3-D Image Denoising By Local Smoothing And Nonparametric Regression

Partha Sarathi Mukherjee and Peihua Qiu
School of Statistics
University of Minnesota
Minneapolis, MN 55455

Abstract

Three-dimensional (3-D) images are becoming increasingly popular in image applications, such as magnetic resonance imaging (MRI), functional MRI (fMRI), and other image applications. Observed 3-D images often contain noise that should be removed beforehand for improving the reliability of subsequent image analyses. In the literature, most existing image denoising methods are for 2-D images. Their direct extensions to 3-D cases generally can not handle 3-D images efficiently, because the structure of 3-D images is often substantially more complicated than that of 2-D images. For instance, edge locations are surfaces in 3-D cases, which are much more challenging to handle, compared to edge curves in 2-D cases. In this paper, we propose a novel 3-D image denoising procedure, based on nonparametric estimation of a 3-D jump surface from noisy data. One important feature of this method is its ability to preserve edges and major edge structures, such as intersections of two edge surfaces, pyramids, pointed corners, and so forth. Both theoretical arguments and numerical studies show that it works well in various applications. Software and proofs are available online as supplemental material.

*K*ey Words: 3-D image denoising, cones, edges, edge structure, eigenvalue, eigenvector, jump-preserving surface estimation, local smoothing, nonparametric regression.

1 Introduction

Three-dimensional (3-D) images are becoming increasingly popular in applications. For instance, in magnetic resonance imaging (MRI) and functional MRI (fMRI), to study the biological mechanism of a 3-D object (e.g., a patient's head), people traditionally acquire a set of two-dimensional

(2-D) images from slices of the 3-D object. Then, the 3-D object is reconstructed from the 2-D images, which is a research area called 3-D image reconstruction in the literature (e.g., Sonka et al. 2008, Chapters 11 and 12). 3-D image reconstruction is technically challenging, and the reconstructed 3-D image often contains substantial error in estimating the true 3-D image. Thanks to the rapid progress in image acquisition techniques, we can now acquire 3-D images directly in certain applications, including MRI and fMRI. However, observed 3-D images often contain noise due to hardware imperfections and other reasons. Noise removal is important for the reliability of subsequent image analyses. This paper focuses on noise removal in 3-D images.

In the literature, most image denoising methods are for analyzing 2-D images. These methods include Markov random field (MRF) modeling (e.g., Geman and Geman 1984, Besag 1986, Godtliebsen and Sebastiani 1994), local median and other robust filtering methods (e.g., Sun et al. 1994, Hillebrand and Müller 2007), bilateral filtering methods (e.g., Chu et al. 1998, Tomasi and Manduchi 1998), adaptive smoothing algorithms (e.g., Saint-Marc et al. 1991, Polzehl and Spokoiny 2000, Takeda et al. 2007, Takeda and Milanfar 2009), diffusion filtering methods (e.g., Perona and Malik 1990, Barash 2002), wavelet transformation methods (e.g., Chang et al. 2000, Portilla et al. 2003), jump surface estimation methods (e.g., Qiu 1998, Gijbels et al. 2006, Qiu and Mukherjee 2010), among many others. See Buades et al. (2005) and Qiu (2007) for an overview on this topic.

