorporating the Min-Slope scheme. The images are in a $200 \times 200$ grid mesh and are added 33\% (left) and 67\% (right) noise.
Table 1: The number of iterations (Iter), the elapsed time (CPU, in second), and $\ell^1$-Err for the three time-marching methods incorporating Min-Slope to reach at the three-digit tolerance. The original image is the same as one used in Figure 5, but on 500 × 500 grid cells.

<table>
<thead>
<tr>
<th>Noise</th>
<th>AOS</th>
<th></th>
<th></th>
<th>FS</th>
<th></th>
<th></th>
<th>ADI</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>Iter 30.9 0.0054</td>
<td>Iter 39.4 0.0050</td>
<td>Iter 10.9 0.0039</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33%</td>
<td>Iter 53.3 0.0082</td>
<td>Iter 36.0 0.0074</td>
<td>Iter 11.2 0.0056</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>Iter 53.9 0.0160</td>
<td>Iter 40.9 0.0143</td>
<td>Iter 15.1 0.0095</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67%</td>
<td>Iter 49.7 0.0340</td>
<td>Iter 47.8 0.0322</td>
<td>Iter 11.4 0.0197</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75%</td>
<td>Iter 55.8 0.0571</td>
<td>Iter 47.7 0.0539</td>
<td>Iter 16.7 0.0296</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5 presents images on a coarser mesh: 200 × 200. For 25%, 33%, 50%, 67%, and 75% noisy images, the $\ell^1$-Err (and the iteration number for the tolerance of $10^{-3}$) become 0.010 (16), 0.014 (16), 0.024 (26), 0.049 (22), and 0.072 (34), respectively. Again, ADI incorporating Min-Slope restores corners and curves very clearly when the noise level is not larger than 50%. Another example of noise elimination for Min-Slope!

In Table 1, we compare the three time-marching methods: AOS, FS, and ADI. The original image is the same as one used in Figure 5, but on 500 × 500 grid cells. Min-Slope is utilized for the edge-stopping FD scheme and the iteration is carried out up to reaching at the three-digit tolerance. As one can see from the table, ADI converges 3-5 times faster than AOS and FS. Also it should be noticed that ADI restores more accurate images. Such superior performances of ADI can be reasonably explained as follows. ADI incorporating the parameters (5.9) reduces medium-high to high frequency components of the error (the noise) not only faster than AOS and FS but also relatively faster than lower frequency components.

In Table 2, the FD schemes are compared to measure their edge-stopping ability, when applied with five iterations of ADI. We choose the original image used in Figure 4 on 500×500 grid cells. Again Min-Slope (3.4) restores the curves and edges clearly for the images of noise level not larger than 75%. MB-FD (3.3) preserves edges quite well when the noise level is not larger than 50%, while the restored image shows unclear portions for the 75% noise level. The one-sided FD method loses the control quickly as the noise level increases.

We have compared the quality of restored images obtained from (1.1) and (2.6) with the edge-stopping functions $g$ in (2.7). The anisotropic diffusion (1.1) turns out
Table 2: The error $\ell^1$-Err for three FD methods for $|\nabla u|$ applied with 5 iterations of ADI. The original image is the same as one used in Figure 4 on $500 \times 500$ grid cells.

<table>
<thead>
<tr>
<th>Noise</th>
<th>One-Sided</th>
<th>MB-FD</th>
<th>Min-Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>0.0104</td>
<td>0.0029</td>
<td>0.0027</td>
</tr>
<tr>
<td>33%</td>
<td>0.0166</td>
<td>0.0042</td>
<td>0.0037</td>
</tr>
<tr>
<td>50%</td>
<td>0.0517</td>
<td>0.0090</td>
<td>0.0068</td>
</tr>
<tr>
<td>67%</td>
<td>0.0874</td>
<td>0.0286</td>
<td>0.0158</td>
</tr>
<tr>
<td>75%</td>
<td>0.1298</td>
<td>0.0621</td>
<td>0.0289</td>
</tr>
</tbody>
</table>

to be superior to (2.6) and has shown satisfactory properties when combined with the MB and Min-Slope FD schemes.

