

# Edge Structure Preserving 3D Image Denoising by Local Surface Approximation

Peihua Qiu and Partha Sarathi Mukherjee

**Abstract**—In various applications, including magnetic resonance imaging (MRI) and functional MRI (fMRI), 3D images are becoming increasingly popular. To improve the reliability of subsequent image analyses, 3D image denoising is often a necessary preprocessing step, which is the focus of the current paper. In the literature, most existing image denoising procedures are for 2D images. Their direct extensions to 3D cases generally cannot handle 3D images efficiently because the structure of a typical 3D image is substantially more complicated than that of a typical 2D image. For instance, edge locations are surfaces in 3D cases which would be much more challenging to handle compared to edge curves in 2D cases. We propose a novel 3D image denoising procedure in this paper, based on local approximation of the edge surfaces using a set of surface templates. An important property of this method is that it can preserve edges and major edge structures (e.g., intersections of two edge surfaces and pointed corners). Numerical studies show that it works well in various applications.

**Index Terms**—Edge-preserving image restoration, jump regression analysis, nonparametric regression, surface estimation.

## 1 INTRODUCTION

IN real life, most objects are three-dimensional (3D). Due to limitations of image acquisition techniques, people traditionally acquire a set of two-dimensional (2D) images from a sequence of slices of a 3D object (e.g., a patient's head), and then the 3D object is reconstructed from the 2D images to study the biological mechanism of the object. The related research area is termed 3D image reconstruction in the literature (e.g., [1, Chapters 11 and 12]). 3D image reconstruction is technically challenging, and the reconstructed 3D image often contains a substantial amount of estimation error. This limitation was recently lifted in certain applications, including magnetic resonance imaging (MRI) and functional MRI (fMRI), thanks to rapid progress in image acquisition techniques. We can now acquire 3D images directly in these applications, although observed 3D images often contain noise due to hardware imperfection and other reasons.

Noise removal is important for the reliability of subsequent image analyses, and is often one major focus during the preprocessing stage in image processing. However, most existing image denoising methods are for analyzing 2D images (e.g., [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]). Some of them can be extended to 3D cases, but their direct extensions may not be able to handle 3D images efficiently because the structure of a typical 3D image is often substantially more complicated than that of a typical 2D image. For instance, edge locations are surfaces in 3D cases which have much more complicated structures (e.g., intersections of two edge surfaces, pointed corners,

and so forth) compared to edge curves in 2D cases. Of course, a 3D image can also be denoised by first dividing the 3D image into slices of 2D images and then applying 2D image denoising procedures to the 2D slices. But, it would not be efficient to denoise 3D images in that way because the 3D spatial information is not used efficiently when denoising individual 2D images.

In the literature, a number of 3D image denoising procedures have been developed. Some of them are constructed after properly generalizing and/or modifying their 2D counterparts. For instance, 3D image denoising based on minimization of total variation (TV) has gained certain popularity in the literature (e.g., [16], [17], [18]), and the TV approach is initially suggested for denoising 2D images (e.g., [19]). Matlab programs for 3D image denoising using anisotropic diffusion have also been developed (e.g., [20]). Other existing 3D image denoising procedures include the ones based on 3D wavelet transformations (e.g., [21], [22], [23], [24]), nonlocal means (e.g., [25]), distance-weighted Wiener filtering (e.g., [26]), and so forth.

Besides noise removal, another important requirement for image denoising procedures is that they should preserve important image structures such as edges and major edge features (e.g., intersections of two edge surfaces, pointed corners, and so forth). Most of the existing image denoising procedures mentioned above can preserve parts of the edges where the curvature of edge surfaces is relatively small, but would usually blur or round certain edge features at places where the curvature of edge surfaces is relatively large. For instance, the discrete version of a 2D bilateral filtering procedure can be written as follows (cf., [27, Section 7.4.2]):

$$S^{(t+1)}(x, y) = \frac{1}{N^{(t+1)}(x, y)} \sum_{i=-k}^k \sum_{j=-k}^k S^{(t)}(x+i, y+j) w^{(t)}(x+i, y+j),$$

- The authors are with the School of Statistics, University of Minnesota, 224 Church St. SE, Minneapolis, MN 55455.  
E-mail: {qiuxx008, mukh0044}@umn.edu.

Manuscript received 17 Apr. 2010; revised 17 Apr. 2011; accepted 26 Sept. 2011; published online 20 Dec. 2011.

Recommended for acceptance by W. Förstner.

For information on obtaining reprints of this article, please send e-mail to: [tpami@computer.org](mailto:tpami@computer.org), and reference IEEECS Log Number TPAMI-2010-04-0295.

Digital Object Identifier no. 10.1109/TPAMI.2011.261.

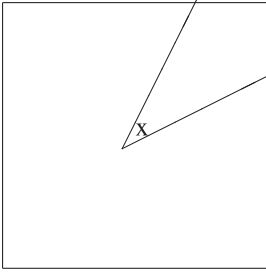


Fig. 1. An angular edge curve exists in a neighborhood of a given pixel located at the center that is marked by “x.”

where  $k > 0$  is the window size of the filter,  $S^{(t+1)}(x, y)$  denotes the denoised image intensity at the pixel  $(x, y)$  in the  $(t + 1)$ th iteration,

$$w^{(t)}(x + i, y + j) = \exp\left(-\frac{i^2 + j^2}{2\sigma_d^2}\right) \exp\left(-\frac{(S^{(t)}(x + i, y + j) - S^{(t)}(x, y))^2}{2\sigma_r^2}\right),$$

$\sigma_d^2$  and  $\sigma_r^2$  are two parameters and  $N^{(t+1)}(x, y)$  is the summation of all weights  $\{w^{(t)}(x + i, y + j)\}$ . From the above expressions, it can be seen that, if a pixel  $(x + i, y + j)$  and the given pixel  $(x, y)$  are on two different sides of an edge curve in the neighborhood, then the weight  $w^{(t)}(x + i, y + j)$  will be small because  $S^{(t)}(x + i, y + j) - S^{(t)}(x, y)$  would be relatively large in such cases. However,  $w^{(t)}(x + i, y + j)$  is still a positive value, although it is relatively small. In other words, observed image intensities in both sides of the edge curve are actually used in defining  $S^{(t+1)}(x, y)$ . Now, let us demonstrate this using a toy example shown in Fig. 1, in which there is an angular edge curve in a neighborhood of a given pixel located at the center (marked by “x”). In such cases, most pixels in the neighborhood are on a different side of the edge curve compared to the given pixel. Therefore, the angle of the edge curve would be blurred by the bilateral filter.

In our opinion, edge features corresponding to relatively large curvature of the edge surfaces are an important component of the image under study because they often represent major characteristics of the image objects and are easier to capture our visual attention compared to places on the edge surfaces with relatively small curvature. Therefore, they should be well preserved during image denoising. In this paper, we propose a new 3D image denoising procedure which can preserve edges and major edge features well. Compared to most existing 3D image denoising procedures, we make the following two innovations in our method: First, we propose detecting edge voxels and then approximating the underlying edge surfaces locally based on the detected edge voxels by a surface template chosen from a prespecified surface template family. The surface template family is specified in a way that major features of the edge surfaces can be accommodated well by its surface templates, which makes it possible for us to preserve edges and major edge features in 3D image denoising. As a comparison, some existing 2D/3D methods (e.g., the TV method [19] and the anisotropic diffusion method [10]) control the amount of data smoothing around a given pixel/voxel based on the estimated gradient of the underlying image intensity

function, and the estimated gradient cannot distinguish different edge structures well. Some other methods (e.g., the kernel smoothing method proposed in [15]) try to preserve edges by locally approximating the edge curves/surfaces by straight lines/planes. By these methods, pointed corners and other structures with large curvature cannot be preserved well. Second, we use certain jump regression analysis (JRA) methodologies [27] in our proposed image denoising procedure. Under the JRA framework, the image denoising problem can be regarded as a special nonparametric regression problem discussed in the statistical literature, where the image intensity function is regarded as a jump regression function with jumps at the edge locations. Then, image denoising can be accomplished by estimating the regression function with jumps preserved. For a detailed discussion about the connection and difference between image processing and JRA, see [14]. One benefit to adopting the JRA framework is that some local smoothing methodologies developed in the statistical literature can be used for image denoising. It has been demonstrated in the literature that such local smoothing methods have flexibility in choosing proper windows and in assigning proper weights for efficient local weighted averaging, which is helpful for different tasks in image processing [15], [28].

The rest part of the paper is organized as follows: Our proposed procedure is described in detail in Section 2. Some numerical examples are presented in Section 3 to evaluate its performance. Several remarks conclude the paper in Section 4. Two appendices are included in a supplementary file, which can be found in the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.261>.

## 2 PROPOSED METHOD

Our proposed 3D image denoising procedure consists of three major steps. First, edge voxels are detected using a 3D edge detector that is constructed under the JRA framework. Second, in a neighborhood of a given voxel, the underlying edge surfaces are approximated by a surface template chosen from a prespecified surface template family. Third, the true image intensity at the given voxel is estimated by a weighted average of the observed image intensities in the neighborhood whose voxels are located on the same side of the surface template as the given voxel. Details of these steps are described in the three parts below.

