

Fractional-Order Anisotropic Diffusion for Image Denoising

Jian Bai and Xiang-Chu Feng

Abstract—This paper introduces a new class of fractional-order anisotropic diffusion equations for noise removal. These equations are Euler–Lagrange equations of a cost functional which is an increasing function of the absolute value of the fractional derivative of the image intensity function, so the proposed equations can be seen as generalizations of second-order and fourth-order anisotropic diffusion equations. We use the discrete Fourier transform to implement the numerical algorithm and give an iterative scheme in the frequency domain. It is one important aspect of the algorithm that it considers the input image as a periodic image. To overcome this problem, we use a folded algorithm by extending the image symmetrically about its borders. Finally, we list various numerical results on denoising real images. Experiments show that the proposed fractional-order anisotropic diffusion equations yield good visual effects and better signal-to-noise ratio.

Index Terms—Anisotropic diffusion, image smoothing, fractional-order partial differential equation, fractional-order difference, image denoising.

I. INTRODUCTION

SINCE the work of Perona and Malik [1], which replaces the isotropic diffusion by an anisotropic diffusion, many methods in connecting adaptive smoothing with systems of nonlinear partial differential equations (PDEs) [2]–[8] have been proposed to preserve important structures in images, while removing noise. Anisotropic diffusion is associated with an energy-dissipating process that seeks the minimum of the energy functional. When the energy functional is the total variation norm of the image, the well-known total variation (TV) minimization model [9] can be obtained. Although these techniques have been demonstrated to be able to achieve a good tradeoff between noise removal and edge preservation, the images resulting from the application of these techniques in the presence of noise are often piecewise constant; thus, the finer details in the original image may not be recovered satisfactorily, and affine regions will look “blocky.” To reduce the blocky effect, while preserving sharp jump discontinuities (edges), many other nonlinear filters have been suggested in the literature [10]–[18] and during the last few years, fourth-order PDEs have been of special interest [13]–[18]. For example, You and Kaveh proposed a class of fourth-order PDEs which are Euler–Lagrange equations of a cost functional which is an increasing function of the

absolute value of the Laplacian of the image intensity function [15]. Since second-order derivatives are zero only if the image intensity function is planar, unlike second-order PDEs, this class of fourth-order PDEs will evolve toward and settle down to a planar image if the image support is infinite. Piecewise planar images look more natural than the step images that are stationary points of second-order PDEs.

Recently, fractional-order PDEs have been studied in computer vision. For example, Cuesta proposed fractional-order linear integro-differential equations which interpolate heat equations and wave equations using the Riemann–Liouville (R–L) fractional derivative [19], while in [20] and [21], fractional-order scale spaces (α scale spaces) and the fractional high-order linear filtering were introduced. The fractional derivative can be seen as the generalization of the integer-order derivative. It has been studied by many mathematicians (for example Euler, Hardy, Littlewood, and Liouville) [22]. Not until Mandelbrot found fractals and applied the R–L fractional derivative to the Brownian motion did the fractional derivative cause great attention. The fractional derivative has been applied to many fields such as noise detection and estimation [23], [24], electromagnetic theory [25], wavelets, and splines [26], [27]. There are many methods that can define the fractional derivative only if it consists with the integer-order derivative. The usual definitions among them involve: R–L fractional derivative, Cauchy-integral fractional derivative, frequency-domain (Fourier-domain) fractional derivative.

In this paper, we propose fractional-order anisotropic diffusion equations, which are Euler–Lagrange equations of a cost functional which is an increasing function of the absolute value of the fractional derivative of the image intensity function. The proposed pseudo-PDEs will lead to a “natural interpolation” between Perona–Malik equations and fourth-order anisotropic diffusion equations in [15]. The frequency-domain fractional derivative definition is a relatively simple definition, so we use the Fourier transform to compute the fractional derivative. Correspondingly, the discrete Fourier transform is used to compute the fractional-order difference. We compute the evolution, along flow, work in the Fourier domain. Related works inspired from the OSV model [28] and also using the Fourier transform for computations and variants are by Roudenko [29] and Lieu [30]. Our algorithm is easy to implement and has low computational cost. This is owing to the implementation of the fast discrete Fourier transform. However, the use of the discrete Fourier transform automatically imposes the period boundary condition on the proposed models, so we use a folded algorithm to eliminate jump discontinuities across edges. Next, we present an analysis of the folded algorithm and find an

Manuscript received January 14, 2007; revised June 27, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Philippe Salembier.

The authors are with the Department of Applied Mathematics, Xidian University, Xi'an 710071, China (e-mail: keywhite26@126.com).

Digital Object Identifier 10.1109/TIP.2007.904971

interesting property of the fractional-order anisotropic diffusion. We use the proposed algorithm for noise removal and show that the proposed algorithm yields better signal-to-noise ratio (SNR) and visual effects. To reduce the blocky effect and avoid speckle artifacts, we suggest taking the differential-order $\alpha = 1.5$. The stability analysis of the difference scheme is also established, then we can apply an order-dependent step size to the numerical algorithm. We also compare our algorithm with the common Perona–Malik algorithm which uses forward and backward difference schemes. Results show that the common Perona–Malik algorithm cannot smooth severely noise-contaminated point, and yet our algorithm can reduce this effect and obtain higher SNR. This is owing to the use of the central difference scheme.

The outline of this paper is as follows. In Section II, we give a simple introduction of the fractional derivative and show that the time evolution of the proposed models seeks to minimize an increasing cost functional of the absolute value of the fractional derivative of the image intensity function. In Section III, we use the discrete Fourier transform (DFT) to implement the numerical algorithm. An iterative solution of the proposed models is obtained. In Section IV, we develop a folded algorithm to avoid possible discontinuities across the image borders and illustrate an interesting finding of the fractional-order anisotropic diffusion. Numerical examples are presented in Section V and the paper is concluded in Section VI. The stability analysis of the difference scheme is illustrated in the Appendix.

II. FRACTIONAL-ORDER ANISOTROPIC DIFFUSION

We now briefly describe the fractional derivative. The fractional-order derivative can be seen as the generalization of the integer-order derivative. It can be computed in a number of ways only if it consists with the integer-order derivative. Many found, using their own notation and methodology, definitions that fit the concept of a noninteger-order derivative. The most famous of these definitions that have been popularized are the R–L and Grunwald–Letnikov definitions. However, in this paper, we use the frequency domain definition because it is easy to implement. For any function $f(t) \in L^2(R)$, the Fourier transform of it is

$$\hat{f}(w) = \int_R f(t) \exp(-jwt) dt. \quad (1)$$

The equivalent form of the first-order derivative in the frequency domain is

$$Df(t) \leftrightarrow (jw)\hat{f}(w) \quad (2)$$

where “ \leftrightarrow ” denotes the Fourier transform pair. It is straightforward that the Fourier transform of the α th-order derivative, $D_\alpha f(t)$, is $(jw)^\alpha \hat{f}(w)$. Of course, there is no reason why α must be an integer, α can be any real number, hence, the fractional derivative.

Thus, we define the fractional derivative in the frequency domain

$$D_\alpha f(t) \leftrightarrow (jw)^\alpha \hat{f}(w), \quad \alpha \in R^+. \quad (3)$$

It is obvious that the semi-group property of fractional derivative operators holds, namely

$$\begin{aligned} (D_\alpha)(D_\beta)f &= (D_\beta)(D_\alpha)f \\ &= (D_{\alpha+\beta})f, \quad \text{for any } \alpha, \beta \in R^+. \end{aligned} \quad (4)$$

For any $g(x, y) \in L^2(R^2)$, the corresponding 2-D Fourier transform of it is

$$\hat{g}(w_1, w_2) = \int_{R^2} g(x, y) \exp(-j(w_1x + w_2y)) dx dy. \quad (5)$$

Thus, fractional-order partial derivatives are

$$D_{\alpha x}g = F^{-1}((jw_1)^\alpha \hat{g}(w_1, w_2)) \quad (6)$$

$$D_{\alpha y}g = F^{-1}((jw_2)^\alpha \hat{g}(w_1, w_2)) \quad (7)$$

where F^{-1} is the inverse 2-D continuous Fourier transform operator.