Remark. Due to the choice involved in the Min-Slope FD scheme, it holds flat portions of the image tightly. For the cases the noise incorporates near-flat portions, it takes a large number of iterations for a convergence unless there involves an acceleration mechanism. As a matter of fact, the double parameters in (5.9) are introduced as an acceleration scheme particularly for Min-Slope. However, the double parameters of ADI are too strong to be applied to more general images. ADI incorporating the MB scheme and the single parameter (5.7) has proven efficient enough for general images; see §6.2.

6.2. General images

For the noise removal for general images, we prefer to apply ADI due to the following two reasons:

- First, ADI is fast and accurate. It is fast, as shown in the convergence analysis in §5. It can be applied focusing on frequency components of the noise, which makes the algorithm restore more accurate images, as numerically verified for piecewise constant images.

- Secondly, but more importantly, ADI provides an acceptable way of automatic stop of the diffusion iteration. When the single parameter $\Delta t$ is chosen as in (5.7), we have [18]

$$\max_{x,y \in [a_0, \beta_0]} \left| \frac{1 - \frac{\Delta x}{2} x}{1 + \frac{\Delta x}{2} x} \cdot \frac{1 - \frac{\Delta y}{2} y}{1 + \frac{\Delta y}{2} y} \right| \leq \gamma_0 := \sqrt{\beta_0 - \alpha_0} \frac{\sqrt{\beta_0} + \alpha_0}{\sqrt{\beta_0} - \alpha_0}.$$  \hspace{1cm} (6.1)

Thus the number of iterations $n$ required to reduce all frequency components of the noise corresponding to $[\alpha_0, \beta_0]$ by a factor of $\varepsilon$ becomes

$$n \approx \log \varepsilon / \log \gamma_0,$$  \hspace{1cm} (6.2)
under the assumption that the diffusion in piecewise smooth faces of the image follows the heat transfer. In practice, it suffices to choose $\varepsilon \in [0.05, 0.01]$.

It has been observed that when the noise level is not greater than 0.25 (25%), the noise hardly forms lower frequency components of the error during the diffusion. Thus one may choose $a_0$ as

$$a_0 = (a_{0*})^\delta (a_0^*)^{1-\delta}, \quad \delta_0 = \max(0, \delta - 0.25),$$

where $\delta \in [0, 1]$ is the noise level; see (5.4) and (5.5). When $\varepsilon = 0.03$ and $\|A_1\|_{\infty} = 4$, the numbers of iterations for images on 200$^2$ to 600$^2$ meshes turn out to be 3-8, almost linearly starting from 25% to 75% noise level.

In Figure 6, we present images processed by ADI with MB-FD on a 400x400 mesh: (a) the original image, (b) 50% noisy image, (c) the image in 2 iterations, and (d) the denoised image in 5 iterations. We set the noise reduction tolerance $\varepsilon = 0.03$ and the iteration stops automatically in five iterations, taking 2.1 seconds of CPU time. The $\ell^1$-errors are 0.0150 and 0.0139 for the second and fifth iterates, respectively. The error is monitored up to ten iterations; the noisy image ($u^0$) contains the error of 0.1429 and the error is minimized in four iterations ($\ell^1$-Err=0.0138) and increases to be 0.0308 for the 10th iterate.

Figure 7 shows the performance of ADI applied to the 20% noisy Lena image in Figure 2. The automatic stopping (6.1)-(6.3) determines four iterations for the tolerance 0.03. For Min-Slope, the restored image, Figure 7(a), shows clearly the main features and the boundaries of locally smooth faces of the image ($\ell^1$-Err=0.012), although there still remains some parts of the noise; a caution is required when the scheme is adopted. When MB-FD is utilized, ADI restores the image satisfactorily as shown in Figure 7(b) ($\ell^1$-Err=0.012). The noise has been reduced sufficiently in four iterations (taking about 3.1 seconds) and the main features are hardly smeared out.