### 2.1 Identification of Edge Voxels

Under the JRA framework, an observed 3D image can be described by the regression model

$$\xi_{ijk} = f(x_i, y_j, z_k) + \varepsilon_{ijk}, \quad \text{for } i, j, k = 1, 2, \dots, n, \quad (1)$$

where  $(x_i, y_j, z_k) = (i/n, j/n, k/n)$  are the equally spaced voxels in the design space  $\Omega = [0, 1] \times [0, 1] \times [0, 1]$ ,  $\{\varepsilon_{ijk}\}$  are i.i.d. random errors with mean 0 and unknown variance  $\sigma^2$ ,  $f(x, y, z)$  is an unknown image intensity function, and  $N = n^3$  is the total number of voxels. In (1), we assume that all indices  $i, j$ , and  $k$  change from 1 to  $n$ , for simplicity of presentation. Our proposed method in this paper can also work well in cases when their ranges are different.

At a given voxel  $(x, y, z) \in \Omega$ , to know whether it is an edge voxel let us consider its spherical neighborhood  $O^*(x, y, z)$  with radius  $h_n^*$ . In  $O^*(x, y, z)$ , we fit a 3D plane using the local linear kernel (LLK) smoothing procedure [29] as follows:

$$\min_{a,b,c,d} \sum_{i,j,k=1}^n \{ \xi_{ijk} - [a + b(x_i - x) + c(y_j - y) + d(z_k - z)] \}^2 K_{h_n^*}(x_i, y_j, z_k), \quad (2)$$

where  $K_{h_n^*}(x_i, y_j, z_k) = K(\frac{x_i - x}{h_n^*}, \frac{y_j - y}{h_n^*}, \frac{z_k - z}{h_n^*})$ , and  $K$  is a 3D spherically symmetric density kernel function defined in a unit ball. The solution to  $(a, b, c, d)$  of (2) is denoted as  $(\hat{a}(x, y, z), \hat{b}(x, y, z), \hat{c}(x, y, z), \hat{d}(x, y, z))$ , and their expressions are given in Appendix A, available in the online supplemental material. Then,  $\hat{a}(x, y, z)$  is the LLK estimator of  $f(x, y, z)$ , and  $(\hat{b}(x, y, z), \hat{c}(x, y, z), \hat{d}(x, y, z))$  are LLK estimators of  $(f'_x(x, y, z), f'_y(x, y, z), f'_z(x, y, z))$ . In (2), a 3D plane is fitted in the neighborhood  $O^*(x, y, z)$  by a weighted least square procedure for approximating the underlying image intensity function  $f$  around the given voxel  $(x, y, z)$ . The amount of data involved is determined by the radius  $h_n^*$ . Intuitively, at places where the noise level is high,  $h_n^*$  should be chosen relatively large to better reduce noise. At places where the curvature of the image intensity function is small (i.e., the image intensity surface is quite straight),  $h_n^*$  can be chosen relatively large as well. In practice, however, the noise and the true image intensity function are both unobservable. All we have are the observed image intensities  $\{\xi_{ijk}\}$ . A data driven procedure for choosing  $h_n^*$  is discussed in Section 2.3.

The estimated gradient vector  $\hat{\beta}(x, y, z) = (\hat{b}(x, y, z), \hat{c}(x, y, z), \hat{d}(x, y, z))^T$  provides an estimate of the direction that  $f$  increases the fastest around  $(x, y, z)$ . Let us consider a plane that passes  $(x, y, z)$  and is orthogonal to  $\hat{\beta}(x, y, z)$ . Then, this plane divides  $O^*(x, y, z)$  into two halves,  $O_1^*(x, y, z)$  and  $O_2^*(x, y, z)$ . In  $O_1^*(x, y, z)$  and  $O_2^*(x, y, z)$ , we compute weighted averages of the observed image intensities, respectively, with the weights determined by  $K$ . The weighted averages are denoted as  $\hat{a}_1(x, y, z)$  and  $\hat{a}_2(x, y, z)$ . Then,  $(x, y, z)$  is flagged as an edge voxel if

$$|\hat{a}_1(x, y, z) - \hat{a}_2(x, y, z)| > T_n, \quad (3)$$

where  $T_n$  is a threshold. In the case when there are no edge voxels in  $O^*(x, y, z)$ , it is derived, in Appendix B, available in the online supplemental material, that  $\hat{a}_1(x, y, z) - \hat{a}_2(x, y, z)$  is distributed approximately as  $N(0, 4\hat{\sigma}^2 \sum K_{h_n^*}^2(x_i, y_j, z_k) / [\sum K_{h_n^*}(x_i, y_j, z_k)]^2)$ , where  $\sum$  is over all voxels and  $\hat{\sigma}^2 = \frac{1}{N} \sum_{i,j,k=1}^n (\xi_{ijk} - \hat{a}(x_i, y_j, z_k))^2$ . Therefore, a reasonable choice of  $T_n$  is

$$T_n = \frac{2Z_{1-\alpha_n/2} \sqrt{\sum K_{h_n^*}^2(x_i, y_j, z_k)}}{\sum K_{h_n^*}(x_i, y_j, z_k)}, \quad (4)$$

where  $Z_{1-\alpha_n/2}$  is the  $(1 - \alpha_n/2)$ th quantile of the standard normal distribution and  $\alpha_n$  is a prespecified significance level. The value of  $\alpha_n$  provides a measure of the chance that a nonedge voxel would be detected as an edge voxel. Its

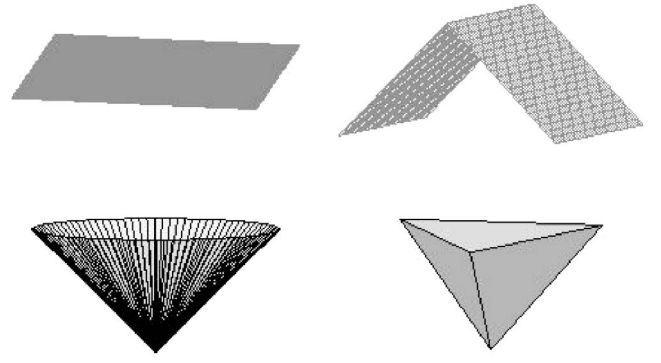


Fig. 2. A family of four surface templates for approximating edge surfaces.

value should be chosen relatively small if the total number of voxels is large. Otherwise, there will be many false edge voxels detected. We have performed many numerical studies in which the number of voxels of each test image is  $r \times 10^6$  with  $r \in [0.5, 10]$  (e.g., images with  $128 \times 128 \times 64$  or  $256 \times 256 \times 128$  voxels), and find that results are reasonably good when we choose  $\alpha_n \in [0.0001, 0.001]$ . In such cases, each nonedge voxel has less than 0.1 percent chance of being detected as an edge voxel. In all numerical examples presented in Section 3, we use a single value  $\alpha_n = 0.0002$  for simplicity, although the value of  $\alpha_n$  can always be adjusted in different cases.

## 2.2 Template Surfaces and Edge Surface Approximation

After detection of edge voxels, the next step of our image denoising procedure is to locally approximate the edge surfaces around a given voxel based on the detected edge voxels, using a template surface from a prespecified family. The template surfaces in the family should describe major features of the edge surfaces well. In 2D cases, people usually focus on two major features of the edge curves: planar parts and angular parts [15], [30], [31]. In 3D cases, the structure of the edge surfaces is much more complicated than that of the edge curves of 2D images. The major complexity is that there are many different types of angular parts in 3D cases, among which two major ones are intersections of two planar edge surfaces (i.e., roofs/valleys of the edge surfaces) and pointed corners. Obviously, these two types of angular parts should be treated differently. So, in 3D cases, we need to describe three major features of the edge surfaces: 1) planar parts, 2) intersections of two planar edge surfaces, and 3) pointed corners. To describe these three major features, we consider the family of three basic surface templates shown in the first row and the left panel of the second row of Fig. 2. Among them, the plane is used for approximating planar parts of the edge surfaces, the intersection of two half planes is for approximating roofs/valleys of the edge surfaces, and the cone is for approximating pointed corners. Of course, there are many variants of each type of edge features in real images. For instance, for pointed corners, instead of the cone shape included in the family, we can also consider the shape of a triangular pyramid that is displayed in the lower right panel of Fig. 2. However, because the edge surface approximation is performed locally in a small neighborhood of a given voxel, consideration of this and other variants can

only marginally improve the edge surface approximation, which is confirmed by numerical examples in Section 3. Further, consideration of extra surface templates would add a substantial amount of computation to the entire image denoising process. For these reasons, we recommend using the first three surface templates shown in Fig. 2, and adding some extra surface templates to the family only in cases when we know that they are common edge features in a specific application.