Often when working with images, we are dealing with functions defined only on a bounded domain. On the other hand, the fractional derivative is defined using the Fourier transform which is defined for functions given on the whole space. Thus, when the input image function $u \in H^{2\alpha}(\Omega)$ ($\alpha > 0$) is defined only on $\Omega \subset R^2$, we will consider prolongating it onto R^2 and keep its regularity. The Sobolev space can be defined as follows.

Definition 1: For any $s \in R$, the space $H^s(R^2)$ consists of tempered distributions g such that

$$(1 + |w_1|^2 + |w_2|^2)^{s/2} \hat{g} \in L^2(R^2).$$

The space $H^s(R^2)$ is a Hilbert space equipped with the inner product

$$\langle g, h \rangle_s = \int (1 + |w_1|^2 + |w_2|^2)^s \hat{g} \bar{\hat{h}} dw_1 dw_2$$

and the associated norm $\|g\|_s = \sqrt{\langle g, g \rangle_s}$. The space $H^s(\Omega)$ is defined as the restriction of $H^s(R^2)$ on Ω .

We assume the domain Ω has certain regularities, then the prolongation theorem [31] ensures there exists a linear prolongation operator E which projects $H^{2\alpha}(\Omega)$ onto $H^{2\alpha}(R^2)$, so we can prolongate $u \in H^{2\alpha}(\Omega)$ to $Eu \in H^{2\alpha}(R^2)$ by the linear prolongation operator E .

Smoothing (by local weighted averaging) is an effective image regularization method that has been used for denoising, restoration, and enhancement. A drawback is that smoothing can damage image features such as edges, lines, and textures. To avoid the damage, the smoothing has to be adaptively controlled by the amount of smoothing or the direction of smoothing. A classic example of adaptive smoothing is the anisotropic diffusion scheme of Perona and Malik [1], in which the smoothing process is formulated by a partial differential equation. Let t denote the time and $c(\cdot)$ the diffusion coefficient, the anisotropic diffusion as formulated in [1] may be presented as

$$\frac{\partial u}{\partial t} = \operatorname{div}(c(|\nabla u|^2) \nabla u). \quad (8)$$

This equation is associated with the following energy functional:

$$E(u) = \int_{\Omega} f(|\nabla u|) d\Omega \quad (9)$$

where Ω is the image support, and $f(\cdot) \geq 0$ is an increasing function associated with the diffusion coefficient as

$$c(s) = \frac{f'(\sqrt{s})}{\sqrt{s}}. \quad (10)$$

Anisotropic diffusion is then shown to be an energy-dissipating process that seeks the minimum of the energy functional. We consider the following functional defined in the space of continuous images over a support of Ω

$$E(u) = \int_{\Omega} f(|D_{\alpha} u|) d\Omega \quad (11)$$

where D_{α} denotes the fractional derivative operator defined by $D_{\alpha} u = (D_{\alpha x} u, D_{\alpha y} u)$ and $|D_{\alpha} u| = \sqrt{D_{\alpha x}^2 + D_{\alpha y}^2}$. We can formally compute the Euler–Lagrange equation for this minimization problem as follows.

Take any test function $\eta \in C^{\infty}(\Omega)$. Define

$$\Phi(a) = \int_{\Omega} f(|D_{\alpha} u + a D_{\alpha} \eta|) dx dy. \quad (12)$$

We obtain

$$\begin{aligned} \Phi'(0) &= \frac{d}{da} \int_{\Omega} f(|D_{\alpha} u + a D_{\alpha} \eta|) dx dy \Big|_{a=0} \\ &= \int_{\Omega} \left(f'(|D_{\alpha} u|) \frac{D_{\alpha x} u}{|D_{\alpha} u|} D_{\alpha x} \eta \right. \\ &\quad \left. + f'(|D_{\alpha} u|) \frac{D_{\alpha y} u}{|D_{\alpha} u|} D_{\alpha y} \eta \right) dx dy \\ &= \int_{\Omega} (D_{\alpha x}^*(c(|D_{\alpha} u|^2) D_{\alpha x} u) \\ &\quad + D_{\alpha y}^*(c(|D_{\alpha} u|^2) D_{\alpha y} u)) \eta dx dy \end{aligned}$$

for all $\eta \in C^{\infty}(\Omega)$, where $D_{\alpha x}^*$ is the adjoint of $D_{\alpha x}$ and $D_{\alpha y}^*$ is the adjoint of $D_{\alpha y}$. Thus, the Euler–Lagrange equation is

$$D_{\alpha x}^*(c(|D_{\alpha} u|^2) D_{\alpha x} u) + D_{\alpha y}^*(c(|D_{\alpha} u|^2) D_{\alpha y} u) = 0. \quad (13)$$

The Euler–Lagrange equation may be solved through the following gradient descent procedure:

$$\frac{\partial u}{\partial t} = -D_{\alpha x}^*(c(|D_{\alpha} u|^2) D_{\alpha x} u) - D_{\alpha y}^*(c(|D_{\alpha} u|^2) D_{\alpha y} u) \quad (14)$$

with the observed image as the initial condition. The solution is arrived when $t \rightarrow \infty$, but the time evolution may be stopped earlier to achieve an optimal tradeoff between noise removal and edge preservation.

Note that in our Euler–Lagrange equation, for the unknown $u \in H^{2\alpha}(\Omega)$, we will first prolongate u to $Eu \in H^{2\alpha}(R^2)$ by the prolongation operator E when we work with the Fourier

transform \hat{u} of u and then impose $D_{\alpha} u \cdot \vec{n} = 0$ on $\partial\Omega$, where \vec{n} is the unit outward normal to $\partial\Omega$. When $\alpha = 1$, (14) is precisely the Perona–Malik equation (8); when $\alpha = 2$, (14) is precisely the fourth-order anisotropic diffusion equation in [15]; when $1 \leq \alpha \leq 2$, (14) leads to a “natural interpolation” between them. In this paper, we are interested in $1 \leq \alpha \leq 2$ and consider it an appropriate interval for image smoothing.

III. NUMERICAL ALGORITHM

For practical applications, we first assume that the initial discrete image u is $m \times m$ pixels, and that it has been sampled from its continuous version at uniformly spaced points starting at $(0, 0)$, i.e., $u(x, y) = u(x\Delta x, y\Delta y)$ for $x, y = 0, \dots, m-1$. The grid size Δx and Δy is chosen as $\Delta x = \Delta y = 1$. We use the 2-D discrete Fourier transform (2-D DFT) to compute the fractional-order difference. It is one important aspect of the algorithm that it considers the input image as a periodic image, which is equivalent to imposing a period boundary condition on (14). The other difference schemes (e.g., in [19]) do not have this restriction. Although the use of the Fourier transform forces us to choose the periodic boundary condition, the discrete Fourier transform is easy to implement and has low computational cost. Moreover, it corresponds with the continuous Fourier transform which is used to compute the fractional derivative. Note that in the continuous case in order to compute the fractional derivative $D_{\alpha} u$ in (14), we must prolongate u to Eu because the continuous Fourier functions are defined on the whole space R^2 , however, the discrete Fourier basis functions are defined on a square domain, so we need not prolongate the discrete image in practical computations.