To loosen the tight holding of Min-Slope, one may apply MB-FD once each several iterations of Min-Slope. For example, for the Lena image in Figure 7, the $\ell^1$-error of the fourth iterate of ADI and Min-Slope becomes 0.009 when the second iteration is replaced by MB-FD. Superb! However, the combination is yet problematic. Strategies of automatic pointwise combination of MB-FD and Min-Slope would be presented elsewhere [9].

In Figure 8, we present the performance of ADI and MB-FD for the Lena image in higher noise levels: 33% (left) and 67% (right). For the 33% noise level, the restored image does not show any big changes from the original image ($\ell^1$-Err=0.016). When two thirds of the cell values are replaced by random values (67% noise), the noisy image becomes hardly readable. In noise removal, the algorithm stops automatically in seven iterations. The restored image looks not so bad ($\ell^1$-Err=0.058); main features
Figure 6: Performance of ADI with MB-FD: (a) the original image on a 400 × 400 mesh, (b) 50% noisy image, (c) the image in 2 iterations, and (d) the denoised image in 5 iterations.
Figure 7: Performance of ADI for the 20% Lena image in Figure 2: (a) the restored image with the Min-Slope scheme, and (b) the restored image with MB-FD.

Figure 8: Performance of ADI and MB-FD for the Lena image: (left) 33% noisy image and the restored in four iterations and (right) 67% noisy image and the restored in six iterations.
Figure 9: Performance of ADI and MB-FD for a fish brain image in a 600 × 467 mesh, generated for an education purpose. (top) the original image and (bottom) the restored image in four iteration.
are mostly recovered, although there still remain some parts of the noise. It is verified by checking the error and appearance of the restored images for various iterations that the fourth and sixth iterates are the best choices respectively for 33% and 67% noisy images.

Figure 9 shows the original and restored images for a fish brain, scanned from an old book for an education purpose. Due to lots of dirt spots, the original image is not convenient to look. The automatic stopping scheme selects four iterations to restore the image, taking 2.8 seconds. Readability is improved for the image to look more conveniently; the noise is eliminated except large spots and no readable letter in the original image is diffused to become unreadable. The image is enhanced satisfactorily!

We summarize what we have found in this section:

- Min-Slope seems not diffusing but eliminating the noise, although it is hard to work at some (flat) portions of the noise.
- ADI incorporating the MB scheme (3.3) preserves features of images quite satisfactorily.
- The automatic stopping strategy in (6.1), (6.2), and (6.3) works well.

7. Discussion

In noise removal, the key issues are (a) edge-stopping differential/difference formulas, (b) computational efficiency, and (c) automatic stop of the diffusion iteration. It seems relatively easy to successfully handle problems related to (b) and (c), as suggested and numerically verified in previous sections. In this section, we discuss edge-stopping formulas.

**Edge-stopping formulas:** There have been various anisotropic diffusion equations that incorporate the curvature and the gradient magnitude. Let the given image have a homogeneous areal feature, for simplicity. Consider the anisotropic diffusion (1.1) and its corresponding level set equation

\[
\frac{\partial \phi}{\partial t} = \kappa(\phi) |\nabla \phi|,
\]

(7.1)

whose zero level set represents the interface of \( u \). (See Sethian [17] for details of level set methods.) Let us further assume that the image is divided half and half by a straight vertical line segment, having a different value in each side. Then, when the level set function is initialized with the signed distance, its curvature is completely zero for both the analytic solution and the numerical approximation, and therefore no movement can happen for the level set function. However, for the anisotropic diffusion (1.1) in which \( u \) is initialized with the original image, only the region where
the flow can possibly have a nonzero diffusion is a vicinity of the interface. That is, the
most vulnerable area through the anisotropic diffusion is the edges themselves. Such
an observation implies that (a) mathematical/computational principles developed for
the level set equation may not be directly applied to the anisotropic diffusion and
(b) a curvature-related anisotropic diffusion equation itself is not sufficient enough to
save edges (even a straight line segment) from the diffusion process. Special cares
should be taken for an effective edge-stopping; in this article, we have tried to find
sufficient conditions for non-blurring noise removal from numerical manipulations.