Next, we describe our proposed algorithms for locally approximating the edge surfaces by the surface templates shown in Fig. 2. At a given voxel  $(x, y, z)$ , let us consider its spherical neighborhood  $O(x, y, z)$  with radius  $h_n$ , where  $h_n$  could be different from  $h_n^*$  used in (2). Again, a data driven procedure for choosing  $h_n$  will be discussed in Section 2.3. Let  $\{s_l, l = 1, 2, \dots, m\}$  be the detected edge voxels in  $O(x, y, z)$ ,  $\{\beta_l^*, l = 1, 2, \dots, m\}$  be the corresponding estimated gradient directions (with unit lengths) at these edge voxels by (2), and

$$G = (w_1\hat{\beta}_1^*, w_2\hat{\beta}_2^*, \dots, w_m\hat{\beta}_m^*)(w_1\hat{\beta}_1^*, w_2\hat{\beta}_2^*, \dots, w_m\hat{\beta}_m^*)^T,$$

where  $w_l = |\hat{a}_1(s_l) - \hat{a}_2(s_l)| - T_n$ , which are all positive at detected edge voxels  $\{s_l, l = 1, 2, \dots, m\}$  (cf., (3)). Therefore,  $G$  is a weighted second moment from origin of  $\{\beta_l^*\}$ , and the weights are determined by the significance of individual detected edge voxels. The eigenvalues of  $G$  are denoted as  $\lambda_1 \leq \lambda_2 \leq \lambda_3$ , and the corresponding eigenvectors with unit lengths are  $\mathbf{v}_1, \mathbf{v}_2$ , and  $\mathbf{v}_3$ . Then, if all  $\beta_l^*$ s are the same (i.e., the underlying edge surface is a plane),  $G$  would have a rank of 1 and  $\mathbf{v}_3$  would be the normal direction of the edge plane, and vice versa. Therefore, to approximate the edge surfaces in  $O(x, y, z)$  by the first surface template shown in Fig. 2, a reasonable solution is the plane that passes the weighted center  $\bar{s}$  of  $\{s_l\}$  with the weights  $\{w_l\}$  and has the normal direction of  $\mathbf{v}_3$ .

To approximate the edge surfaces in  $O(x, y, z)$  by the second surface template (i.e., the one shown in the upper right panel of Fig. 2), we proceed in two steps. First,  $\{s_l\}$  are divided into two groups by a plane that passes  $\bar{s}$  along the directions of  $\mathbf{v}_1$  and  $\bar{\beta}^*$ , where  $\bar{\beta}^*$  denotes the weighted average of  $\{\beta_l^*, l = 1, 2, \dots, m\}$  with the weights  $\{w_l, l = 1, 2, \dots, m\}$ . Second, with each group of the detected edge voxels, find an approximation plane passing their weighted averaged locations with its normal direction to be the weighted average of the estimated gradient directions at the detected edge voxels. Then, the two resulting half planes that cross each other and form a subspace in  $O(x, y, z)$  containing  $\bar{s}$  are used for approximating the edge surfaces in  $O(x, y, z)$ .

To approximate the edge surfaces in  $O(x, y, z)$  by a cone (i.e., the surface template shown in the lower left panel of Fig. 2), we need to specify its central axis, vertex position, and the angle between the central axis and any generatrix. Assume that the direction of the central axis is  $\mathbf{d} = (1, d_2, d_3)^T$ . Since the angle between this direction and the normal direction at any point on the cone is a constant,  $d_2$  and  $d_3$  can be estimated by minimizing the weighted sample variance of the inner products between  $\mathbf{d}$  and  $\{\beta_l^*, l = 1, 2, \dots, m\}$ . Simple calculations show that

$$d_2 = (\Sigma_{23}\Sigma_{13} - \Sigma_{33}\Sigma_{12})/(\Sigma_{22}\Sigma_{33} - \Sigma_{23}^2)$$

$$d_3 = (\Sigma_{12}\Sigma_{23} - \Sigma_{22}\Sigma_{13})/(\Sigma_{22}\Sigma_{33} - \Sigma_{23}^2),$$

where  $\Sigma_{j_1j_2}$  is the  $(j_1, j_2)$ th component of the weighted sample covariance matrix of  $\{\beta_l^*, l = 1, 2, \dots, m\}$ , for  $j_1, j_2 = 1, 2, 3$ . To specify the location of the central axis, let us consider a sphere  $\tilde{O}(x, y, z)$  of radius  $\tilde{h}_n > h_n$ . The plane  $\tilde{P}$  passing  $\bar{s}$  with the normal direction of  $\mathbf{d}$  would divide  $\tilde{O}(x, y, z)$  into two parts. Weighted centers of the detected edge voxels in the two parts are then calculated, and the one closer to  $\tilde{P}$  is denoted as  $\bar{s}^* = (c_x^*, c_y^*, c_z^*)$ . Then, the line passing  $\bar{s}^*$  along direction  $\mathbf{d}$  is defined to be the central axis of the cone. In this paper, we choose  $\tilde{h}_n = 3h_n$ . As a matter of fact, selection of  $\tilde{h}_n$  does not have much effect on the final results. After the central axis is determined, the angle between the central axis and any generatrix can be easily estimated by the weighted average of the angles between  $\mathbf{d}$  and  $\{\beta_l^*, l = 1, 2, \dots, m\}$ . The location of the vertex  $(v_x, v_y, v_z)$  of the cone can be estimated by minimizing the weighted orthogonal distance between the cone and the detected edge voxels in  $O(x, y, z)$ . After some algebraic manipulations, the estimator can be calculated by  $(c_x^* + \bar{\beta}_1^*\rho, c_y^* + \bar{\beta}_2^*\rho, c_z^* + \bar{\beta}_3^*\rho)$ , where  $\bar{\beta}^* = (\bar{\beta}_1^*, \bar{\beta}_2^*, \bar{\beta}_3^*)^T$ ,  $\rho$  is one of

$$\left[ \sum w_l d_l t_l / (\tan \hat{\theta} \|\bar{\beta}^*\|) + \sum w_l t_l^2 \right] / \sum w_l t_l$$

and

$$\left[ \sum w_l d_l t_l / (\tan \hat{\theta} \|\bar{\beta}^*\|) - \sum w_l t_l^2 \right] / \left( - \sum w_l t_l \right),$$

that minimizes  $\sum w_l (d_l \cos \hat{\theta} - |t - t_l| \|\bar{\beta}^*\| \sin \hat{\theta})^2$ ,  $t_l = (\bar{\beta}^*)^T (s_l - \bar{s}) / \|\bar{\beta}^*\|^2$ , and  $d_l = \|s_l - \bar{s}^* - t_l \bar{\beta}^*\|$ .

To approximate the edge surfaces in  $O(x, y, z)$  by a triangular pyramid (i.e., the surface template shown in the lower right panel of Fig. 2), we need to estimate its vertical axis (i.e., the line passing through the vertex and the center of its triangular base), the angular orientation of the triangular base, the normal directions of the three lateral planes, and the position of the vertex. To estimate the vertical axis, let us consider the spherical neighborhood  $\tilde{O}(x, y, z)$  that is described in the previous paragraph. The weighted center of the detected edge voxels within  $\tilde{O}(x, y, z)$  is denoted as  $\tilde{s} = (\tilde{c}_x, \tilde{c}_y, \tilde{c}_z)$ . Then, the vertical axis is estimated by the line passing  $\tilde{s}$  in the direction of the weighted mean of the estimated gradient directions at all detected edge voxels in  $\tilde{O}(x, y, z)$ . To determine the angular orientation of the triangular base, we consider a plane that passes  $\tilde{s}$  and is orthogonal to the vertical axis. We also fix a half straight line  $\tilde{L}$  on the plane that starts from  $\tilde{s}$ . Another two half straight lines starting from  $\tilde{s}$  on the plane can then be determined such that the three half lines are 120 degrees apart and partition the plane into three equal parts. The detected edge voxels in  $O(x, y, z)$  can then be divided into three groups based on the parts of the plane that their projections to the plane belong to. Intuitively, if the three groups of the detected edge voxels are located on the three lateral planes of the triangular pyramid, respectively, then the variability of the estimated gradient directions corresponding to each group of detected edge voxels should be small. Based on this intuition, the sample variances of the three groups of the estimated gradient directions are

TABLE 1

In Each Entry, the First Line Presents the Estimated MISE Value and Their Standard Errors (in Parentheses), the Second Line Presents the Searched Procedure Parameter Values

Method	$\sigma = 80$	$\sigma = 100$	$\sigma = 120$	variable $\sigma$
TV	1119.3 (2.9) .0030	1401.1 (3.9) .0025	1649.1 (4.9) .0019	1073.1 (2.8) .0032
AD	1157.8 (3.5) 475,1	1525.0 (4.8) 700,1	1916.9 (6.3) 900,1	1088.9 (3.3) 225,2
ONLM	884.8 (3.4) 12,1	1127.5 (4.6) 16,1	1364.0 (6.2) 25,1	845.2 (3.2) 11,1
NEW	837.3 (2.4) .0172,.0156	1022.7 (3.1) .0188,.0188	1189.2 (4.1) .0188,.0188	811.5 (2.2) .0172,.0156
NEW-CV	850.4 (2.4) .0117,.0156	1029.0 (2.9) .0172,.0188	1189.2 (4.1) .0188,.0188	823.8 (2.4) .0117,.0156
NEW-P	835.3 (2.5) .0172,.0156	1011.8 (3.5) .0180,.0180	1184.1 (4.7) .0180,.0180	809.7 (2.2) .0172,.0156

computed, respectively, and their weighted average is also computed, with the weights proportional to the numbers of detected edge voxels in the three groups. This weighted average is called the within-group variance hereafter. We then keep rotating  $\tilde{L}$  clockwise (or counterclockwise) within the plane with a step of  $\nu$  degrees until the cumulative rotated degrees reaching 120 degrees. For each position of  $\tilde{L}$ , the within-group variance is calculated. Then, the final position of  $\tilde{L}$  is chosen to be the one among all its positions considered with the smallest within-group variance. After the final position of  $\tilde{L}$  is determined, the normal directions of the three lateral sides of the pyramid can be estimated by the weighted means of the estimated gradient directions within the three groups of detected edge voxels, respectively. Finally, the vertex of the pyramid can be searched on the vertical axis to minimize the weighted sum of the squared orthogonal distances from the detected edge voxels in  $O(x, y, z)$  to the respective lateral sides of the pyramid. To this end, we search for the pyramid vertex on both sides of the vertical axis within  $O(x, y, z)$ , starting from  $\tilde{s}$  with a step of  $\tau$ . In the algorithm just described, there are two parameters  $\nu$  and  $\tau$ . In our numerical study presented in Section 3, we choose  $\nu = 5$  and  $\tau = 0.05/n$  because results with smaller values of these parameters would hardly change in all cases considered.