The 2-D DFT of $u(x, y)$ is

$$\hat{u}(w_1, w_2) = \frac{1}{m} \sum_{x,y=0}^{m-1} u(x, y) \exp(-j2\pi(w_1 x + w_2 y)/m). \quad (15)$$

The translation property of the 2-D DFT is

$$\begin{aligned} u(x - x_0, y - y_0) &\leftrightarrow \exp(-j2\pi(w_1 x_0 + w_2 y_0)/m) \hat{u}(w_1, w_2) \quad (16) \end{aligned}$$

here $u(x, y) = u(x \pm m, y) = u(x, y \pm m) = u(x \pm m, y \pm m)$.

Thus, the equivalent form of the first-order partial difference in the frequency domain is

$$\begin{aligned} u(x, y) - u(x - 1, y) &\leftrightarrow (1 - \exp(-j2\pi w_1/m)) \hat{u}(w_1, w_2). \quad (17) \end{aligned}$$

We define the fractional-order partial difference in the frequency domain

$$D_{\alpha x} u \leftrightarrow (1 - \exp(-j2\pi w_1/m))^{\alpha} \hat{u}(w_1, w_2). \quad (18)$$

Note that the formula (3) cannot be directly used to compute the fractional-order difference because it does not consist with the integer-order difference, however, the formula (18) does. The discrete image is periodically translated by the formula

(16), which implies using the period boundary condition in computing $D_{\alpha x}u$.

In practical computations, we use the central difference scheme to compute the fractional-order difference. This is equivalent to translating $D_{\alpha x}u$ by $\alpha/2$ units. Usually, $\alpha/2$ is not an integer, so we must interpolate the discrete image. Fortunately, the trigonometric interpolation can be automatically implemented by the translation property of the 2-D DFT, namely

$$u(x + \frac{\alpha}{2}, y) \leftrightarrow \exp(j\pi\alpha w_1/m)\hat{u}(w_1, w_2) \quad (19)$$

where u is the continuous interpolated image by the trigonometric interpolation. Since the formula (19) makes sense for all $\alpha/2$ and not only for integer $\alpha/2$, the continuous image u is unambiguously defined not just at integer x in the range $0 \leq x < m$, but in fact for all real number x . Thus, we obtain the central difference

$$\begin{aligned} \tilde{D}_{\alpha x}u &= D_{\alpha x}\left(u\left(x + \frac{\alpha}{2}, y\right)\right) \\ &\leftrightarrow (1 - \exp(-j2\pi w_1/m))^{\alpha} \\ &\quad \times \exp(j\pi\alpha w_1/m)\hat{u}(w_1, w_2). \end{aligned} \quad (20)$$

In correspondence to the inverse 2-D continuous Fourier transform operator F^{-1} , we use the notation F to denote the 2-D DFT operator and F^{-1} to denote the inverse 2-D discrete Fourier transform (2-D IDFT) operator, so we have

$$\begin{aligned} \tilde{D}_{\alpha x}u &= F^{-1}((1 - \exp(-j2\pi w_1/m))^{\alpha} \\ &\quad \times \exp(j\pi\alpha w_1/m)F(u)). \end{aligned} \quad (21)$$

Here, we introduce an interpolation using a complex trigonometric polynomial. However, for the fractional-order difference $\tilde{D}_{\alpha x}u$, we have the following result.

Proposition 1: When m is an odd integer, $\tilde{D}_{\alpha x}u$ is real value.

Proof: Let

$$\begin{aligned} p(w_1) &= (1 - \exp(-j2\pi w_1/m))^{\alpha} \\ &\quad \times \exp(j\pi\alpha w_1/m), -\frac{m-1}{2} \leq w_1 \leq \frac{m-1}{2}. \end{aligned}$$

Then we have

$$\begin{aligned} \text{conj}(p(-w_1)) &= \text{conj}((1 - \exp(j2\pi w_1/m))^{\alpha} \exp(-j\pi\alpha w_1/m)) \\ &= \text{conj}((1 - \exp(j2\pi w_1/m))^{\alpha}) \\ &\quad \times \text{conj}(\exp(-j\pi\alpha w_1/m)) \\ &= (1 - \exp(-j2\pi w_1/m))^{\alpha} \exp(j\pi\alpha w_1/m) \\ &= p(w_1) \end{aligned}$$

where $\text{conj}(\cdot)$ is the complex conjugation, so the function $p(w_1)$ is conjugate-symmetric with respect to $w_1 = 0$. Thus, the matrix $p(w_1)\hat{u}(w_1, w_2)(-(m-1)/2 \leq w_1, w_2 \leq (m-1)/2)$ is conjugate-symmetric with respect to $w_1, w_2 = 0$, so we have $\tilde{D}_{\alpha x}u$ is real value. \square

When m is an even integer, the domain of definition of the function $p(w_1)$ is not symmetric with respect to $w_1 = 0$. So usually, there is a complex component in $\tilde{D}_{\alpha x}u$. However, when

image size is enough large, this complex component is very small so that it can be ignored in computations.

Now we turn to compute the adjoint of $\tilde{D}_{\alpha x}$. Let K_1 be a purely diagonal operator in the frequency domain, defined by

$$K_1 = \text{diag}((1 - \exp(-j2\pi w_1/m))^{\alpha} \exp(j\pi\alpha w_1/m)).$$

We get

$$\tilde{D}_{\alpha x} = F^{-1} \circ K_1 \circ F. \quad (22)$$

Let $\tilde{D}_{\alpha x}^*$ be the adjoint of $\tilde{D}_{\alpha x}$, we get

$$\begin{aligned} \tilde{D}_{\alpha x}^* &= (F^{-1} \circ K_1 \circ F)^* \\ &= (K_1 \circ F)^* \circ (F^{-1})^* = F^* \circ K_1^* \circ (F^{-1})^*. \end{aligned}$$

Since the adjoint of F is F^{-1} , we get

$$\tilde{D}_{\alpha x}^* = F^{-1} \circ K_1^* \circ F.$$

Since K_1 is a purely diagonal operator, K_1^* is the complex conjugation of K_1 , we have

$$\begin{aligned} \tilde{D}_{\alpha x}^*u &\leftrightarrow \text{conj}((1 - \exp(-j2\pi w_1/m))^{\alpha} \\ &\quad \times \exp(j\pi\alpha w_1/m)\hat{u}(w_1, w_2)). \end{aligned} \quad (23)$$

The same algorithm is used for calculations of $\tilde{D}_{\alpha y}$ and $\tilde{D}_{\alpha y}^*$.

For central difference operators $\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha y}$, we have the following result.

Proposition 2: When α is an even integer, $\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha y}$ are symmetric; when α is an odd integer, $\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha y}$ are anti-symmetric.

Proof: Let $\varepsilon = -2\pi w_1/m$, then we have

$$\begin{aligned} K_1(w_1, w_1) &= (1 - \exp(j\varepsilon))^{\alpha} \exp\left(-\frac{\alpha}{2}j\varepsilon\right) \\ &= \sum_{r=0}^{\alpha} (-1)^r C_{\alpha}^r \exp(j\varepsilon)^r \exp\left(-\frac{\alpha}{2}j\varepsilon\right). \end{aligned}$$

When α is an even integer, we have

$$(-1)^r C_{\alpha}^r = (-1)^{\alpha-r} C_{\alpha}^{\alpha-r}$$

so

$$\begin{aligned} K_1(w_1, w_1) &= \sum_{r=0}^{\alpha} (-1)^r C_{\alpha}^r \exp\left(j\varepsilon(r - \frac{\alpha}{2})\right) \\ &= C_{\alpha}^{\frac{\alpha}{2}} (-1)^{\frac{\alpha}{2}} \\ &\quad + \sum_{r=0}^{\frac{\alpha}{2}-1} (-1)^r C_{\alpha}^r \left(\exp\left(j\varepsilon\left(r - \frac{\alpha}{2}\right)\right)\right. \\ &\quad \left.+ \exp\left(j\varepsilon\left(\frac{\alpha}{2} - r\right)\right)\right) \\ &= C_{\alpha}^{\frac{\alpha}{2}} (-1)^{\frac{\alpha}{2}} + 2 \sum_{r=0}^{\frac{\alpha}{2}-1} (-1)^r C_{\alpha}^r \cos\left(\varepsilon\left(\frac{\alpha}{2} - r\right)\right). \end{aligned}$$

Thus, we have $K_1^* = K_1$, so $\tilde{D}_{\alpha x}^* = \tilde{D}_{\alpha x}$.

