

Research Article

A Numerical Method for Preserving Curve Edges in Nonlinear Anisotropic Smoothing

Shaoxiang Hu,¹ Zhiwu Liao,² Dan Sun,² and Wufan Chen³

¹ School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

² School of Computer Science, Sichuan Normal University, Chengdu 610066, China

³ Institute of Medical Information and Technology, School of Biomedical Engineering, Southern Medical University, Guangzhou 510515, China

Correspondence should be addressed to Zhiwu Liao, liaozhiwu@163.com

Received 28 January 2011; Accepted 18 February 2011

Academic Editor: Ming Li

Copyright © 2011 Shaoxiang Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We focus on nonlinearity for images and propose a new method which can preserve curve edges in image smoothing using nonlinear anisotropic diffusion (NAD). Unlike existing methods which diffuse only among the spatial variants, the new method suggests that the diffusion should be performed both among the time variants and spatial variants, named time and space nonlinear anisotropic diffusion (TSNAD). That is, not only the differences of the spatial variants should be estimated by the nearby spatial points but also the differences of the time variants should be approximated by the weighted time differences of nearby points, according to the differences of gray levels between them and the consideration point. Since the time differences of nearby points using NAD can find more points with similar gray levels which form a curve belt for the center pixel on a curve edge, TSNAD can provide satisfied smoothing results while preserving curve edges. The experiments for digital images also show us the ability of TSNAD to preserve curve edges.

1. Introduction

Nonlinear systems are systems that cannot be mathematically described as the sum of their components, and their mathematical modeling is often very difficult or impossible. As a result, nonlinear systems are often studied through use of simulations [1]. Nonlinearity for many real applications has been widely discussed [2–11], and some methods based on NAD have been proposed in the early 90s [12–15]. Recently, some efforts for improving the performance of NAS have been proposed related to numerical method for partial differential function (PDF) theory, adaptive smoothing, and multiresolution analysis [16–22].

Although NAD provides perfect mathematical theory for preserving edges, how to preserve curve edges in image smoothing using NAD is still an unsolved problem [23, 24].

In a digital image, a curve edge is composed by discrete points instead of a continuous curve line. Moreover, the discrete points on a curve edge are composed by two types of points: one is edge and the other is corner. Note that most of curve edge points are corners instead of edge points since the curve or declining parts of a curve edge are composed by corners. Note that unlike the definition of corners in mathematics, a corner point defined in a digital image is a point with both large modules of gradients and tangents [12–14] or two large eigenvalues for the tensor matrix [15].

In order to preserve curve edges, some schemes are proposed for different types of points. For edge points, two famous diffusion schemes, which are diffusion only along tangent lines or diffusion along backward directions of the gradients, are proposed in [14, 15]. For corners, Alvarez et al. suggest to stop diffusion at [13].

However, these schemes lead to noise amplification, oscillation effects, and blurred curve edges [23, 24]. Analyzing behavior of these schemes, we can find that the diffusion for most of discrete points of a curve edge, corners, is stopped or only is carried on a very small scale. Therefore, the noise is amplified near edges, and corners are blurred even with a very small-scale diffusion.

Motivated by diffusion along tangents, we propose that the diffusion should be performed along curve edges to preserve curve edges. However, until now, we can not find a method to perform the diffusion along the curve edges. It is the result of one-step diffusion along tangent! Note that diffusion along the tangent means that the diffusion is performed on a straight line passing through the consideration point and with the direction of the tangent. Therefore, even with an elaborately selected scale, a straight line can not approximate a curve edge.

Fortunately, it is well known that a curve edge can be approximated well by multistraight lines. Although one-step method only gets one straight line, we have to design a multistep scheme to obtain multistraight lines. This multistep scheme can be realized when we try to approximate the differences of the time variant of the consideration point using the time differences of nearby points! Since each time difference relates to one straight diffusion line, the time differences of nearby points can find multistraight diffusion lines along the curve edge. Thus, it forms a curve belt along the curve edge!

Based on this theory, a new scheme, named time and space NAD (TSNAD), is proposed that approximating the difference of time variant uses the time difference of nearby points with NAD manner. That is, the time differences of nearby points are weighted by the absolute value of the gray differences between the nearby points and the consideration point. Then, all time differences are computed by the space differences according to the diffusion equation.

Since more time differences allow us to find more straight lines, TSNAD can find enough points with similar gray levels along the curve edges. Therefore, it can preserve curve edges well.

The rest of this paper is as follows. Section 2 is the method of TSNAD. In Section 3, the experimental results are given and discussed. We also give conclusions, future works, and finally acknowledgments.

2. The Method

Just as the above discussion, TSNAD provides a more NAD for time variant to preserve curve edges. In this section, we will introduce the theory of TSNAD in detail.

2.1. Motivation

The main objective for image smoothing is reducing image details or slightly noises without removing significant parts of the image content. Nonlinear anisotropic diffusion (NAD) is a well-known method in this task [12].