In practice, we need to choose one of the four estimated surface templates based on observed image intensities for approximating the edge surfaces in  $O(x, y, z)$ . For that purpose, one natural solution is to choose the estimated surface template with the smallest residual sum of squares

(RSS), where RSS is defined to be the sum of squared orthogonal distances from  $\{s_l, l = 1, 2, \dots, m\}$  to the estimated surface template. However, by this idea, the first surface template would never be selected because it is a special case of the second surface template and, consequently, its RSS value will never be the smallest one. To overcome this difficulty, we suggest the following two-step algorithm. Let  $RSS_1$ ,  $RSS_2$ ,  $RSS_3$ , and  $RSS_4$  be the RSS values of the four estimated surface templates, respectively. Then,

1. the third template is selected if  $RSS_3$  is the smallest one among  $RSS_1$ ,  $RSS_2$ ,  $RSS_3$ , and  $RSS_4$ , and the fourth template is selected if  $RSS_4$  is the smallest RSS value;
2. otherwise, the first template is selected if

$$F(x, y, z) = \frac{(RSS_1(x, y, z) - RSS_2(x, y, z))/3}{RSS_2(x, y, z)/(m - 6)} \leq \chi_{3,1-\tilde{\alpha}}^2,$$

and the second template is selected if  $F(x, y, z) > \chi_{3,1-\tilde{\alpha}}^2$ , where  $\chi_{3,1-\tilde{\alpha}}^2$  is the  $(1 - \tilde{\alpha})$ th quantile of the  $\chi_3^2$  distribution.

In step 2, we have used the statistical result that  $F(x, y, z)$  is distributed asymptotically as  $\chi_3^2$  when  $m$  increases [32], which is true here because the first template is a special case of the second template and the second template has three more parameters than the first template. For the value of  $\tilde{\alpha}$ , if it is chosen smaller, then the second template would have a smaller chance to be selected. Based on our numerical experience, results are reasonably good if we choose  $\tilde{\alpha} = 0.01$ . For simplicity, this value of  $\tilde{\alpha}$  is used in all numerical examples presented in this paper, although its value can always be adjusted in different cases. Note that, if we decide to use the first three surface templates only in our image denoising procedure, then step 1 should be changed to

1. the third template is selected if  $RSS_3$  is the smallest one among  $RSS_1$ ,  $RSS_2$ , and  $RSS_3$ ,

and step 2 is not changed.

### 2.3 3D Image Denoising

After the edge surfaces in the neighborhood  $O(x, y, z)$  are approximated by a surface template, the neighborhood is divided by the surface template into subneighborhoods. The one containing the voxel  $(x, y, z)$  is denoted as  $U(x, y, z)$ . Then,  $f(x, y, z)$  can be estimated by the solution to a of the minimization problem (2), after  $K_{h_n^*}(x_i, y_j, z_k)$  is replaced by  $I((x_i, y_j, z_k) \in U(x, y, z))K_{h_n}(x_i, y_j, z_k)$ , where  $I(\cdot)$  is the

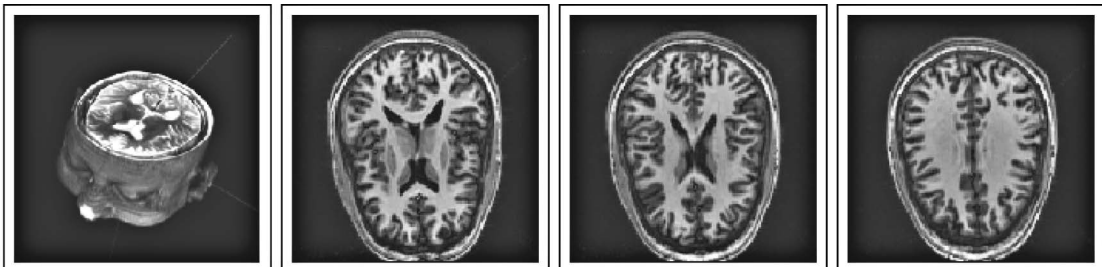


Fig. 3. A demonstration of a 3D image and its three 2D slices.



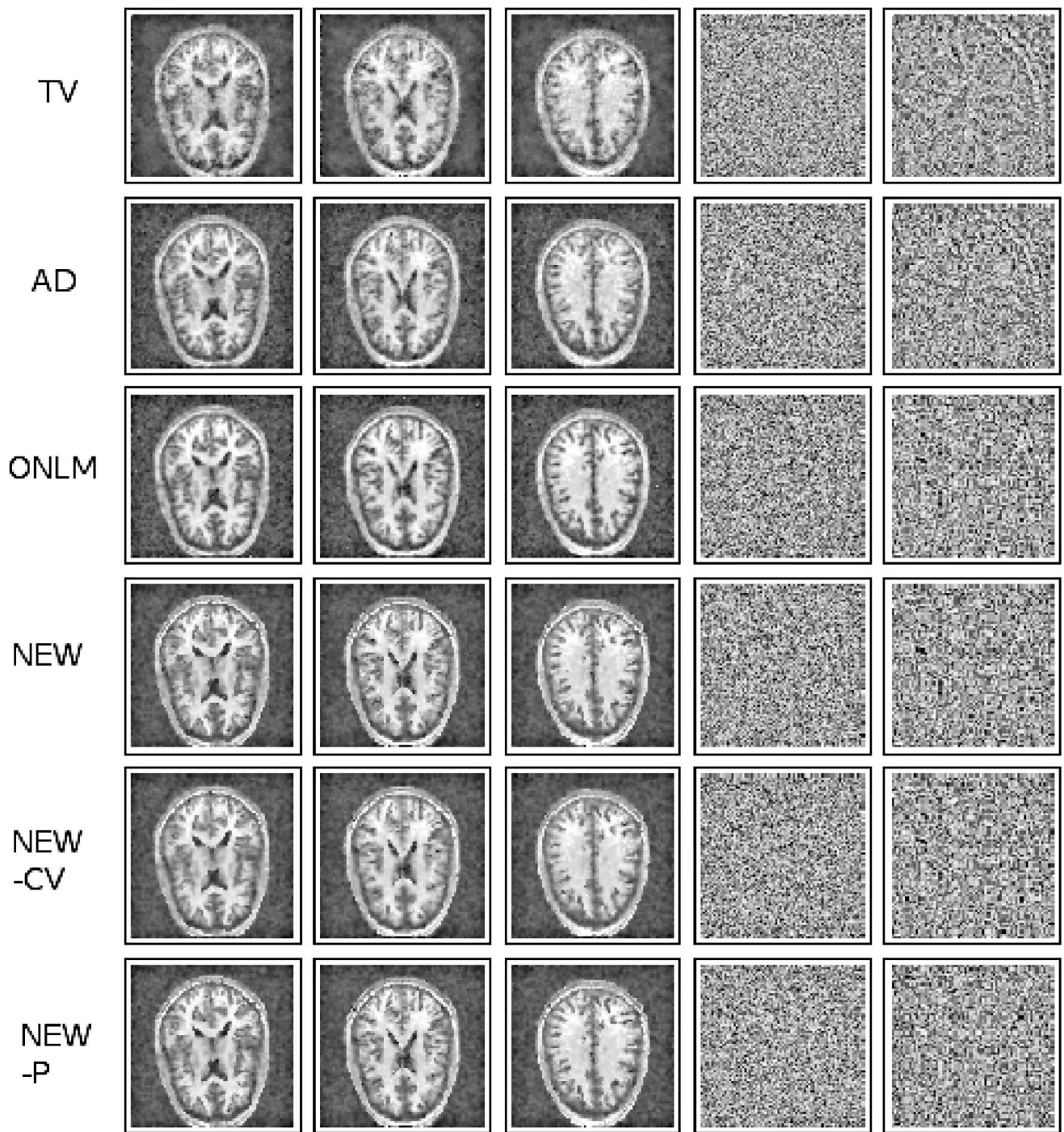


Fig. 4. The first three columns show the three slices of the denoised images by procedures TV, AD, ONLM, NEW, NEW-CV, and NEW-P, respectively, in six rows when  $\sigma = 100$ . Images in column 4 are residual images of those shown in column 1. Images in column 5 show the closeup views of an upper-right portion of the images in column 4.

indicator function, and it equals 1 if its argument is “true” and 0 otherwise. From (2) and the expressions given in Appendix A, available in the online supplemental material, for its solutions, we can see that  $\hat{f}(x, y, z)$  is a weighted average of the observed image intensities whose voxels are located on the same side of the estimated surface template in  $O(x, y, z)$  as the given voxel  $(x, y, z)$ . Intuitively, as long as the estimated surface template approximates the underlying edge surfaces well,  $\hat{f}(x, y, z)$  should preserve edges and major edge features well.