When α is an odd integer, we have

$$(-1)^r C_{\alpha}^r = -(-1)^{\alpha-r} C_{\alpha}^{\alpha-r}$$

so

$$\begin{aligned}
& K_1(w_1, w_1) \\
&= \sum_{r=0}^{\alpha} (-1)^r C_\alpha^r \exp\left(j\varepsilon\left(r - \frac{\alpha}{2}\right)\right) \\
&= \sum_{r=0}^{\frac{\alpha-1}{2}} (-1)^r C_\alpha^r \left(\exp\left(j\varepsilon\left(r - \frac{\alpha}{2}\right)\right) \right. \\
&\quad \left. - \exp\left(j\varepsilon\left(\frac{\alpha}{2} - r\right)\right) \right) \\
&= -2j \sum_{r=0}^{\frac{\alpha-1}{2}} (-1)^r C_\alpha^r \sin\left(\varepsilon\left(\frac{\alpha}{2} - r\right)\right).
\end{aligned}$$

Thus, we have $K_1^* = -K_1$, so $\tilde{D}_{\alpha x}^* = -\tilde{D}_{\alpha x}$. The same method is used to prove the symmetry of $\tilde{D}_{\alpha y}$. \square

The symmetry is lost when α is a noninteger. In Section IV, we use this property to analyze the folded algorithm.

Having algorithms of $\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha x}^*$, we can easily obtain our noise removal algorithm. Let

$$g = \tilde{D}_{\alpha x}^*(c(|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha x}u) + \tilde{D}_{\alpha y}^*(c(|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha y}u)$$

where $\tilde{D}_\alpha u = (\tilde{D}_{\alpha x}u, \tilde{D}_{\alpha y}u)$, then

$$\begin{aligned}
\hat{g} &= K_1^* \circ F(c(|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha x}u) \\
&\quad + K_2^* \circ F(c(|\tilde{D}_\alpha u|^2)\tilde{D}_{\alpha y}u) \quad (24)
\end{aligned}$$

where

$$\begin{aligned}
K_1^* &= \text{diag}(\text{conj}((1 - \exp(-j2\pi w_1/m))^\alpha \\
&\quad \times \exp(j\pi\alpha w_1/m))), \\
K_2^* &= \text{diag}(\text{conj}((1 - \exp(-j2\pi w_2/m))^\alpha \\
&\quad \times \exp(j\pi\alpha w_2/m))).
\end{aligned}$$

We compute the evolution of the initial image u , along flow (14), work in the Fourier domain, only returning to the spatial domain when computing $h_{xn} = c(|\tilde{D}_\alpha u_n|^2)\tilde{D}_{\alpha x}u_n$ and $h_{yn} = c(|\tilde{D}_\alpha u_n|^2)\tilde{D}_{\alpha y}u_n$.

To summarize, our noise removal approach is done in following steps.

- 1) Let the input image be u and set $n = 1$, $u_n = u$, k , Δt , $t = k\Delta t$, compute the 2-D DFT \hat{u}_n of u_n .
- 2) Compute α -order partial differences $\tilde{D}_{\alpha x}u_n$ and $\tilde{D}_{\alpha y}u_n$ using (21).
- 3) Compute $h_{xn} = c(|\tilde{D}_\alpha u_n|^2)\tilde{D}_{\alpha x}u_n$ and $h_{yn} = c(|\tilde{D}_\alpha u_n|^2)\tilde{D}_{\alpha y}u_n$ in the spatial domain.
Compute $\hat{g}_n = K_1^* \circ F(h_{xn}) + K_2^* \circ F(h_{yn})$.
- 4) Compute $\hat{u}_{n+1} = \hat{u}_n - \hat{g}_n \times \Delta t$ and set $n = n + 1$; if $n = k$, compute the 2-D IDFT of \hat{u}_n , stop; else go to 2).

When image size m is 512, the complex component is very small so that it can be completely neglected in practical computations and does not cause any effects on iterative images. If we

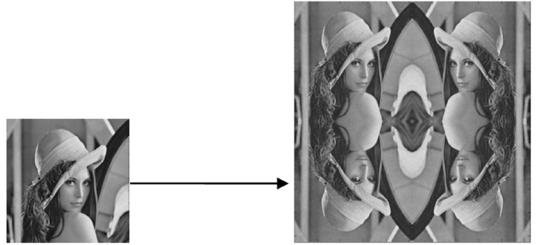


Fig. 1. Extending the image symmetrically about its borders in order to reduce discontinuities across the image borders due to the periodization.

want to eliminate this small complex component, we can extend the observed image u using the following method:

$$\begin{aligned}
uo(x, y) &= u(x, y), \quad 0 \leq x, y < m \\
uo(x, m) &= u(x, m - 1), \quad 0 \leq x < m \\
uo(m, y) &= u(m - 1, y), \quad 0 \leq y < m \\
uo(m, m) &= u(m - 1, m - 1).
\end{aligned}$$

The size of the extended image uo is $m + 1$ which is an odd integer. Thus, the complex component goes away in practical computations when applying the proposed algorithm to uo .

IV. FOLDED ALGORITHM

Since the use of the discrete Fourier transform automatically imposes the period boundary condition on the proposed equations, jump discontinuities could result across the image borders. To overcome this problem, we use a similar algorithm in [32] by extending the image symmetrically about its borders.

Let us consider an $m \times m$ input image $u(x, y)$, $0 \leq x, y < m$. Fold this image with respect to the lines $x = m - 1$ and $y = m - 1$ to produce an $2m \times 2m$ image $\tilde{u}(x, y)$, in which (also see Fig. 1)

$$\begin{aligned}
\tilde{u}(x, y) &= u(x, y), \quad 0 \leq x, y < m \\
\tilde{u}(x, y) &= \tilde{u}(2m - 1 - x, y) \\
&= \tilde{u}(x, 2m - 1 - y) \\
&= \tilde{u}(2m - 1 - x, 2m - 1 - y) \\
&\quad 0 \leq x, y < 2m.
\end{aligned}$$

The periodization of $\tilde{u}(x, y)$ is symmetric and continuous across borders of the original image, thus eliminating the jump discontinuity that would have resulted from the periodic extension of $u(x, y)$. The denoised image $\bar{u}(x, y)$ can be obtained by applying the proposed algorithm to $\tilde{u}(x, y)$. We then calculate the finally denoised image from four small images in the following way:

$$\begin{aligned}
ud(x, y) &= \frac{1}{4} \times (\bar{u}(x, y) + \bar{u}(2m - 1 - x, y) \\
&\quad + \bar{u}(x, 2m - 1 - y) \\
&\quad + \bar{u}(2m - 1 - x, 2m - 1 - y)) \\
&\quad 0 \leq x, y < m. \quad (25)
\end{aligned}$$