Formally, NAD is defined as

$$\frac{\partial u(x, y, t)}{\partial t} = \text{div}(g(x, y, t) \nabla u(x, y, t)), \quad (2.1)$$

where $u(x, y, 0)$ is the initial gray-scale image, $u(x, y, t)$ is the smooth gray-scale image at time t , ∇ denotes the gradient, $\text{div}(\cdot)$ is the divergence operator, and $g(x, y, t)$ is the diffusion coefficient. $g(x, y, t)$ controls the rate of diffusion and is usually chosen as a monotonically decreasing function of the module of the image gradient. Two functions proposed in [12] are

$$g(\|\nabla u(x, y, t)\|) = e^{-\|\nabla u(x, y, t)\|/\sigma}, \quad (2.2)$$

$$g(\|\nabla u(x, y, t)\|) = \frac{1}{1 + (\|\nabla u(x, y, t)\|/\sigma)^2}, \quad (2.3)$$

where $\|\cdot\|$ is the module of the vector, and the constant σ controls the sensitivity to edges.

Perona and Malik also propose a simple method to approach the modules of gradients which is called PM method as follows from the paper [12]. Its discretization for Laplacian operator is

$$u(i, j, t + 1) = u(i, j, t) + \frac{1}{4} \left[c_N \cdot \nabla_N^2 u(i, j, t) + c_S \cdot \nabla_S^2 u(i, j, t) + c_E \cdot \nabla_E^2 u(i, j, t) + c_W \cdot \nabla_W^2 u(i, j, t) \right], \quad (2.4)$$

where

$$\begin{aligned} \nabla_N^2 u(i, j, t) &= u(i - 1, j, t) - u(i, j, t), \\ \nabla_S^2 u(i, j, t) &= u(i + 1, j, t) - u(i, j, t), \\ \nabla_E^2 u(i, j, t) &= u(i, j + 1, t) - u(i, j, t), \\ \nabla_W^2 u(i, j, t) &= u(i, j - 1, t) - u(i, j, t). \end{aligned} \quad (2.5)$$

According to (2.2) and (2.3), the diffusion coefficient is defined as a function of module of the gradient. However, computing a gradient accurately in discrete data is very complex, and

the module of the gradient is simplified as the absolute values of four directions, and diffusion coefficients are

$$\begin{aligned}
 c_N(i, j, t) &= g(|\nabla_N u(i, j, t)|), \\
 c_S(i, j, t) &= g(|\nabla_S u(i, j, t)|), \\
 c_E(i, j, t) &= g(|\nabla_E u(i, j, t)|), \\
 c_W(i, j, t) &= g(|\nabla_W u(i, j, t)|),
 \end{aligned} \tag{2.6}$$

where $|\cdot|$ is the absolute value of the number and $g(\cdot)$ is defined in (2.2) or (2.3).

Although this scheme is not the exact discretization of (2.1) from the view of mathematics, it is the best way for preserving singularities in existing NAD methods since it can find neighbors with similar gray levels correctly without computing the gradients. Thus, it reduces the discretization errors greatly.

In addition, unlike other methods which have to distinguish corners from the edges, PM method allows to handle corners and edges using a unified scheme diffusion only among neighbors with similar gray levels. This advantage also decreases the influence of incorrect classification for corners and edges.

However, even for this best scheme in existing methods, how to preserve curve edges in image smoothing is an unsolved problem. The main difficulty for PM method and other existing methods is that approximation of a curve edge using a straight line segment (a line passing through the consideration point and along the tangent with a certain scale) is impossible.

Thus, we need a new scheme to find multistraight lines to approximate a curve edge. Intuitively, these multistraight lines should be related to some tangent lines passing through some of neighbors of the consideration points. Thus, these multistraight lines can be sought out using a quasimultisteps. That is, differences of the time variant are computed by the time differences of nearby pixels. It is obvious that each time difference relates to one tangent line. Thus, multitime differences relate to multitangent lines, which form a curve belt along the curve edge.

By this way, the new scheme, TSNAD, can provide satisfied smoothing results for preserving curve edges and deleting details.

2.2. The Space Difference

Existing methods approximate the time difference for the left hand of (2.1) by

$$\frac{\partial u(x, y, t)}{\partial t} = \frac{u(i, j, t+1) - u(i, j, t)}{\Delta t}, \tag{2.7}$$

where $u(i, j)$ is the consideration point, and t is the time variant.

The space difference for a two-dimension digital image $u(i, j)$ is defined as half-point differences between the center point $u(i, j)$ and the half points of its eight nearest neighbors

$$\nabla \mathbf{u}(i, j) = [\nabla_0 u(i, j), \nabla_1 u(i, j), \nabla_2 u(i, j), \nabla_3 u(i, j), \nabla_4 u(i, j), \nabla_5 u(i, j), \nabla_6 u(i, j), \nabla_7 u(i, j)]^T, \quad (2.8)$$

where T represents the transpose of the vector, and $\nabla u_k(i, j)$, $k = 0, \dots, 7$ are defined as

$$\begin{aligned} \nabla_0 u(i, j) &= u(i, j + 0.5) - u(i, j), & \nabla_1 u(i, j) &= u(i - 0.5, j + 0.5) - u(i, j), \\ \nabla_2 u(i, j) &= u(i - 0.5, j) - u(i, j), & \nabla_3 u(i, j) &= u(i - 0.5, j - 0.5) - u(i, j), \\ \nabla_4 u(i, j) &= u(i, j - 0.5) - u(i, j), & \nabla_5 u(i, j) &= u(i + 0.5, j - 0.5) - u(i, j), \\ \nabla_6 u(i, j) &= u(i + 0.5, j) - u(i, j), & \nabla_7 u(i, j) &= u(i + 0.5, j + 0.5) - u(i, j). \end{aligned} \quad (2.9)$$