In the proposed denoising procedure, there are two parameters  $h_n^*$  and  $h_n$  involved. We suggest choosing them by the following cross-validation (CV) procedure:

$$CV(h_n^*, h_n) = \frac{1}{n^3} \sum_{i,j,k=1}^n (\xi_{ijk} - \hat{f}_{-i,-j,-k}(x_i, y_j, z_k))^2, \quad (5)$$

where  $\hat{f}_{-i,-j,-k}(x_i, y_j, z_k)$  is the estimate of  $f(x_i, y_j, z_k)$  when the  $(i, j, k)$ th voxel  $(x_i, y_j, z_k)$  is excluded from all subsequent steps of the proposed image denoising procedure after edge detection. Then,  $h_n^*$  and  $h_n$  are chosen by minimizing

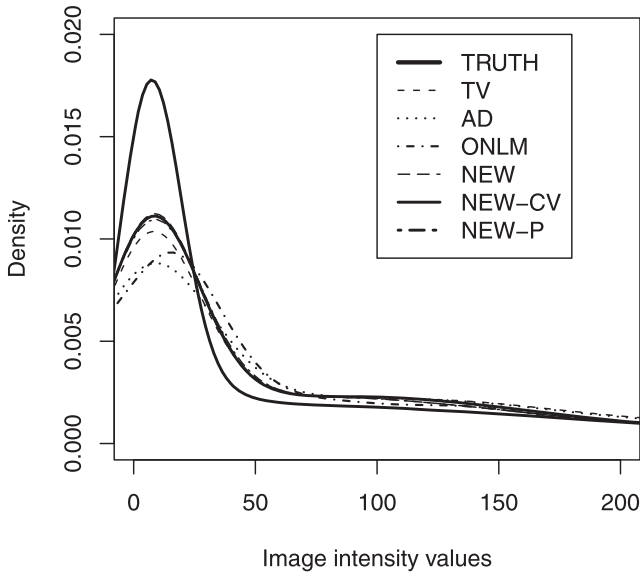


Fig. 5. Density curves of the image intensities of the true 3D image and the six denoised 3D images when  $\sigma = 100$ .

$CV(h_n^*, h_n)$ . Because each  $\hat{f}_{-i,-j,-k}(x_i, y_j, z_k)$  is computed in a small neighborhood  $O(x, y, z)$ , computation involved in the CV procedure (5) is manageable. Of course, the amount of computation depends on the image size, the number of image intensity levels, the values of  $h_n^*$  and  $h_n$  considered, and so forth. In the example of Table 1 discussed in Section 3, when  $h_n^*$  and  $h_n$  take those values listed in the table it takes about a minute to compute the quantity  $CV(h_n^*, h_n)$  in (5) for each set of the values of  $(h_n^*, h_n)$  in each case considered there on our Pentium-M 1.6 MHz computer running Linux operating system.

It should be pointed out that the denoising procedure discussed above is for cases when the pointwise noise contained in the observed image has a mean of 0 (cf., model (1)). In some applications, such as MRI imaging, the noise level at a given voxel may depend on the true image intensity level, and the noise distribution can often be described by a Rician, Poisson, noncentral Chi-square, or other models [33]. In such cases, denoised images by procedures designed for removing additive zero-mean noise would be biased for estimating the true images, and many bias-correction methods have been proposed in the literature (e.g., [34], [25], [35]). In the next section, we will present a numerical example with Rician noise involved.

For a real image, there are regions where  $f$  is smooth. In these regions, the number of detected edge voxels should be small. So, before estimating  $f$ , we suggest counting the number of detected edge voxels in  $O(x, y, z)$ . If the number is so small (e.g., smaller than  $(nh_n)^2$ ) that a potential edge surface in  $O(x, y, z)$  is unlikely, then  $f(x, y, z)$  can be estimated simply by the conventional LLK estimator constructed from all observations in  $O(x, y, z)$ . The denoising procedure described in the first paragraph of this section is used only when the number of detected edge voxels in  $O(x, y, z)$  is relatively large. There are at least two benefits to doing so. One is that much computation is saved because the conventional LLK estimator is much easier to compute compared to the proposed denoising procedure based on local edge surface approximation. The other

benefit is that the estimated  $f$  would be more reliable in cases when the number of detected edge voxels is small because it is constructed from all observations in  $O(x, y, z)$  instead of from part of the observations. Considering the fact that most voxels in an image are not edge voxels, these benefits should be substantial.

### 3 NUMERICAL STUDY

In this section, we present some numerical examples to evaluate the performance of the proposed 3D image denoising procedure in comparison with three existing procedures, including the ones based on total variation [18] (denoted as TV), anisotropic diffusion [20] (denoted as AD), and optimized nonlocal means [25] (denoted as ONLM). In Section 2.2, it was pointed out that the first three surface templates shown in Fig. 2 could describe major edge features in 3D images well. In order to investigate the impact of the inclusion of the fourth surface template (i.e., the triangular pyramid shown in the lower right panel of Fig. 2) on the performance of our proposed method, we consider its two versions. One uses the first three surface templates only (denoted as NEW), and the other uses all four surface templates (denoted as NEW-P). In both versions, the kernel function  $K$  is chosen to be the Epanechnikov kernel function  $K(x, y, z) = C(1 - x^2)(1 - y^2)(1 - z^2)$  defined in the unit ball  $\{(x, y, z) : x^2 + y^2 + z^2 \leq 1\}$ , where  $C > 0$  is a normalization constant. It has been shown in the statistical literature [36] that this kernel function has certain optimality properties. The procedure TV has a regularization parameter involved, the procedure AD is an iterative algorithm and contains two parameters, i.e., the diffusion parameter and the number of iterations, and the procedure ONLM has two bandwidth parameters to choose. To evaluate the performance of a denoised image  $\hat{f}$ , a standard statistical criterion is the mean integrated squared error (MISE), defined as  $MISE(\hat{f}, f) = \int_0^1 \int_0^1 \int_0^1 [\hat{f}(x, y, z) - f(x, y, z)]^2 dx dy dz$ , which is estimated by the sample mean of

$$\frac{1}{n^3} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n [\hat{f}(x_i, y_j, z_k) - f(x_i, y_j, z_k)]^2,$$

based on 100 replications.

We first consider a 3D MRI image of a human brain which has  $128 \times 128 \times 52$  voxels. Its image intensity levels range from 0 to 809. A demonstration of the 3D image and its three slices are shown in Fig. 3. Then, i.i.d. random noise from the distribution  $N(0, \sigma^2)$  is added to the image, and  $\sigma$  is chosen to be 80, 100, or 120, representing low, medium, and high noise levels. With each  $\sigma$  value, the parameters of all denoising procedures are chosen such that their estimated MISE values based on 100 replications reach the minimum. For the proposed denoising procedure NEW, we also consider choosing its parameters by the CV procedure (5) to investigate the effectiveness of the CV procedure. The corresponding method is denoted as NEW-CV.

The estimated MISE values, their standard errors, and the procedure parameter values of the related methods are presented in the first three columns of Table 1. From the table, we can see that the proposed procedure NEW outperforms all three competing methods TV, AD, and ONLM in all cases, and the outperformance is statistically

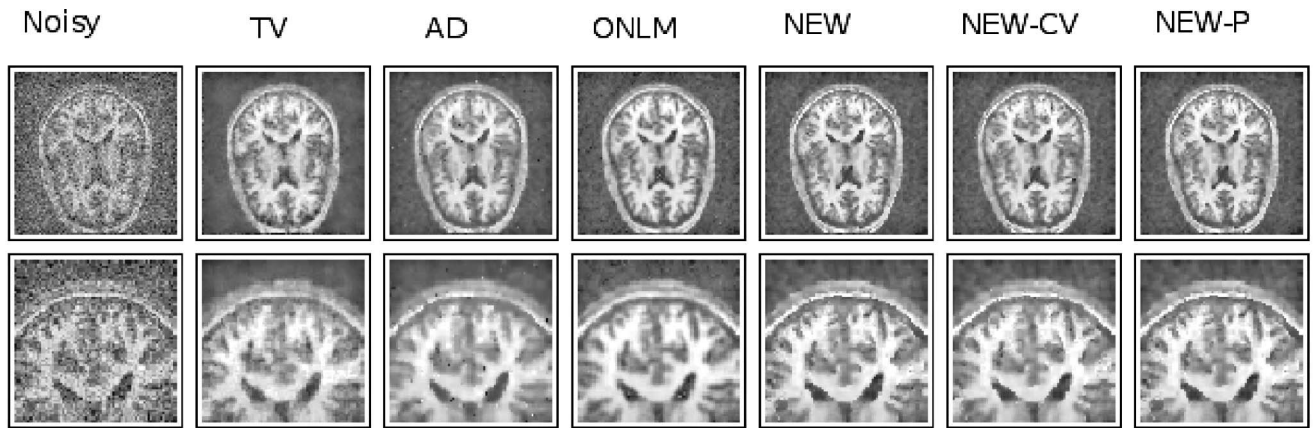


Fig. 6. Images in the first row show the first slice (shown in Fig. 3) of the noisy image and the denoised images by procedures TV, AD, ONLM, NEW, NEW-CV, and NEW-P, respectively, when the noise level is variable, as defined in (6). Images in the second row show the closeup views of a top portion of the corresponding images in the first row.