For these four small images, we have the following analysis. When α is an even integer, we have $\tilde{D}_{\alpha x} =$

$\tilde{D}_{\alpha x}^*$ and $\tilde{D}_{\alpha y} = \tilde{D}_{\alpha y}^*$, which implies $\tilde{D}_{\alpha x}\tilde{u}(x, y) = \tilde{D}_{\alpha x}\tilde{u}(2m - 1 - x, y)$ and $\tilde{D}_{\alpha y}\tilde{u}(x, y) = \tilde{D}_{\alpha y}\tilde{u}(2m - 1 - x, y)$, so diffusion coefficients (see the first equation shown at the bottom of the page). When α is an odd integer, we have $\tilde{D}_{\alpha x} = -\tilde{D}_{\alpha x}^*$ and $\tilde{D}_{\alpha y} = -\tilde{D}_{\alpha y}^*$, which implies $\tilde{D}_{\alpha x}\tilde{u}(x, y) = -\tilde{D}_{\alpha x}\tilde{u}(2m - 1 - x, y)$ and $\tilde{D}_{\alpha y}\tilde{u}(x, y) = -\tilde{D}_{\alpha y}\tilde{u}(2m - 1 - x, y)$, so diffusion coefficients (see the second equation shown at the bottom of the page).

The symmetry is lost when α is a noninteger, which implies $\tilde{D}_{\alpha x}^2\tilde{u}(x, y) = \tilde{D}_{\alpha x}^2\tilde{u}(x, 2m - 1 - y) \neq \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, y) = \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, 2m - 1 - y)$ and $\tilde{D}_{\alpha y}^2\tilde{u}(x, y) = \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, y) \neq \tilde{D}_{\alpha y}^2\tilde{u}(x, 2m - 1 - y) = \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, 2m - 1 - y)$, so diffusion coefficients (see the third equation shown at the bottom of the page).

We also note that if the linear diffusion

$$\frac{\partial u}{\partial t} = -D_{\alpha x}^*(D_{\alpha x}u) - D_{\alpha y}^*(D_{\alpha y}u)$$

is used, these four small images are always the same due to symmetries of $\tilde{D}_{\alpha x}^*\tilde{D}_{\alpha x}$ and $\tilde{D}_{\alpha y}^*\tilde{D}_{\alpha y}$. Thus, when α is an integer, these four small images are the same anyway; when α is a noninteger, these four small images are not the same due to different diffusion coefficients, so the average between four small images is less sensitive to additive noise. Numerical experiments also prove our conclusions.

V. NUMERICAL EXPERIMENTS

In this section, we present numerical results obtained by applying our proposed new anisotropic diffusion models to image denoising. We test the proposed algorithm on “Lena,” “Fishing Boat,” “Peppers,” and “Barbara” images with size 512×512 . In our implementation, α is kept as a parameter. We will show numerical results obtained with various values of α . We use the function $c(s) = 1/(s + 1)$ as diffusion coefficients and employ the signal-to-noise ratio (SNR) to quantify the denoising performance. We use an order-dependent step size $\Delta t = 4^{-\alpha}$ because the difference scheme is stable in the l^2 norm. The proof is illustrated in the Appendix.

In Table I, we show results obtained with different fractional orders α across three noise levels on four different images. Original images and noisy images are listed in Fig. 2. The

$$\begin{aligned} & \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(x, y) + \tilde{D}_{\alpha y}^2\tilde{u}(x, y)} \\ &= \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(x, 2m - 1 - y) + \tilde{D}_{\alpha y}^2\tilde{u}(x, 2m - 1 - y)} \\ &= \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, y) + \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, y)} \\ &= \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, 2m - 1 - y) + \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, 2m - 1 - y)} \end{aligned}$$

$$\begin{aligned} & \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(x, y) + \tilde{D}_{\alpha y}^2\tilde{u}(x, y)} \\ &= \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(x, 2m - 1 - y) + \tilde{D}_{\alpha y}^2\tilde{u}(x, 2m - 1 - y)} \\ &= \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, y) + \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, y)} \\ &= \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, 2m - 1 - y) + \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, 2m - 1 - y)} \end{aligned}$$

$$\begin{aligned} & \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(x, y) + \tilde{D}_{\alpha y}^2\tilde{u}(x, y)} \\ &\neq \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(x, 2m - 1 - y) + \tilde{D}_{\alpha y}^2\tilde{u}(x, 2m - 1 - y)} \\ &\neq \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, y) + \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, y)} \\ &\neq \frac{1}{1 + \tilde{D}_{\alpha x}^2\tilde{u}(2m - 1 - x, 2m - 1 - y) + \tilde{D}_{\alpha y}^2\tilde{u}(2m - 1 - x, 2m - 1 - y)} \end{aligned}$$

TABLE I
SNR WITH DIFFERENT CHOICES OF α ACROSS THREE NOISE LEVELS

Image	Lenna						Fishing Boat					
	α	1	1.2	1.4	1.6	1.8	2	1	1.2	1.4	1.6	1.8
$\sigma = 15$	16.91	17.21	17.39	17.51	17.65	17.63	14.74	14.93	15.02	15.07	15.13	15.02
$\sigma = 20$	15.55	15.87	16.09	16.26	16.39	16.20	13.35	13.55	13.66	13.73	13.78	13.56
$\sigma = 25$	14.51	14.84	15.08	15.25	15.39	15.05	12.30	12.53	12.65	12.72	12.75	12.45
Image	Barbara						Peppers					
	$\sigma = 15$	14.25	14.35	14.33	14.37	14.46	14.44	17.59	17.82	18.01	18.29	18.49
$\sigma = 20$	12.52	12.61	12.60	12.63	12.71	12.63	16.33	16.59	16.84	17.12	17.29	17.06
$\sigma = 25$	11.29	11.38	11.39	11.41	11.45	11.30	15.34	15.64	15.86	16.14	16.30	15.96

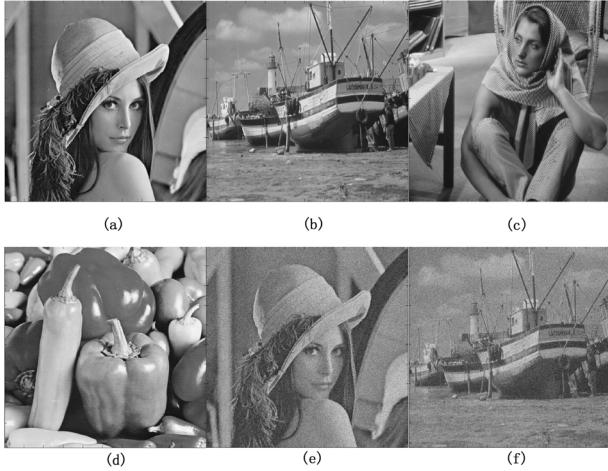


Fig. 2. Various Images used for performance testing. (a) Original Lena image. (b) Original Fishing Boat image. (c) Original Barbara image. (d) Original Peppers image. (e) Corrupted image $\sigma = 25$. (f) Corrupted image $\sigma = 25$.

stopping time was chosen so that the best SNR is obtained because this criterion was used in many references (for example, [30]). The other criteria such as the signal-noise decorrelation criterion [33] can also be used. In Table II, we list the optimal stopping time t of ‘‘Lena’’ and ‘‘Fishing Boat’’ images. As we expect, the bigger the variance of white noise, the longer stopping times are. We also find that the optimal stopping times of different differential-order α are almost the same except $\alpha = 2$. The optimal stopping times of $\alpha = 2$ are slightly longer than the others, which is due to that the fourth-order anisotropic model introduces speckle noise. In Figs. 3 and 4, we list denoised ‘‘Lena’’ and ‘‘Fishing Boat’’ images with different fractional orders α . The denoised images exhibit blocky effects when the second-order PDE is used and appear speckles when the fourth-order PDE is used. The proposed fractional-order pseudo-PDEs obtain better SNR and visual effects. We also report results using the folded algorithm in Table III. The folded algorithm gives better SNR than the original algorithm. We also note that the noninteger-order models make clear improvements because the average is less sensitive to noise. This can be regarded as a clear advantage of the fractional-order anisotropic diffusion.