**Fourth-order anisotropic diffusion:** As edge-stopping difference formulas, we
have introduced the minimum-biased (MB) and Min-Slope FD schemes, (3.3) and
(3.4). These schemes preserves piecewise constant images perfectly (with an excep-
tion at tips of thin curves for MB-FD) when no noise is added. Min-Slope has shown
excellent edge-stopping properties for the noise removal from piecewise constant im-
ages. MB-FD is a little more diffusive than Min-slope; however, it turns out to be
satisfactory in both removing the noise and preserving edges for general images.

It seems natural to ask the following question: What difference formulas can
preserve edges of piecewise linear or quadratic textures (when no noise is present)?
Note that the gradient itself at individual grid points hardly implies any of smooth
properties of the image. Thus one has to consider higher-order derivatives of \( u \) to get
more relevant information on the texture smoothness. Consider the level set equation
incorporating intrinsic Laplacian of the curvature [17, §14.6 and references therein]:

\[
\frac{\partial \phi}{\partial t} = \kappa_{\alpha\alpha} |\nabla \phi|,
\]

where \( \alpha \) is the arc length. Since a unit vector in the tangential direction (to the arc)
is \( (-\phi_y, \phi_x)/|\nabla \phi| \), the corresponding anisotropic diffusion equation can be formulated
as follows:

\[
\frac{\partial u}{\partial t} = |\nabla u| \tau \cdot \nabla \left( \nabla \cdot \left( \nabla \cdot \frac{\nabla u}{|\nabla u|} \right) \right), \quad \tau = \frac{(-u_y, u_x)}{|\nabla u|}.
\]

However, its numerical schemes can be either easily unstable or extremely slow. Fur-
thermore, the physical meaning of \( \kappa(u)_{\alpha\alpha} \) is ambiguous. In a view point of efficient
noise removal, it is not surprising to replace the gradient and the arc-length Laplacian
respectively by a more generalized term and the standard Laplacian:

\[
\frac{\partial u}{\partial t} = g(|\nabla u|, |\Delta u|) \Delta \kappa(u),
\]

where \( g \) acts as an edge-stopper satisfying (2.10) for both variables. In a companion
paper [8], we introduce and compare stable and efficient numerical algorithms
for the fourth-order anisotropic diffusion (7.3) that preserve piecewise linear images
completely.
8. Conclusions

We have introduced edge-stopping difference formulas for noise removal for piecewise constant images: the minimum-biased finite difference (MB-FD) and minimum slope (Min-Slope) schemes. Three different locally one-dimensional time-marching algorithms (ADI, FS, and AOS) have been reviewed and their convergence behaviors are analyzed for the purposes of accelerating the convergence speed, minimizing the image blur, and establishing an automatic stopping strategy for the diffusion process. The FD formulas and time-marching algorithms are applied to noise removal for various images including the Lena image. It has been numerically verified that (a) ADI converges fast and performs superior to FS and AOS, (b) MB-FD and Min-Slope have excellent edge-stopping properties, and (c) the automatic stopping strategy works satisfactorily for various images. The discussed are anisotropic diffusion formulas that can preserve piecewise linear images precisely.

Acknowledgment

The author wants to express sincere thanks to Prof. Osher, UCLA, for his interests, helpful comments, and encouragement. The code utilized in this article has been implemented for the model-code library at the University of Kentucky: Graduate Research and Applications for Differential Equations (GRADE), and is available through internet access to www.ms.uky.edu/~skim/GRADE/ or by asking the author (skim@ms.uky.edu).

References