significant because the difference of the MISE values of the procedure NEW and any one of these methods is larger than 2 times of the sum of the two corresponding standard errors. Also, when its parameters are chosen by the CV procedure (cf., (5)), its performance gets slightly worse, compared to the procedure NEW with the parameters chosen by minimizing MISE. But, NEW-CV still outperforms all three competing methods in all cases by a reasonably large margin. By comparing NEW with NEW-P, it can be seen that the performance of NEW-P is indeed only marginally better than that of NEW. The first three columns in Fig. 4 demonstrate the three slices of the denoised images by procedures TV, AD, ONLM, NEW, NEW-CV, and NEW-P, when  $\sigma = 100$ . The corresponding slices of the original image are those shown in Fig. 3. From the images, it can be seen that edge structures are better preserved by procedures NEW, NEW-CV, and NEW-P (cf., several central dark regions in images shown in the first column). The residual plots of the images in the first column are shown in the fourth column, from which we can see that the residual plots of procedures TV, AD, and ONLM show slight patterns around the brain boundary and such patterns are almost invisible in the residual plots of procedures NEW, NEW-CV, and NEW-P, which can be confirmed by the close-up views of an upper right portion of the residual images that are shown in column 5 of Fig. 4. To further investigate the denoised images, the density curves of the image intensities of the denoised images when  $\sigma = 100$  are shown in Fig. 5. Because the density curves of the procedures NEW, NEW-CV, and NEW-P are almost identical, only the one of NEW-CV is visible in the plot. From the plot, it can be seen that, when image intensities are in the range from 0 to about 70, the density curves of NEW, NEW-CV, and NEW-P are closer to the density curve of the true image, shown by the thick solid curve in the plot, than the density curves of the three competing procedures. In cases when image intensities are larger than 70, the density curves of all procedures are close to each other. From the plot, it can also be seen that more than 80 percent image intensities of the true image or each of the denoised images are below 70. So, we can conclude that overall the image intensities of the denoised images by procedures NEW, NEW-CV, and NEW-P are closer to the image intensities of

the true image, compared to the image intensities of the denoised images by the three competing procedures.

In practice, the noise level may not be homogeneous in the entire image. To simulate this scenario, we consider the following variable  $\sigma$ :

$$\sigma(x, y, z) = 30 + 50 \exp \left[ -\frac{(x - .5)^2 + (y - .5)^2 + (z - .5)^2}{4} \right]. \quad (6)$$

The MISE values, their standard errors, and the searched procedure parameter values of the six methods in this case are shown in the last column of Table 1. Denoised images are shown in Fig. 6 for the first slice presented in Fig. 3. From Table 1 and Fig. 6, it can be seen that procedures NEW, NEW-CV, and NEW-P also perform better than the three competing procedures in this case.

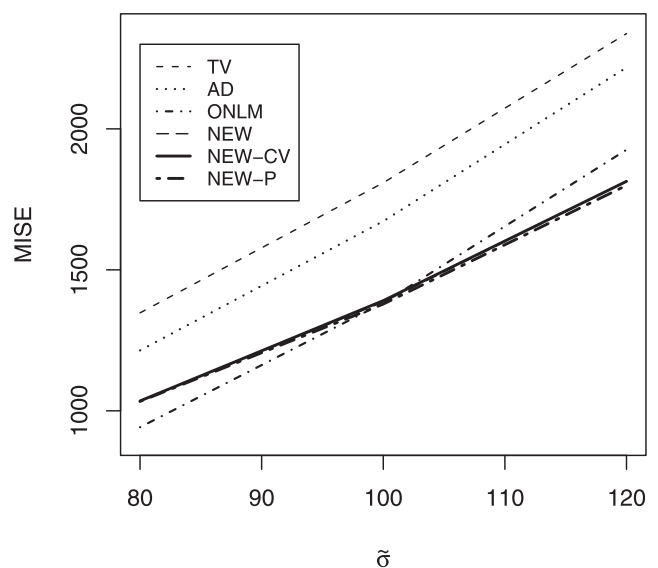


Fig. 7. MISE values of various methods based on 100 replications when  $\tilde{\sigma} = 80, 100$ , and 120.



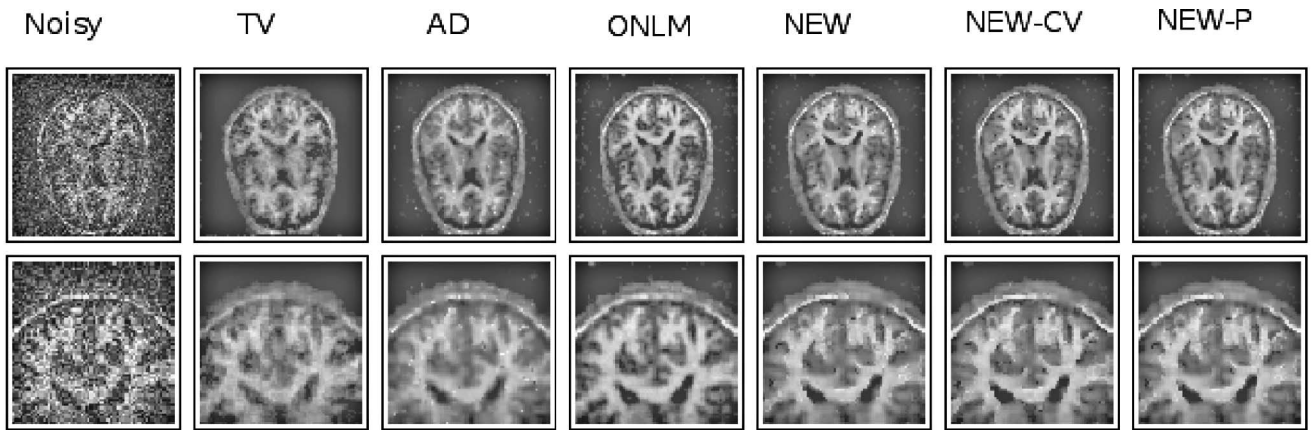


Fig. 8. Images in the first row show a slice of the noisy image and its denoised versions by procedures TV, AD, ONLM, NEW, NEW-CV, and NEW-P, respectively, when Rician noise with  $\tilde{\sigma} = 100$  is considered. The slice presented here corresponds to the second slice shown in Fig. 3. Images in the second row show the closeup views of the top portion of the images in the first row.

Next, we consider adding the Rician noise to the 3D test image shown in Fig. 3. By adding the Rician noise, the observed image intensity at the voxel  $(x, y, z)$  is generated by

$$\sqrt{[f(x, y, z) + \epsilon_1]^2 + \epsilon_2^2},$$

where  $\epsilon_1$  and  $\epsilon_2$  are i.i.d. noise from the distribution  $N(0, \tilde{\sigma}^2)$ . As in the previous examples, we consider cases when  $\tilde{\sigma} = 80, 100$ , and  $120$ . For each denoising procedure, we use the bias correction method proposed in [37] to remove estimation bias. Let  $\tilde{f}(x, y, z)$  be the intensity of the denoised image by a given denoising procedure. Then, its bias-corrected version is defined to be

$$\sqrt{\tilde{f}^2(x, y, z) - 2\tilde{\sigma}^2}.$$

In practice,  $\tilde{\sigma}$  is often unknown and should be estimated from the observed image. In this example, it is estimated by  $\sqrt{s/2}$ , where  $s$  is the sample standard deviation of the squared observed intensities at the first  $16 \times 16 \times 13$  voxels. The calculated MISE values based on 100 replications of all six procedures are presented in Fig. 7. Again, the MISE curve of the procedure NEW is not visible in the plot because it is overlapped with the one of NEW-CV. From the plot, it can be seen that procedures NEW, NEW-CV, and NEW-P outperform all three competing procedures in all cases except the case when  $\tilde{\sigma} = 80$ . In that case, the procedure ONLM is the best. In the case when  $\tilde{\sigma} = 100$ , the procedure ONLM performs similarly to the procedures NEW, NEW-CV, and NEW-P, which can also be seen from Fig. 8 where the denoised images of the six procedures are shown for the second slice presented in Fig. 3.



Fig. 9. Three slices of a 3D test image of a person's head.