In order to show the (damaging) effects of the smoothing schemes, we intentionally choose a large value for iteration number k so that the smoothing effects are easy to see. Fig. 5 shows evolution of the noisy ‘‘Lena’’ image with additive Gaussian white noise of $\sigma = 25$ using the second-order PDE ($k = 2500$), the proposed third-order pseudo-PDE ($k = 5000$), and the fourth-order PDE ($k = 10000$). We also list enlargements of faces. To test our algorithm sufficiently, we use three different models to smooth a real satellite image. The smoothed images are listed in Fig. 6. Image size is 512×512 . Differences between three schemes are obvious. The images resulting from the second-order PDE look ‘‘blocky’’ and produce false edges especially on Lena’s face. The denoised images using the fourth-order PDE suffer from speckle artifacts which are isolated black and white speckles. The images resulting from the proposed third-order pseudo-PDE look natural and do not produce false edges. Speckle artifacts are also avoided.

In our next experiment, we compare our algorithm with the common Perona–Malik algorithm. The essential difference between two implementations is the boundary condition. The Perona–Malik algorithm uses the Neumann boundary condition which ensures that there is no grey-value flux through the boundary, and yet our algorithm considers the input image as a periodic image. Another difference is difference schemes: Our algorithm uses the central difference scheme, and yet for the Perona–Malik algorithm, we use forward and backward difference schemes. In Table IV, we list the SNR using two different algorithms. Our algorithm obtains higher SNR than the common Perona–Malik algorithm. We also note that the bigger the variance of white noise, the worse the results using the Perona–Malik algorithm are. This is due to that the Perona–Malik algorithm cannot remove large oscillations, so the regularization of the Perona–Malik model was proposed in [2]. In Fig. 7, we list denoised images using two algorithms. We can see that the images resulting from the Perona–Malik algorithm suffer from speckles which are severely noise-contaminated points, and yet our algorithm reduces this effect greatly. In that sense, our algorithm has an advantage in comparison with the common Perona–Malik algorithm. This is owing to different difference schemes of two algorithms. The reason may be explained as follows (see Fig. 8). For simplicity, we only illustrate a 1-D example. Let A be a severely noise-contaminated point which

TABLE II
OPTIMAL STOPPING TIME OF “LENA” AND “FISHING BOAT” IMAGES

Image	Lenna						Fishing Boat					
	α	1	1.2	1.4	1.6	1.8	2	1	1.2	1.4	1.6	1.8
$\sigma = 15$	206	205	207	207	204	218	203	200	199	198	198	200
$\sigma = 20$	359	356	355	353	356	380	349	347	344	342	343	354
$\sigma = 25$	546	545	541	539	544	586	533	530	528	525	525	556



Fig. 3. Comparison of results from different α with additive Gaussian white noise of $\sigma = 20$ on the Lena image. (a) $\alpha = 1$, (b) $\alpha = 1.2$, (c) $\alpha = 1.4$, (d) $\alpha = 1.6$, (e) $\alpha = 1.8$, and (f) $\alpha = 2$.

has very high grey-value. B and C are its neighboring pixels. D and E are trigonometric interpolation points. When the forward difference scheme is used to compute the norm of the gradient, the diffusion coefficient of point A is $1/((\text{valueC} - \text{valueA})^2 + 1)$ which is very small, so the noise point A is kept. When the central difference scheme is used, the diffusion coefficient of point A is $1/((\text{valueE} - \text{valueD})^2 + 1)$ which has much higher value, so the noise point A is smoothed.

Finally, we will study the denoising performance to the selection of α . In Fig. 9, we plot the SNR with different α from 0 to 3. We can observe the following: The SNR reaches a maximum at $\alpha = 1.8$ and $\alpha = 2.2$; the SNR at $\alpha = 1$ is lower than

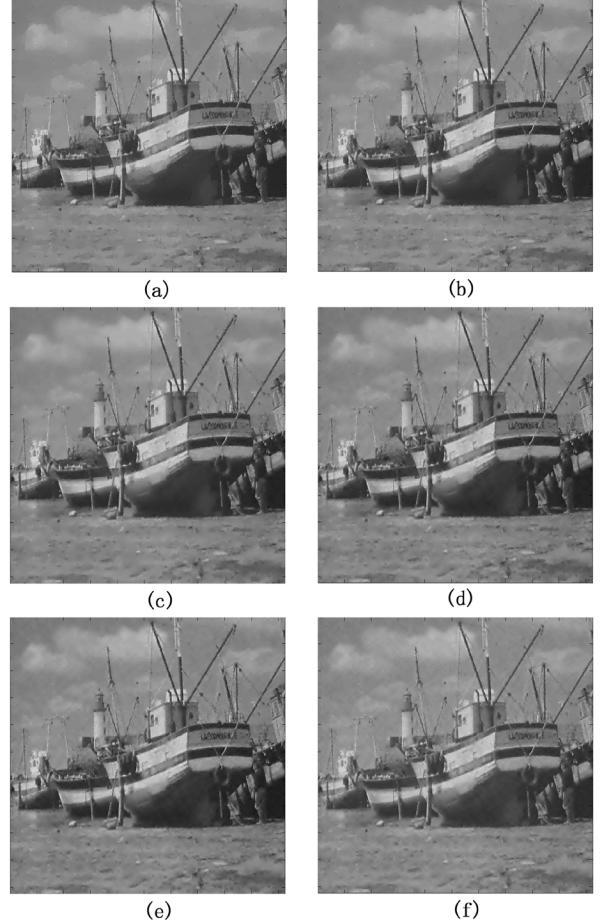


Fig. 4. Results corresponding to the Fishing Boat image.

SNR at $\alpha > 1$, which is owing to the severely blocky effect; a local minimum point appears at $\alpha = 2$ owing to speckle artifacts, however, the noninteger-order models suffer from very few speckle artifacts; when $\alpha \rightarrow 0$, the SNR decreases rapidly and when $\alpha = 0$, the SNR is equal to the SNR of the noisy image.

VI. CONCLUSION

A class of fractional-order anisotropic diffusion models were derived as a process that seeks to minimize a functional proportional to the absolute value of the fractional derivative of the image intensity function. We can choose any positive number α as the differential order, so second-order and fourth-order anisotropic diffusion models can be seen as special examples of our proposed models. We give an iterative algorithm in the

TABLE III
COMPARE THE FOLDED ALGORITHM WITH THE ORIGINAL ALGORITHM WITH ADDITIVE GAUSSIAN WHITE NOISE OF $\sigma = 15$

Image	Lenna						Fishing boat					
	α	1	1.2	1.4	1.6	1.8	2	1	1.2	1.4	1.6	1.8
Folded	17.05	17.94	18.20	18.27	18.25	17.73	14.79	15.43	15.58	15.59	15.54	15.05
Original	16.91	17.21	17.39	17.51	17.65	17.63	14.74	14.93	15.02	15.07	15.13	15.02
Barbara						Peppers						
Folded	14.31	14.86	14.88	14.89	14.89	14.48	17.89	18.73	18.93	18.99	18.92	18.48
Original	14.25	14.35	14.33	14.37	14.46	14.44	17.59	17.82	18.01	18.29	18.49	18.42