Thus, the first-scale second-order difference of $u(i, j)$ is

$$\nabla^2 \mathbf{u}(i, j) = [\nabla_0^2 u(i, j), \nabla_1^2 u(i, j), \nabla_2^2 u(i, j), \nabla_3^2 u(i, j), \nabla_4^2 u(i, j), \nabla_5^2 u(i, j), \nabla_6^2 u(i, j), \nabla_7^2 u(i, j)]^T, \quad (2.10)$$

where T represents the transpose of the vector. From (2.7), we have

$$\begin{aligned} \nabla_0^2 u(i, j) &= u(i, j + 1) - u(i, j), & \nabla_1^2 u(i, j) &= u(i - 1, j + 1) - u(i, j), \\ \nabla_2^2 u(i, j) &= u(i - 1, j) - u(i, j), & \nabla_3^2 u(i, j) &= u(i - 1, j - 1) - u(i, j), \\ \nabla_4^2 u(i, j) &= u(i, j - 1) - u(i, j), & \nabla_5^2 u(i, j) &= u(i + 1, j - 1) - u(i, j), \\ \nabla_6^2 u(i, j) &= u(i + 1, j) - u(i, j), & \nabla_7^2 u(i, j) &= u(i + 1, j + 1) - u(i, j). \end{aligned} \quad (2.11)$$

Let

$$\mathbf{g}(i, j) = [g_0(i, j), g_1(i, j), g_2(i, j), g_3(i, j), g_4(i, j), g_5(i, j), g_6(i, j), g_7(i, j)]^T, \quad (2.12)$$

where T represents the transpose of the vector, and $g_k(i, j)$, $k = 0, \dots, 7$ are defined as

$$g_k(i, j) = g(|\nabla_k u(i, j)|), \quad k = 0, 1, \dots, 7, \quad (2.13)$$

where $\nabla_k u(i, j)$, $k = 0, \dots, 7$ defined in (2.8) are the components of vector $\nabla \mathbf{u}(i, j)$, and g is the decreasing function of absolute value of $\nabla_k u(i, j)$, $k = 0, \dots, 7$. Following (2.2) and (2.3), $g(|\nabla u_k(x, y, t)|)$ can be defined as

$$g(|\nabla u_k(i, j, t)|) = e^{-(|\nabla u_k(i, j, t)|/\sigma)^2}, \quad k = 0, \dots, 7, \quad (2.14)$$

or

$$g(|\nabla u_k(i, j, t)|) = \frac{1}{1 + (|\nabla u_k(i, j, t)|/\sigma)^2}, \quad k = 0, \dots, 7, \quad (2.15)$$

where $|\cdot|$ is the absolute value of the number, and the constant σ controls the sensitivity to edges. However, half-point difference for $\nabla \mathbf{u}(i, j)$ defined in (2.8) cannot be computed directly. Thus, the second-order integral-point difference can be used to approximate the first-order half-point difference. The (2.14) and (2.15) become

$$g(|\nabla u_k(i, j, t)|) = e^{-(|\nabla u_k(i, j, t)|/\sigma)^2} \approx e^{-(|\nabla^2 u_k(i, j, t)|/\sigma)^2}, \quad k = 0, \dots, 7, \quad (2.16)$$

$$g(|\nabla u_k(i, j, t)|) = \frac{1}{1 + (|\nabla u_k(i, j, t)|/\sigma)^2} \approx \frac{1}{1 + (|\nabla^2 u_k(i, j, t)|/\sigma)^2}, \quad k = 0, \dots, 7. \quad (2.17)$$

The NAD is converted to

$$\frac{\partial u(i, j, t)}{\partial t} = \lambda \operatorname{div} \begin{pmatrix} g_0(i, j) \nabla_0 u(i, j, t) \\ g_1(i, j) \nabla_1 u(i, j, t) \\ g_2(i, j) \nabla_2 u(i, j, t) \\ g_3(i, j) \nabla_3 u(i, j, t) \\ g_4(i, j) \nabla_4 u(i, j, t) \\ g_5(i, j) \nabla_5 u(i, j, t) \\ g_6(i, j) \nabla_6 u(i, j, t) \\ g_7(i, j) \nabla_7 u(i, j, t) \end{pmatrix}, \quad (2.18)$$

where λ is a constant to ensure the convergence of the iteration, the $\nabla_k u(i, j, t)$, $k = 0, \dots, 7$ are the components of vector $\nabla \mathbf{u}(i, j, t)$ in (2.10), and $g_k(i, j)$, $k = 0, \dots, 7$ defined in (2.13) are the components of $\mathbf{g}(i, j)$ in (2.12). Moreover, $g_k(i, j)$ in (2.13) can be represented by (2.16) or (2.17).

The above equation also is

$$\frac{\partial u(i, j, t)}{\partial t} = \lambda \sum_{k=0}^7 g_k(i, j) \nabla_k^2 u(i, j, t), \quad (2.19)$$

where λ is a constant to ensure the convergence of the iteration. $\nabla_k^2 u(i, j, t)$ is the second-order difference of the k th components of $u(i, j)$ which can be computed according to (2.10).