Some 2-D image denoising procedures have been generalized for analyzing 3-D images. For instance, 3-D image denoising based on minimization of Total Variation (TV) is popular in the computer science literature (e.g., Keeling 2003, Wang and Zhou 2006). The TV approach was first suggested by Rudin et al. (1992) for denoising 2-D images, and one particular algorithm to accomplish that method is due to Chambolle (2004). Several MATLAB programs based on Chambolle's algorithm have been developed recently for denoising 3-D images (e.g., Getreuer 2007). MATLAB programs for 3-D image denoising using anisotropic diffusion are also available (e.g., Lopes 2007). Other 3-D image denoising procedures include the ones based on 3-D wavelet transformations (e.g., Weickert et al 1998, Hostalkova et al. 2007, Coupe 2008a, Woiselle et al. 2008), non-local means (e.g., Coupe 2008b), distance-weighted Wiener filtering (e.g., Lu et al. 2005), 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. We notice that the structure of a typical 3-D image is much more complicated than that of a 2-D image. For instance, edge locations are surfaces in 3-D cases and they usually have much more complicated structure than the edge curves in 2-D cases. Besides planar parts, an edge surface can have complicated structures, such as intersections of two or more edge surfaces, pyramids, pointed corners, and so forth. See Figure 1 for a demonstration. Most existing image denoising procedures mentioned above can preserve edges at places where the curvature of the edge surfaces is small (i.e., the planar parts). At places where the curvature of the edge surfaces is large (e.g., intersections of two or more edge surfaces, pyramids, and pointed corners), however, the edge features would be blurred or rounded by them. One major reason why this would happen is that the edge features are hidden in the observed image intensities, making them too complicated to be described or measured mathematically, and difficult to accommodate in the image denoising process. In our opinion, major edge features are an important part 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 those relatively planar parts of the edge surfaces. Therefore, they should be well preserved during image denoising. In other words, a good image denoising procedure should preserve not only the planar parts of the edge surfaces but also the major edge features, although the latter goal is much more challenging than the former.

Figure 1: Different structures of edge surfaces.

In this paper, we propose a 3-D image denoising procedure which can preserve edges and major edge features well. Our procedure consists of three major steps, briefly outlined below. First, edge pixels are detected in the whole design space by an edge detector. Second, in a neighborhood of a given pixel, the underlying edge surface is estimated from the detected edge pixels by an

algorithm that takes into account three possible scenarios of the edge surface. Finally, observed image intensities located on the same side of the estimated edge surface, as the given pixel, are averaged by the local linear kernel smoothing procedure for estimating the true image intensity at the given pixel.

The remaining part of the article is organized as follows. Our proposed 3-D image denoising procedure is described in detail in Section 2. Some of its statistical properties are discussed in Section 3. In Section 4, we present some numerical examples to evaluate its numerical performance. Some remarks conclude the article in Section 5. Proofs of the two theorems in Section 3 are provided online as supplementary material.

2 Methodology

We present our proposed methodology in three parts. Section 2.1 describes a 3-D edge detection procedure based on local linear kernel (LLK) smoothing. Local approximation of edge surfaces and local image denoising are described in Section 2.2. Data driven parameter selection is discussed in Section 2.3.

2.1 3-D edge detection by LLK smoothing

As discussed in Section 1, the first step of the proposed 3-D image denoising procedure is to detect edge pixels of a 3-D image using an edge detector. In the literature, there are many edge detectors for analyzing 2-D images (cf., e.g., Canny 1986, Qiu and Yandell 1997, Sun and Qiu 2007). These edge detectors can generally be extended for analyzing 3-D images. Theoretically speaking, any reasonable 3-D edge detector can be used in the first step of our 3-D image denoising procedure. In this paper, we introduce a 3-D edge detector based on LLK smoothing, which can be regarded as a modification of the 2-D edge detector in Qiu and Yandell (1997). Because it is numerically convenient and efficient for detecting edges, it is used in all numerical examples of the paper.

Assume that a 3-D image follows the regression model

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

where $\{(x_i, y_j, z_k) = (i/n, j/n, k/n),\ i, j, k = 1, 2, \ldots, n\}$ are the equally spaced design points (or pixels) 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 σ^2, $f(x, y, z)$ is an unknown regression function denoting the image intensity function, and $N = n^3$ is the sample size. We further assume that there exists a partition $\{\Lambda_l,\ l = 1, 2, \ldots, s\}$ of the design space Ω such that: (i) each Λ_l is a connected region in Ω; (ii) $\bigcup_{l=1}^{s} \Lambda_l = \Omega$; (iii) $f(x, y, z)$ is continuous in $\Lambda_l \backslash \partial\Lambda_l$, for $l = 1, 2, \ldots, s$, where $\partial\Lambda_l$ is the boundary point set of Λ_l, and (iv) there exist at most a finite number of line segments $\{\ell_l,\ l = 1, 2, \ldots, s^*\}$ in $\bigcup_{i=1}^{s} \partial\Lambda_i$ such that for each line segment ℓ_l there are Λ_{l_1},$\Lambda_{l_2} \in \{\Lambda_l,\ l = 1, 2, \ldots, s\}$ satisfying $\ell_l \subseteq \partial\Lambda_{l_1} \bigcap \partial\Lambda_{l_2}$ and for any $(x^*, y^*, z^*) \in \ell_l$