Fig. 5. Evolution of the noisy Lena image with additive Gaussian white noise of $\sigma = 25$. (a) $\alpha = 1$, (b) $\alpha = 1.5$, (c) $\alpha = 2$, (d) $\alpha = 1$ (enlargement of the Lena's face), (e) $\alpha = 1.5$ (enlargement of the Lena's face), and (f) $\alpha = 2$ (enlargement of the Lena's face).

frequency domain using the 2-D DFT and apply it to image denoising. Although the use of the 2-D DFT leads to consider the input image as a periodic image, we can apply the folded algorithm to overcome this drawback. The main advantage of our algorithm is that the proposed pseudo-PDEs exhibit higher perceptual quality than second-order and fourth-order PDEs. The second-order PDE ($\alpha = 1$) preserves edges but has the sometimes undesirable staircase effect; the fourth-order PDE ($\alpha = 2$) avoids the blocky effect but suffers from speckle

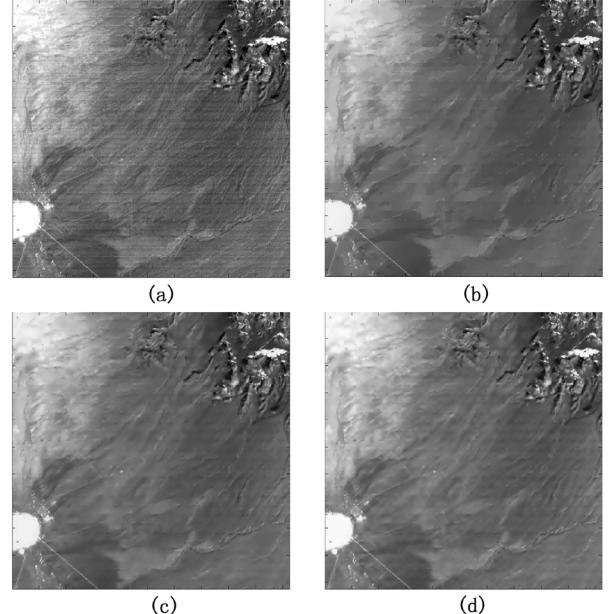


Fig. 6. Evolution of a real noisy satellite image using three different models. (a) original image, (b) $\alpha = 1$ ($k = 450$), (c) $\alpha = 1.5$ ($k = 900$), and (d) $\alpha = 2$ ($k = 1800$).

artifacts; the proposed third-order pseudo-PDE ($\alpha = 1.5$) avoids the blocky effect and has no speckle artifacts. Another advantage is that the average between four small images can improve the SNR, however, the SNR cannot be improved if the integer-order models are used. Moreover, we can apply the central difference scheme to any order pseudo-PDEs. The central difference scheme can obtain better results than forward and backward difference schemes. Future works involve extending the proposed method to other PDEs (for example [34] and [35]) and variational models [36], [9], [28].

APPENDIX STABILITY ANALYSIS OF THE DIFFERENCE SCHEME

In order to analyze the stability of the difference scheme, we freeze diffusion coefficients and see diffusion coefficients as constant c . Although this is not too acceptable from a rigorous point of view, in fact, it is an efficient method.

TABLE IV
COMPARE OUR ALGORITHM WITH THE COMMON PERONA–MALIK ALGORITHM

Image	Lenna			Fishing Boat		
σ	15	20	25	15	20	25
Perona	16.11	14.51	13.35	14.20	12.63	11.47
$\alpha = 1$	16.91	15.55	14.51	14.74	13.35	12.30

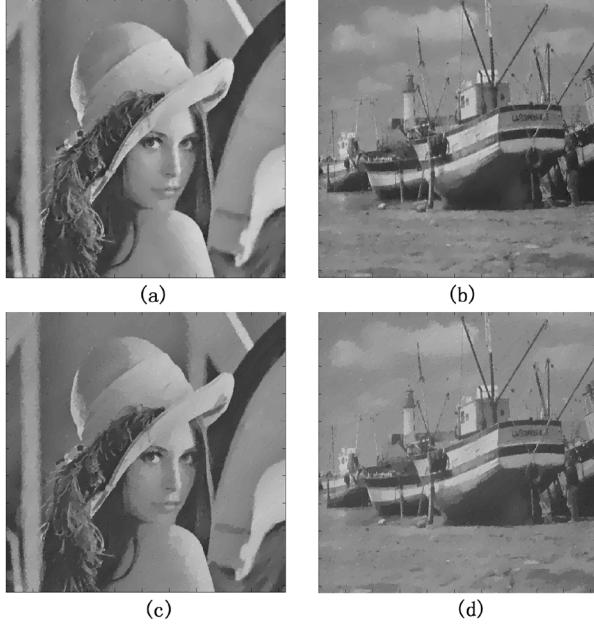


Fig. 7. Comparison of two algorithms with additive Gaussian white noise of $\sigma = 25$; (a) and (b) are by our algorithm and (c) and (d) are by the Perona–Malik algorithm.

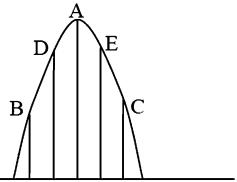


Fig. 8. Comparison of two different difference schemes.

Let $v(w) = (1 - \exp(-j2\pi w/m))^\alpha \exp(j\pi\alpha w/m)$, we have

$$\begin{aligned}
 \|u_{n+1}(x, y)\|_2 &= \|\hat{u}_{n+1}(w_1, w_2)\|_2 \\
 &= \|(1 - \Delta t \cdot \text{conj}(v(w_1)) \cdot c \cdot v(w_1) \\
 &\quad - \Delta t \cdot \text{conj}(v(w_2)) \cdot c \cdot v(w_2)) \hat{u}_n(w_1, w_2)\|_2 \\
 &= \|(1 - \Delta t \cdot c \cdot |v(w_1)|^2 \\
 &\quad - \Delta t \cdot c \cdot |v(w_2)|^2) \hat{u}_n(w_1, w_2)\|_2 \\
 &= \|(1 - \Delta t \cdot c \cdot |1 - \exp(-j2\pi w_1/m)|^{2\alpha} \\
 &\quad - \Delta t \cdot c \cdot |1 - \exp(-j2\pi w_2/m)|^{2\alpha}) \hat{u}_n(w_1, w_2)\|_2 \\
 &= \|(1 - \Delta t \cdot c \cdot (4 \sin^2(-\pi w_1/m))^\alpha \\
 &\quad - \Delta t \cdot c \cdot (4 \sin^2(-\pi w_2/m))^\alpha) \hat{u}_n(w_1, w_2)\|_2
 \end{aligned}$$

where $\|\cdot\|_2$ denotes the l^2 norm. So, the stability condition will be satisfied as soon as

$$\begin{aligned}
 &|1 - \Delta t \cdot c \cdot (4 \sin^2(-\pi w_1/m))^\alpha \\
 &\quad - \Delta t \cdot c \cdot (4 \sin^2(-\pi w_2/m))^\alpha| \leq 1 \\
 &\text{for all } -\frac{m}{2} \leq w_1, w_2 \leq \frac{m}{2} - 1.
 \end{aligned}$$

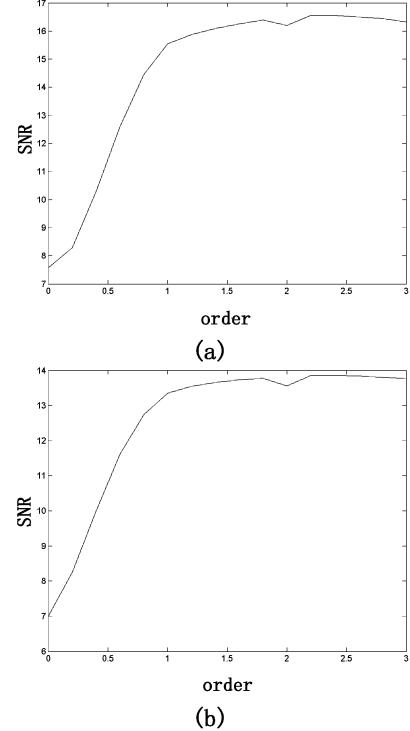


Fig. 9. SNR as a function of α on the Lena and Fishing Boat images with additive Gaussian white noise of $\sigma = 20$. (a) Lena image. (b) Fishing Boat image.