Thus, we have

$$\frac{\partial u(i, j, t)}{\partial t} = u(i, j, t+1) - u(i, j, t) = \lambda \sum_{k=0}^7 g_k(i, j) \nabla_k^2 u(i, j, t), \quad (2.20)$$

where $u(i, j, t+1)$ is the gray level of (i, j) at time $t+1$, and $g_k(i, j)$, $\nabla_k^2 u(i, j, t)$, $k = 0, \dots, 7$ are defined in the same way as in (2.19).

2.3. The New Time Difference

In this paper, we propose that not only the space differences but also the time differences should be estimated using NAD. That is, the time difference of $u(i, j, t)$ should be estimated using the time differences of nearby points. In this section, we will discuss the simplest form for new time differences approximated by the weighted time differences of its eight nearest neighbors.

Representing this approximation using the terms of high-dimensional vector analysis, a vector is defined as

$$\frac{\partial \mathbf{u}(\mathbf{i}, \mathbf{j}, \mathbf{t})}{\partial t} = \left[\frac{\partial u_0(i, j, t)}{\partial t}, \frac{\partial u_1(i, j, t)}{\partial t}, \frac{\partial u_2(i, j, t)}{\partial t}, \frac{\partial u_3(i, j, t)}{\partial t}, \frac{\partial u_4(i, j, t)}{\partial t}, \right. \\ \left. \frac{\partial u_5(i, j, t)}{\partial t}, \frac{\partial u_6(i, j, t)}{\partial t}, \frac{\partial u_7(i, j, t)}{\partial t} \right]^T, \quad (2.21)$$

where T represents the transpose of the vector, and $\partial u_k(i, j, t)/\partial t$, $k = 0, \dots, 7$ are defined as

$$\begin{aligned} \frac{\partial u_0(i, j, t)}{\partial t} &= \frac{\partial u(i, j+1, t)}{\partial t}, & \frac{\partial u_1(i, j, t)}{\partial t} &= \frac{\partial u(i-1, j+1, t)}{\partial t}, \\ \frac{\partial u_2(i, j, t)}{\partial t} &= \frac{\partial u(i-1, j, t)}{\partial t}, & \frac{\partial u_3(i, j, t)}{\partial t} &= \frac{\partial u(i-1, j-1, t)}{\partial t}, \\ \frac{\partial u_4(i, j, t)}{\partial t} &= \frac{\partial u(i, j-1, t)}{\partial t}, & \frac{\partial u_5(i, j, t)}{\partial t} &= \frac{\partial u(i+1, j-1, t)}{\partial t}, \\ \frac{\partial u_6(i, j, t)}{\partial t} &= \frac{\partial u(i+1, j, t)}{\partial t}, & \frac{\partial u_7(i, j, t)}{\partial t} &= \frac{\partial u(i+1, j+1, t)}{\partial t}. \end{aligned} \quad (2.22)$$

The diffusion coefficients for $\partial u_k(i, j, t)/\partial t$, $k = 0, \dots, 7$ are represented by a vector $\mathbf{g}(\mathbf{i}, \mathbf{j})$ defined in (2.12), and its components are defined in (2.16). Thus, the time difference of $u(i, j)$ is

$$\frac{\partial u(i, j, t)}{\partial t} = \lambda_1 \left[\frac{\partial u(i, j, t)}{\partial t} + \operatorname{div} \left(\mathbf{g}(\mathbf{i}, \mathbf{j})^T \frac{\partial \mathbf{u}(\mathbf{i}, \mathbf{j}, \mathbf{t})}{\partial t} \right) \right], \quad (2.23)$$

where T represents the transpose of the vector, λ_1 is a constant to ensure the convergence of the iteration, and $\partial \mathbf{u}(i, j, t) / \partial t$ is defined in (2.21). Thus, (2.23) is

$$\frac{\partial u(i, j, t)}{\partial t} = \lambda_1 \left[\frac{\partial u(i, j, t)}{\partial t} + \sum_{k=0}^7 g_k(i, j) \frac{\partial u_k(i, j, t)}{\partial t} \right], \quad (2.24)$$

where $\partial u_k(i, j, t) / \partial t$, $k = 0, \dots, 7$ are the components of vector $\partial \mathbf{u}(i, j, t) / \partial t$ defined in (2.21), and $g_k(i, j)$, $k = 0, \dots, 7$ are the components of vector $\mathbf{g}(i, j)$ defined in (2.12).

According to (2.24), the time difference of $u(i, j, t)$ is estimated by the weighed time difference of its neighbors according to the difference of the gray levels between them and $u(i, j, t)$. That is, the neighbors whose gray levels are more similar to $u(i, j, t)$ have larger weights.