We next consider another 3D MRI test image of a person's head which has  $128 \times 128 \times 126$  voxels with intensity levels ranging from 0 to 255. Three slices of the image are shown in Fig. 9. Then, i.i.d. noise from the distribution  $N(0, \sigma^2)$  is added to the image. We consider three noise levels:  $\sigma = 50, 75$ , and  $100$ , representing low, medium, and high noise levels. With each noise level, parameters of all the denoising procedures considered are chosen in exactly the same way as that in the example of Table 1. The estimated MISE values based on 100 replications, their standard errors, and the procedure parameter values are shown in the first three columns of Table 2. From the table, we can see that 1) procedures NEW, NEW-CV, and NEW-P are uniformly better than the three competing procedures TV, AD, and ONLM, 2) procedure NEW-CV is slightly worse than procedure NEW, and 3) procedure NEW-P only marginally improves procedure NEW. All these results are consistent with the ones found in the example of Table 1. The first three columns of Fig. 10 demonstrate three slices of the denoised images by procedures TV, AD, ONLM, NEW, NEW-CV, and NEW-P, respectively, when  $\sigma = 75$ . The three slices correspond to the ones shown in Fig. 9. From the images, it seems that the edge

TABLE 2

In Each Entry, the First Line Presents the Estimated MISE Value and Their Standard Errors (in Parentheses), the Second Line Presents the Searched Procedure Parameter Values

Method	$\sigma = 50$	$\sigma = 75$	$\sigma = 100$	salt-and-pepper
TV	557.1 (0.7) .0030	803.5 (1.6) .0019	1020.2 (2.0) .0013	752.1 (2.1) .0033
AD	488.9 (1.0) 135,2	749.6 (1.3) 230,2	964.4 (1.7) 350,2	836.0 (1.7) 800,1
ONLM	504.5 (0.9) 10,1	707.5 (1.7) 14,1	934.2 (2.8) 16,1	629.9 (2.0) 8,1
NEW	462.1 (0.7) .0172,.0141	610.9 (1.0) .0188,.0203	731.3 (1.4) .0219,.0234	594.5 (1.7) .0219,.0156
NEW-CV	482.8 (0.7) .0172,.0188	610.9 (1.0) .0188,.0188	731.7 (1.5) .0219,.0219	617.8 (1.4) .0188,.0188
NEW-P	461.6 (0.7) .0180,.0156	608.9 (1.1) .0188,.0188	729.7 (1.7) .0219,.0203	592.9 (1.5) .0250,.0156

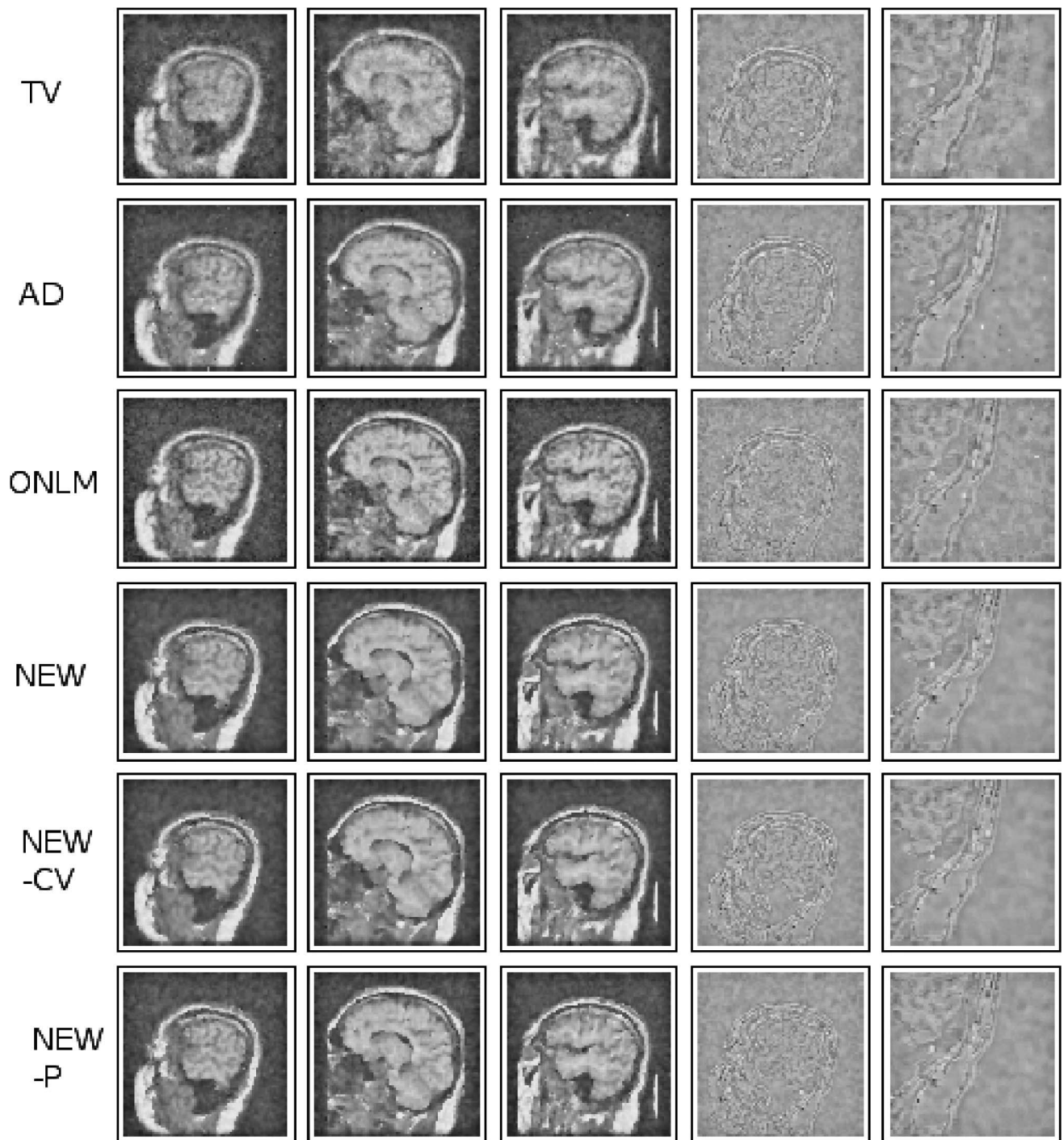


Fig. 10. The first three columns show the three slices of the denoised images of the test image of head by procedures TV, AD, ONLM, NEW, NEW-CV, and NEW-P, respectively, in six rows when  $\sigma = 75$ . Images in column 4 are residual images of those shown in column 1. Images in column 5 show the closeup views of the neck portion of the images in column 4.

structures are better preserved by procedures NEW, NEW-CV, and NEW-P (cf., the boundary of the neck part in images shown in the first column). This can be confirmed by the deviation plots (i.e., plots of true images minus denoised images) of the images in the first column, shown in the fourth column of Fig. 10, and their closeup views (shown in the fifth column of Fig. 10) of the neck portion.

In MRI imaging, salt-and-pepper noise is quite common. Next, we simulate a scenario with salt-and-pepper noise using the test image of head by changing the image

intensities of 5 percent randomly selected voxels to 0 and changing the image intensities of another 5 percent randomly selected voxels to 255. The noisy image is shown in the (1,1)th plot of Fig. 11 for the first slice presented in Fig. 9. Then, all six denoising procedures are applied to the noisy image. Their procedure parameters are chosen in the same way as before. The MISE values, their standard errors, and the searched procedure parameter values of the six procedures in this case are shown in the last column of Table 2. Denoised images are shown in the first row of Fig. 11

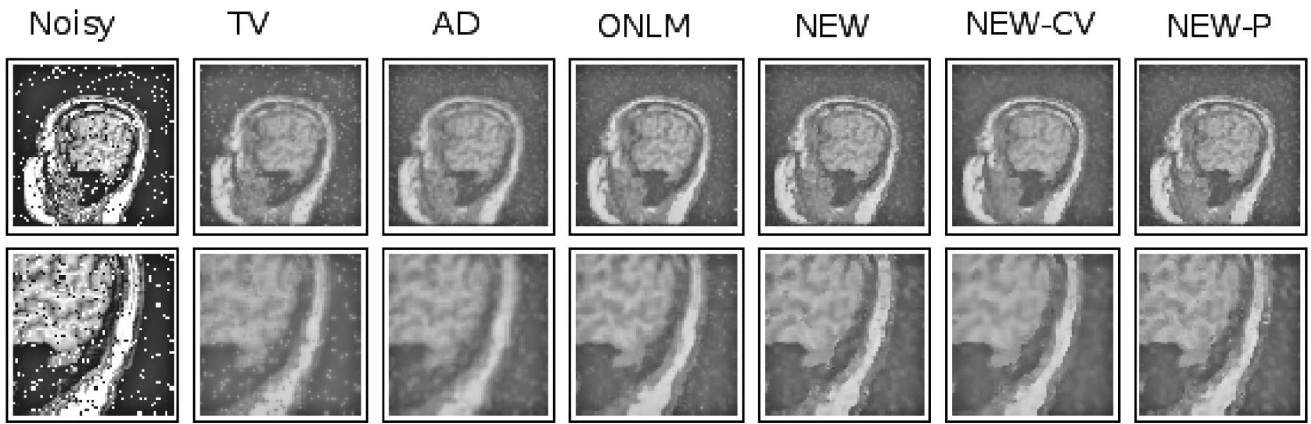


Fig. 11. Images in the first row show the first slice (shown in Fig. 9) of the noisy image and the denoised images by procedures TV, AD, ONLM, NEW, NEW-CV, and NEW-P, respectively, when 10 percent salt-and-pepper noise is added to the test image of head. Images in the second row show the closeup views of the neck portion of the corresponding images in the first row.

for the first slice. Images in the second row of Fig. 11 show the closeup views of the neck portion of the images in the first row. From Table 2 and Fig. 11, it can be seen that procedures NEW, NEW-CV, and NEW-P also perform better than the three competing procedures in this case.