Thus, we have $\Delta t \cdot c \leq 4^{-\alpha}$. Since the function c that have an upper bound of 1 we have the stability condition $\Delta t \leq 4^{-\alpha}$ in the l^2 norm.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers whose comments helped to improve the final version of the manuscript.

REFERENCES

- [1] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.
- [2] F. Catte, P. L. Lions, J. M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numer. Anal.*, vol. 29, no. 1, pp. 182–193, 1992.
- [3] R. A. Carmona and S. Zhong, "Adaptive smoothing respecting feature directions," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 353–358, Mar. 1998.
- [4] S.-H. Lee and J. K. Seo, "Noise removal with Gauss curvature-driven diffusion," *IEEE Trans. Image Process.*, vol. 14, no. 7, pp. 904–909, Jul. 2005.
- [5] K. Chen, "Adaptive smoothing via contextual and local discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1552–1567, Oct. 2005.

- [6] J. Monteil and A. Beghdadi, "A new interpretation and improvement of the nonlinear anisotropic diffusion for image enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 940–946, Sep. 1999.
- [7] D. Barash, "A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 844–847, Jun. 2002.
- [8] A. C.-C. Shih, H. Y. M. Liao, and C.-S. Lu, "A new iterated two-band diffusion equation: Theory and its application," *IEEE Trans. Image Process.*, vol. 12, no. 4, pp. 466–476, Apr. 2003.
- [9] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
- [10] A. Chambolle and P. L. Lions, "Image recovery via total variation minimization and related problems," *Numer. Math.*, vol. 76, no. 2, pp. 167–188, 1997.
- [11] P. Blomgren, P. Mulet, T. F. Chan, and C. K. Wong, "Total variation image restoration: numerical methods and extensions," in *Proc. Int. Conf. Image Processing*, 1997, vol. 3, pp. 384–387.
- [12] T. F. Chan, S. Esedoglu, and F. E. Park, "A fourth order dual method for staircase reduction in texture extraction and image restoration problems," [Online]. Available: <http://basilo.kaist.ac.kr/preprint/uclam/cam05-28.pdf>
- [13] T. F. Chan, A. Marquina, and P. Mulet, "High-order total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 22, no. 2, pp. 503–516, 2000.
- [14] M. Lysaker and X. C. Tai, "Iterative image restoration combining total variation minimization and a second-order functional," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 5–18, 2006.
- [15] Y.-L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1723–1730, Oct. 2000.
- [16] M. Lysaker, S. Osher, and X.-C. Tai, "Noise removal using smoothed normals and surface fitting," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1345–1357, Oct. 2004.
- [17] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric surface processing via normal maps," *ACM Trans. Graph.*, vol. 22, no. 4, pp. 1012–1033, 2003.
- [18] M. Lysaker, A. Lundervold, and X.-C. Tai, "Noise removal using Fourth-order partial differential equation with applications to medical magnetic resonance images in space and time," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1579–1590, Dec. 2003.
- [19] E. Cuesta and J. F. Codes, "Image processing by means of a linear integro-differential equation," in *Proc. 3rd IASTED Int. Conf. Visualization, Imaging, and Image Processing*, 2003, vol. 1, pp. 438–442.
- [20] R. Duits, M. Felsberg, L. Florack, and B. Platel, "α scale spaces on a bounded domain," in *Proc. 4th Int. Conf. Scale Spaces*, 2003, pp. 494–510.
- [21] S. Didas, B. Burgeth, A. Imiya, and J. Weickert, "Regularity and scale-space properties of fractional high order linear filtering," *Scale Spaces PDE Meth. Comput. Vis.*, pp. 13–25, 2005.
- [22] B. Ross, "A brief history and exposition of the fundamental theory of fractional calculus," in *Lecture Notes in Math.* New York: Springer-Verlag, 1975, vol. 457, pp. 40–130.
- [23] S. C. Liu and S. Chang, "Dimension estimation of discrete-time fractional Brownian motion with applications to image texture classification," *IEEE Trans. Image Process.*, vol. 6, no. 8, pp. 1176–1184, Aug. 1997.
- [24] B. Ninness, "Estimation of 1/f noise," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 32–46, Jan. 1998.
- [25] N. Engheta, "On the role of fractional calculus in electromagnetic theory," *IEEE Antennas Propagat. Mag.*, vol. 39, no. 4, pp. 35–46, Apr. 1997.
- [26] M. unser and T. Blu, "Fractional splines and wavelets," *SIAM Rev.*, vol. 42, no. 1, pp. 43–67, 2000.
- [27] M. Unser, "Splines, a perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Jun. 1999.
- [28] S. Osher, A. Solé, and E. Fatemi, "Image decomposition and restoration using total variation minimization and the H^{-1} norm," *Multiscale Model. Simul.*, vol. 1, no. 3, pp. 349–370, 2003.
- [29] S. Roudenko, "Noise and texture detection in image processing," [Online]. Available: <http://math.asu.edu/~svetlana/LANL-Imaging.pdf>
- [30] L. Lieu and L. A. Vese, "Image restoration and decomposition via bounded total variation and negative Hilbert–Sobolev spaces," [Online]. Available: http://www.math.ucdavis.edu/~lieu/ams_meeting1009_slides.pdf
- [31] R. A. Adams, "Sobolev spaces," in *Pure and Applied Mathematics*, ser. , Monographs and Textbooks. New York: Academic, 1975, vol. 65.
- [32] M. N. Do and M. Vetterli, "The finite ridgelet transform for image representation," *IEEE Trans. Image Process.*, vol. 12, no. 1, pp. 16–26, Jan. 2003.
- [33] P. Mrazek, "Selection of optimal stopping time for nonlinear diffusion filtering," *Int. J. Comput. Vis.*, vol. 52, no. 2/3, pp. 189–203, 2003.
- [34] T. F. Chan and J. H. Shen, "PDE models for image inpaintings and applications," in *Proc. Inf. Conf. CISST*, 2001, vol. 1, pp. 30–36.
- [35] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. SIGGRAPH*, New Orleans, LA, 2000, pp. 23–28.
- [36] Y. Meyer, "Oscillating patterns in image processing and nonlinear evolution equations," *Amer. Math. Soc.*, ser. Univ. Lecture, vol. 22, 2001.



Jian Bai was born in Xi'an, China, on December 15, 1979. He received the B.S. degree in applied mathematics from Xidian University, Xi'an, in 2002. He is currently pursuing the Ph.D. degree in applied mathematics at Xidian University.

His research interests include image processing, wavelets, ridgelets, curvelets, and partial differential equations.



Xiang-Chu Feng received the B.S. degree in computational mathematics from the Xian JiaoTong University, Xi'an, China, in 1984, and the M.S. and Ph.D. degrees in applied mathematics from Xidian University, Xi'an, in 1989 and 1999, respectively.

He is currently a Professor of applied mathematics at Xidian University. His research interests include numerical analysis, wavelets, and partial differential equations for image processing.