2.4. Time and Space NAD (TSNAD)

Substituting (2.19) and (2.21) into (2.24), we have the equation of TSNAD,

$$\begin{aligned} \frac{\partial u(i, j, t)}{\partial t} &= \lambda_1 \left[\frac{\partial u(i, j, t)}{\partial t} + \sum_{k=0}^7 g_k(i, j) \frac{\partial u_k(i, j, t)}{\partial t} \right] \\ &= \lambda_1 \left[\lambda \sum_{k=0}^7 g_k(i, j) \nabla_k^2 u(i, j, t) + \lambda g_0(i, j) \sum_{k=0}^7 g_k(i, j+1, t) \nabla_k^2 u(i, j+1, t) \right. \\ &\quad + \lambda g_1(i, j) \sum_{k=0}^7 g_k(i-1, j+1, t) \nabla_k^2 u(i-1, j+1, t) \\ &\quad + \lambda g_2(i, j) \sum_{k=0}^7 g_k(i-1, j, t) \nabla_k^2 u(i-1, j, t) \\ &\quad + \lambda g_3(i, j) \sum_{k=0}^7 g_k(i-1, j-1, t) \nabla_k^2 u(i-1, j-1, t) \\ &\quad + \lambda g_4(i, j) \sum_{k=0}^7 g_k(i, j-1, t) \nabla_k^2 u(i, j-1, t) \\ &\quad + \lambda g_5(i, j) \sum_{k=0}^7 g_k(i+1, j-1, t) \nabla_k^2 u(i+1, j-1, t) \\ &\quad + \lambda g_6(i, j) \sum_{k=0}^7 g_k(i+1, j, t) \nabla_k^2 u(i+1, j, t) \\ &\quad \left. + \lambda g_7(i, j) \sum_{k=0}^7 g_k(i+1, j+1, t) \nabla_k^2 u(i+1, j+1, t) \right]. \quad (2.25) \end{aligned}$$

Let $\lambda_1 \times \lambda = \lambda$ be a constant to ensure the convergence of the iteration, (2.25) becomes

$$\begin{aligned}
\frac{\partial u(i, j, t)}{\partial t} = \lambda & \left[\sum_{k=0}^7 g_k(i, j) \nabla_k^2 u(i, j, t) + g_0(i, j) \sum_{k=0}^7 g_k(i, j+1, t) \nabla_k^2 u(i, j+1, t) \right. \\
& + g_1(i, j) \sum_{k=0}^7 g_k(i-1, j+1, t) \nabla_k^2 u(i-1, j+1, t) \\
& + g_2(i, j) \sum_{k=0}^7 g_k(i-1, j, t) \nabla_k^2 u(i-1, j, t) \\
& + g_3(i, j) \sum_{k=0}^7 g_k(i-1, j-1, t) \nabla_k^2 u(i-1, j-1, t) \\
& + g_4(i, j) \sum_{k=0}^7 g_k(i, j-1, t) \nabla_k^2 u(i, j-1, t) \\
& + g_5(i, j) \sum_{k=0}^7 g_k(i+1, j-1, t) \nabla_k^2 u(i+1, j-1, t) \\
& + g_6(i, j) \sum_{k=0}^7 g_k(i+1, j, t) \nabla_k^2 u(i+1, j, t) \\
& \left. + g_7(i, j) \sum_{k=0}^7 g_k(i+1, j+1, t) \nabla_k^2 u(i+1, j+1, t) \right],
\end{aligned} \tag{2.26}$$

where $g_k(i, j)$, $k = 0, \dots, 7$ defined in (2.13) are the components of vector $\mathbf{g}(\mathbf{i}, \mathbf{j})$ defined in (2.12), and $\nabla_k^2 u(i, j)$, $k = 0, \dots, 7$ are the components of $\nabla^2 \mathbf{u}(\mathbf{i}, \mathbf{j})$ defined in (2.10).

3. The Experiments and Discussion

In order to analyse the performance of TSNAD and compare it with existing methods, two images are selected in our experiments: one is a gray level image of an autotire, most of whose edges are curve edges (see Figure 1(d)); the other is a binary image with some test patterns including circles with different width, filled circles and some shapes with curve edges, and so forth (see Figure 2(d)). Both images are selected from the standard test images of MatLab.

Essentially, there are only three schemes for NAD until now: PM method [12, 13], backward filter [14], and coherence filter [15].

PM provides a simple discretization scheme for (2.1) by diffusion among points in four directions with similar gray levels to the consideration point. Although PM method is not the exact discretization of (2.1) from the view of mathematics, it has the best performance in existing methods because of the direct computing of diffusion coefficients using the differences of gray levels without computing the gradients [12].

Backward filter suggests that the diffusion near edges should be performed along the backward directions of the gradients [14], while coherence filter suggests that the diffusion