## 4 CONCLUSIONS

We have presented a 3D image denoising procedure which can preserve edges and major edge structures well. Numerical examples show that it performs favorably compared to several existing methods in various cases. However, 3D image denoising is a challenging problem. The current version of the proposed procedure may not be able to preserve edges around places where three or more edge surfaces cross or where jump magnitudes in image intensity are close to zero. To solve these problems, it might help to include more surface templates in the denoising procedure than the ones shown in Fig. 2 when locally approximating the edge surfaces. Such potential improvements are left to our future research.

## ACKNOWLEDGMENTS

The authors thank the editor, the associate editor, and the referees for many constructive comments and suggestions, which greatly improved the quality of the paper. This research is supported in part by a US National Science Foundation (NSF) grant.

## REFERENCES

- [1] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, third ed. Thomson Learning, 2008.
- [2] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, Nov. 1984.
- [3] J. Besag, "On the Statistical Analysis of Dirty Pictures (with Discussion)," *J. Royal Statistical Soc. (Series B)*, vol. 48, pp. 259-302, 1986.
- [4] F. Godtlielsen and G. Sebastiani, "Statistical Methods for Noisy Images with Discontinuities," *J. Applied Statistics*, vol. 21, pp. 459-477, 1994.
- [5] T. Sun, M. Gabbouj, and Y. Neuvo, "Center Weighted Median Filters: Some Properties and Their Applications in Image Processing," *Signal Processing*, vol. 35, pp. 213-229, 1994.
- [6] C.K. Chu, I.K. Glad, F. Godtlielsen, and J.S. Marron, "Edge-Preserving Smoothers for Image Processing (with Discussion)," *J. Am. Statistical Assoc.*, vol. 93, pp. 526-541, 1998.
- [7] M. Hillebrand and C.H. Muller, "Outlier Robust Corner-Preserving Methods for Reconstructing Noisy Images," *The Annals of Statistics*, vol. 35, pp. 132-165, 2007.
- [8] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 839-846, 1998.
- [9] P. Saint-Marc, J. Chen, and G. Medioni, "Adaptive Smoothing: A General Tool for Early Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 514-529, June 1991.
- [10] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, July 1990.
- [11] J. Portilla, V. Strela, M. Wainwright, and E.P. Simoncelli, "Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain," *IEEE Trans. Image Processing*, vol. 12, no. 11, pp. 1338-1351, Nov. 2003.
- [12] P. Qiu, "Discontinuous Regression Surfaces Fitting," *The Annals of Statistics*, vol. 26, pp. 2218-2245, 1998.
- [13] P. Qiu, "The Local Piecewisely Linear Kernel Smoothing Procedure for Fitting Jump Regression Surfaces," *Technometrics*, vol. 46, pp. 87-98, 2004.
- [14] P. Qiu, "Jump Surface Estimation, Edge Detection, and Image Restoration," *J. Am. Statistical Assoc.*, vol. 102, pp. 745-756, 2007.
- [15] I. Gijbel, A. Lambert, and P. Qiu, "Edge-Preserving Image Denoising and Estimation of Discontinuous Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1075-1087, July 2006.
- [16] S. Keeling, "Total Variation Based Convex Filters for Medical Imaging," *Applied Math. and Computation*, vol. 139, pp. 101-119, 2003.
- [17] Y. Wang and H. Zhou, "Total Variation Wavelet-Based Medical Image Denoising," *Int'l J. Biomedical Imaging*, vol. 2006, pp. 1-6, 2006.
- [18] P. Getreuer, "tvdenoise.m (a Matlab Program)," <http://www.mathworks.co.uk/matlabcentral/fileexchange/16236>, 2007.
- [19] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, vol. 60, pp. 259-268, 1992.
- [20] D.S. Lopes, "anisodiff3d.m (a Matlab Program)," <http://www.mathworks.co.uk/matlabcentral/fileexchange/14995>, 2007.
- [21] J. Weickert, B.M. ter Haar Romeny, and M.A. Viergever, "Efficient and Reliable Schemes for Nonlinear Filtering," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 398-410, Mar. 1998.
- [22] E. Hostalkova, V. Oldrich, and A. Prochazka, "Multi-Dimensional Biomedical Image De-Noising Using Haar Transform," *Proc. 15th Int'l Conf. Digital Signal Processing*, pp. 175-178, 2007.
- [23] P. Coupe, P. Hellier, S. Prima, C. Kervrann, and C. Barillot, "3D Wavelet Subbands Mixing for Image Denoising," *Int'l J. Biomedical Imaging*, vol. 2008, pp. 1-11, 2008.
- [24] A. Waiselle, J.-L. Starck, and J. Fadili, "3D Image Restoration with the Curvelet Transform," *Proc. Astronomical Data Analysis V*, 2008.

- [25] P. Coupe, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An Optimized Blockwise Nonlocal Means Denoising Filter for 3D Magnetic Resonance Images," *IEEE Trans. Medical Imaging*, vol. 27, no. 4, pp. 425-441, Apr. 2008.
- [26] H. Lu, J.-H. Cheng, G. Han, L. Li, and Z. Liang, "A 3D Distance-Weighted Wiener Filter for Poisson Noise Reduction in Sinogram Space for Spect Imaging," *Proc. SPIE Series*, vol. 4320, pp. 905-913, 2001.
- [27] P. Qiu, *Image Processing and Jump Regression Analysis*. John Wiley, 2005.
- [28] J. Sun and P. Qiu, "Jump Detection in Regression Surfaces Using Both First-Order and Second-Order Derivatives," *J. Computational and Graphical Statistics*, vol. 16, pp. 289-311, 2007.
- [29] J. Fan and I. Gijbels, *Local Polynomial Modelling and Its Applications*. Chapman & Hall, 1996.
- [30] G.Z. Yang, P. Burger, D.N. Firmin, and S.R. Underwood, "Structure Adaptive Anisotropic Image Filtering," *Image and Vision Computing*, vol. 14, pp. 135-145, 1996.
- [31] F. Chabat, G.Z. Yang, and D.M. Hansell, "A Corner Orientation Detector," *Image and Vision Computing*, vol. 17, pp. 761-769, 1999.
- [32] G.A.F. Seber, *Linear Regression Analysis*. John Wiley, 1977.
- [33] A. Macovski, "Noise in MRI," *Magnetic Resonance in Medicine*, vol. 36, pp. 494-497, 1996.
- [34] M. Martin-Fernandez, E. Muoz-Moreno, L. Cammoun, J.P. Thiran, C.F. Westin, and C. Alberola-Lpez, "Sequential Anisotropic Multichannel Wiener Filtering with Rician Bias Correction Applied to 3D Regularization of DWI Data," *Medical Image Analysis*, vol. 13, pp. 19-35, 2009.
- [35] T. Le, "A Variational Approach to Reconstructing Images Corrupted by Poisson Noise," *J. Math. Imaging and Vision*, vol. 27, pp. 257-263, 2007.
- [36] V. Epanechnikov, "Non-Parametric Estimation of a Multivariate Probability Density," *Theory of Probability and Its Applications*, vol. 14, pp. 153-158, 1969.
- [37] N. Wiest-Daessle, S. Prima, P. Coupe, S.P. Morrissey, and C. Barillot, "Rician Noise Removal by Non-Local Means Filtering for Low Signal-to-Noise Ratio mri: Applications to Dt-mri," *Proc. 11th Int'l Conf. Medical Image Computing and Computer-Assisted Intervention*, pp. 171-179, 2008.



**Peihua Qiu** received the BS degree in mathematics and the MS degree in statistics from Fudan University, Shanghai, China, and the MS degree in statistics from the University of Georgia, Athens, and the PhD degree in statistics from the University of Wisconsin-Madison. From 1996 to 1998, he was with the Biostatistics Center at the Ohio State University, where he worked on biostatistical research and consulting. He was an assistant professor during

1998-2002, an associate professor during 2002-2007, and has been a full professor since 2007, all in the School of Statistics at the University of Minnesota, Minneapolis. He is an elected fellow of the American Statistical Association, an elected fellow of the Institute of Mathematical Statistics, and an elected member of the International Statistical Institute. Currently, he serves as an associate editor for the *Journal of the American Statistical Association*, *Technometrics*, *Biometrics*, and *Statistical Papers*. His research interests include curve and surface estimation from noisy data, image segmentation, image denoising, image deblurring, image registration, statistical process control, survival analysis, and longitudinal data analysis.



**Partha Sarathi Mukherjee** received the bachelor's of statistics degree and the master's of statistics degree from the Indian Statistical Institute, Kolkata, and the MS degree in statistics from the University of Minnesota. Currently, he is a fifth year PhD student in the school of Statistics at University of Minnesota. His current research interests include 2D and 3D image restoration, medical imaging, and jump regression analysis.

► **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**