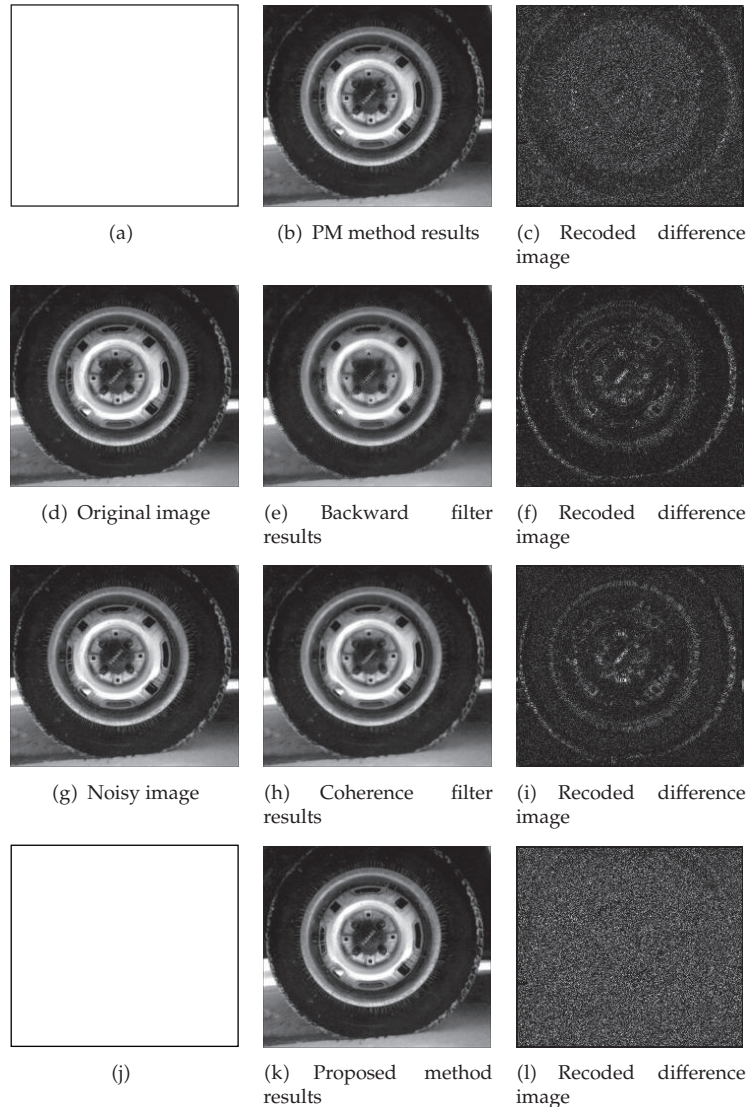


Figure 1: Smoothing results for the noisy image (g) and the recoded difference images between the relative smoothing images at their left and the original image (d) using different schemes.

should be performed along the tangents [15]. These two methods have to approximate the gradients and tangents using few discrete directions which leads to discretization errors. Moreover, since these two methods assume that the curve edges are composed by both corners and edges which should be handled separately, the diffusion has to stop at the corners or perform in a very small scale. The former will amplify noises at the corners while the latter will blur the corners. Therefore, both schemes can not provide satisfied smoothing results for curve edges.

TSNAD has all advantages for the above schemes. Firstly, its diffusion coefficients are computed according to PM method which can reduce the discretization errors greatly. Secondly, it adopts the scheme of diffusion along the curve edges to reduce the influence of

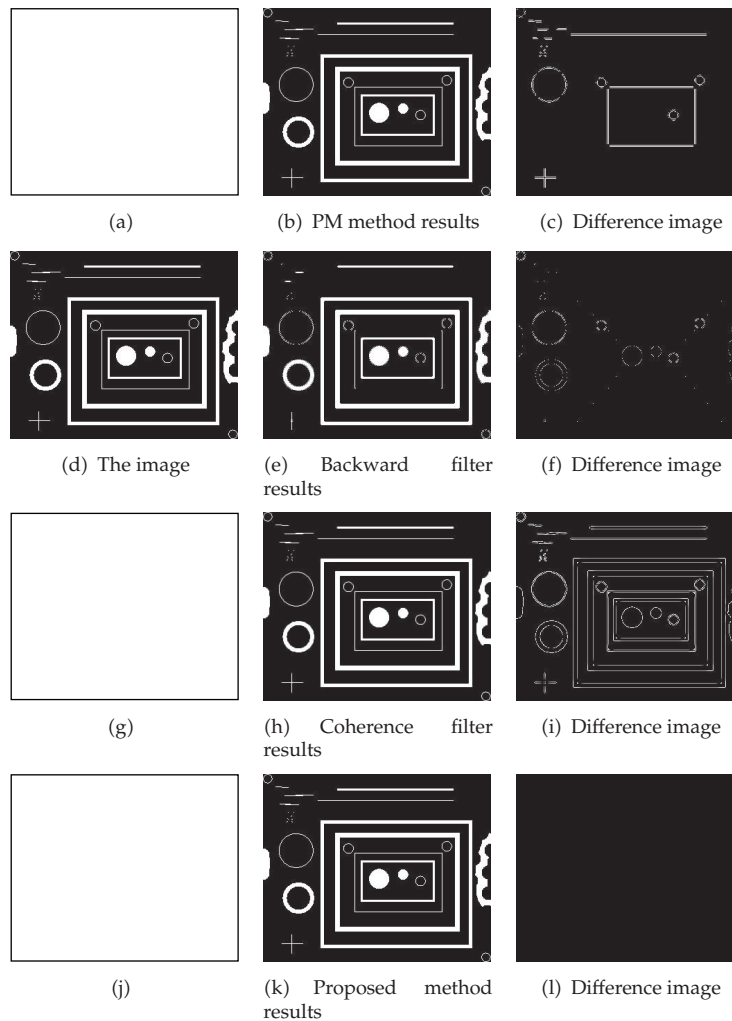


Figure 2: Smoothing results for the image (d) and the difference images between the relative smoothing images at their left and the original image (d) using different schemes.

the noises. Thirdly, it provides a more NAD for time variant to track the curve edges. Thus TSNAD has good performance both in image smoothing and curve edge preserving.

In the first image (see Figure 1(d)) is added a Gaussian white noise (GWN) with standard deviation 5 (see Figure 1(g)) to test the smoothing performance at slight noises. The NAD is performed on the noisy image, while the difference image is absolute difference value images between the relative smoothing images and the original image (see Figures 1(c), 1(f), 1(i), and 1(l)). Note that the difference images are *recorded* to show the relation in difference images more clearly. Thus, although Figure 1(l), which is the difference image between the smoothing image using TSNAD and the original image, has the much more white points, it does not mean that TSNAD has the biggest absolute difference to the original image because of “recorded” image. However, the difference images at least can provide what parts of the original image represented by white points are corrupted more seriously. For example, the white points in Figures 1(f) and 1(i) are all on the curve edges means that backward filter and

coherence filter will blur the curve edges, while the white points in Figure 1(l) are distributed very randomly means that the curve edges will be kept well.

The second image is a binary image with many test patterns (see Figure 2(d)). Observing the difference images in Figures 2(c), 2(f), 2(i), and 2(l), we can find that the difference image of the TSNAD has no white points means that it is very similar to the original image (see Figure 2(l)). Moreover, the difference image for the PM method has big differences at thinning edges (see Figure 2(c)), while the backward filter only lost curve edges (see Figure 2(f)). The difference image for the coherence filter shows that all edges for test patterns are lost (see Figure 2(i)).

From the above experiments and discussion, we can conclude that TANAD can preserve curve edge well in image smoothing.

4. Conclusions

In this paper, we propose TSNAD to preserve curve edges in image smoothing. Unlike existing methods which only perform NAD for the space variant, TSNAD also allows us to approximate the differences of time variant using NAD. The more time differences for NAD form curve belts along the curve edges, which can provide better fitness for curve edges. Thus, TSNAD can get satisfied results both in curve edge preserving and image smoothing.

5. Future Works

Recently, fractional-order transforms become a hot topic in many fields both in theory and in application [5–7, 25–31] for their attractive natures in image and signal processing. Thus, our future works will be devoted to the combination of NAD and the fractal; it includes the following.

- (1) It is a work trying the kernel of fractional order in (2.2) instead of power of 2 [32].
- (2) The future work will consider the fractional Gaussian noise (fGn) following [25–27, 32, 33]. In addition to fGn, other classes of Gaussian noise, such as the generalized Cauchy (GC) process [29–31], are also worth experiments. Further, multiscaled fGn or the GC noise [28–31] are worth noting in experiments.

Acknowledgments

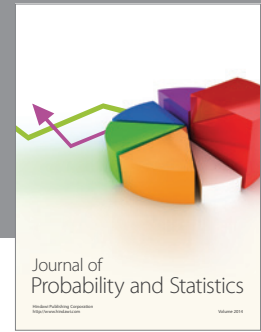
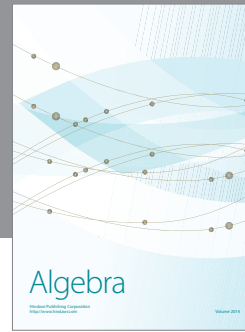
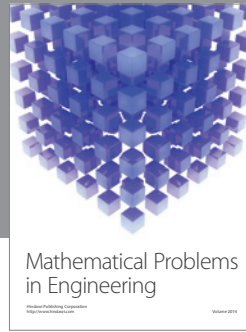
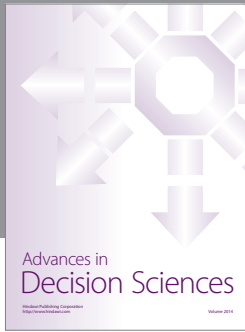
This paper is supported by the National Natural Science Foundation of China (nos. 60873102 and 60873264), Major State Basic Research Development Program (no. 2010CB732501), and Open Foundation of Visual Computing and Virtual Reality Key Laboratory Of Sichuan Province (no. J2010N03). This work was supported by a Grant from the National High Technology Research and Development Program of China (no. 2009AA12Z140).

References

- [1] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 2001.
- [2] J. V. Lorenzo-Ginori, K. N. Plataniotis, and A. N. Venetsanopoulos, "Nonlinear filtering for phase image denoising," *IEE Proceedings*, vol. 149, no. 5, pp. 290–296, 2002.

- [3] Y. J. Zhang and L. L. Geb, "Edge preserving regularization for the pde-based piecewise smooth mumford-shah algorithm," *Imaging Science Journal*, vol. 57, no. 3, pp. 119–127, 2009.
- [4] S. Y. Chen and Y. F. Li, "Determination of stripe edge blurring for depth sensing," *IEEE Sensors Journal*, vol. 11, no. 2, pp. 389–390, 2011.
- [5] Z. Jun and W. Zhihui, "A class of fractional-order multi-scale variational models and alternating projection algorithm for image denoising," *Applied Mathematical Modelling*, vol. 35, no. 5, pp. 2516–2528, 2011.
- [6] E. G. Bakhoun and C. Toma, "Dynamical aspects of macroscopic and quantum transitions due to coherence function and time series events," *Mathematical Problems in Engineering*, vol. 2010, Article ID 428903, 2010.
- [7] G. Mattioli, M. Scalia, and C. Cattani, "Analysis of large-amplitude pulses in short time intervals: application to neuron interactions," *Mathematical Problems in Engineering*, vol. 2010, Article ID 895785, 15 pages, 2010.
- [8] Y. Shih, C. Rei, and H. Wang, "A novel PDE based image restoration: convection-diffusion equation for image denoising," *Journal of Computational and Applied Mathematics*, vol. 231, no. 2, pp. 771–779, 2009.
- [9] C. Chaux, L. Duval, A. Benazza-Benyahia, and J. C. Pesquet, "A nonlinear Stein-based estimator for multichannel image denoising," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3855–3870, 2008.
- [10] S. Y. Chen, H. Tong, Z. Wang, S. Liu, M. Li, and B. Zhang, "Improved generalized belief propagation for vision processing," *Mathematical Problems in Engineering*, vol. 2011, Article ID 416963, 2011.
- [11] Z. Liao, S. Hu, and W. Chen, "Determining neighborhoods of image pixels automatically for adaptive image denoising using nonlinear time series analysis," *Mathematical Problems in Engineering*, vol. 2010, Article ID 914564, 14 pages, 2010.
- [12] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [13] L. Alvarez, P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. II," *SIAM Journal on Numerical Analysis*, vol. 29, no. 3, pp. 845–866, 1992.
- [14] L. Alvarez and L. Mazorra, "Signal and image restoration using shock filters and anisotropic diffusion," *SIAM Journal on Numerical Analysis*, vol. 31, no. 2, pp. 590–605, 1994.
- [15] J. Weickert, "Coherence-enhancing diffusion filtering," *International Journal of Computer Vision*, vol. 31, no. 2, pp. 111–127, 1999.
- [16] J. Yu, Y. Yang, and A. Campo, "Approximate solution of the nonlinear heat conduction equation in a semi-infinite domain," *Mathematical Problems in Engineering*, vol. 2010, Article ID 421657, 24 pages, 2010.
- [17] M. Li, M. Scalia, and C. Toma, "Nonlinear time series: computations and applications," *Mathematical Problems in Engineering*, vol. 2010, Article ID 101523, 5 pages, 2010.
- [18] N. Sochen, G. Gilboa, and Y. Y. Zeevi, "Color image enhancement by a forward and backward adaptive Beltrami flow," in *Proceedings of the 2nd International Workshop Algebraic Frames for the Perception-Action Cycle (AFPAC '00)*, G. Sommer and Y. Y. Zeevi, Eds., Lecture Notes in Computer Science 1888, pp. 319–328, Springer, 2000.
- [19] F. Zhang, Y. M. Yoo, L. M. Koh, and Y. Kim, "Nonlinear diffusion in laplacian pyramid domain for ultrasonic speckle reduction," *IEEE Transactions on Medical Imaging*, vol. 26, no. 2, pp. 200–211, 2007.
- [20] M. Ceccarelli, V. De Simone, and A. Murli, "Well-posed anisotropic diffusion for image denoising," *IEE Proceedings*, vol. 149, no. 4, pp. 244–252, 2002.
- [21] V. B. S. Prasath and A. Singh, "Well-posed inhomogeneous nonlinear diffusion scheme for digital image denoising," *Journal of Applied Mathematics*, vol. 2010, Article ID 763847, 14 pages, 2010.
- [22] J. Ling and A. C. Bovik, "Smoothing low-SNR molecular images via anisotropic median-diffusion," *IEEE Transactions on Medical Imaging*, vol. 21, no. 4, pp. 377–384, 2002.
- [23] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 123–139, 2008, special section: selection of papers for CVPR 2005, Guest Editors: C. Schmid, S. Soatto and C. Tomasi.
- [24] A. Buades, B. Coll, and J. M. Morel, "Image denoising methods. A new nonlocal principle," *SIAM Review*, vol. 52, no. 1, pp. 113–147, 2010.
- [25] M. Li, "Modeling autocorrelation functions of long-range dependent teletraffic series based on optimal approximation in Hilbert space-A further study," *Applied Mathematical Modelling*, vol. 31, no. 3, pp. 625–631, 2007.

- [26] C. Cattani, "Harmonic wavelet approximation of random, fractal and high frequency signals," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 207–217, 2010.
- [27] C. Cattani, "Fractals and hidden symmetries in DNA," *Mathematical Problems in Engineering*, vol. 2010, Article ID 507056, 31 pages, 2010.
- [28] M. Li, "Generation of teletraffic of generalized Cauchy type," *Physica Scripta*, vol. 81, no. 2, Article ID 025007, 2010.
- [29] M. Li and S. C. Lim, "Modeling network traffic using generalized Cauchy process," *Physica A*, vol. 387, no. 11, pp. 2584–2594, 2008.
- [30] M. Li and W. Zhao, "Representation of a Stochastic Traffic Bound," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1368–1372, 2010.
- [31] M. Li, W. Zhao, and S. Chen, "MBm-based scalings of traffic propagated in internet," *Mathematical Problems in Engineering*, vol. 2011, Article ID 389803, 21 pages, 2011.
- [32] M. Li, "Fractal time series—a tutorial review," *Mathematical Problems in Engineering*, vol. 2010, Article ID 157264, 26 pages, 2010.
- [33] B. B. Mandelbrot, *Gaussian Self-Affinity and Fractals*, Selected Works of Benoit B. Mandelbrot, Springer, New York, NY, USA, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