$$\lim_{(x,y,z)\to(x^*,y^*,z^*),(x,y,z)\in\Lambda_{l_1}} f(x, y, z) = \lim_{(x,y,z)\to(x^*,y^*,z^*),(x,y,z)\in\Lambda_{l_2}} f(x, y, z).$$

Then, we call $D := [\bigcup_{l=1}^{s} \partial\Lambda_l] \bigcap \Omega$ the *jump location surfaces (JLSs)* of $f(x, y, z)$. Obviously, JLSs describe the places where f has jumps. So, they also describe edge surfaces.

To detect edge pixels at a given point $(x, y, z) \in \Omega$, let us consider its spherical neighborhood

$$O^*(x, y, z) = \{(u, v, w) : (u, v, w) \in \Omega, \sqrt{(u - x)^2 + (v - y)^2 + (w - z)^2} \le h_n^*\},$$

where h_n^* is a bandwidth parameter. In $O^*(x, y, z)$, a 3-D plane is fitted using the local linear kernel (LLK) smoothing by solving the minimization problem

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

where K is a 3-D density kernel function defined in a unit ball. The solution to (a, b, c, d) of the minimization problem (2) is denoted as $(\widehat{a}(x, y, z), \widehat{b}(x, y, z), \widehat{c}(x, y, z), \widehat{d}(x, y, z))$. Then $\widehat{a}(x, y, z)$ is the LLK estimator of $f(x, y, z)$, and $(\widehat{b}(x, y, z), \widehat{c}(x, y, z), \widehat{d}(x, y, z))$ are LLK estimators of $(f'_x(x, y, z), f'_y(x, y, z), f'_z(x, y, z))$. It is not difficult to check that

$$\begin{pmatrix} \widehat{a}(x, y, z) \\ \widehat{b}(x, y, z) \\ \widehat{c}(x, y, z) \\ \widehat{d}(x, y, z) \end{pmatrix} = \begin{pmatrix} w_{000} & w_{100} & w_{010} & w_{001} \\ w_{100} & w_{200} & w_{110} & w_{101} \\ w_{010} & w_{110} & w_{020} & w_{011} \\ w_{001} & w_{101} & w_{011} & w_{002} \end{pmatrix}^{-1} \begin{pmatrix} \sum \xi_{ijk} K_{ijk} \\ \sum \xi_{ijk}(x_i - x) K_{ijk} \\ \sum \xi_{ijk}(y_j - y) K_{ijk} \\ \sum \xi_{ijk}(z_k - z) K_{ijk} \end{pmatrix}, \tag{3}$$

where $K_{ijk} = K(\frac{x_i - x}{h_n^*}, \frac{y_j - y}{h_n^*}, \frac{z_k - z}{h_n^*})$, $w_{i_1 i_2 i_3} = \sum (x_i - x)^{i_1} (y_j - y)^{i_2} (z_k - z)^{i_3} K_{ijk}$, for $i_1, i_2, i_3 = 0, 1, 2$, and $\sum$ is the sum over all design points.

The estimated gradient vector $\widehat{\boldsymbol{\beta}}(x, y, z) = (\widehat{b}(x, y, z), \widehat{c}(x, y, z), \widehat{d}(x, y, z))^T$ provides an estimate of the direction that the underlying regression function f increases the fastest. So, if its magnitude is larger, f would be steeper around (x, y, z), and it is more likely that (x, y, z) is an edge pixel. However, when f is steep but continuous in $O^*(x, y, z)$, $\widehat{\boldsymbol{\beta}}(x, y, z)$ can also have a relatively large magnitude. To remove this slope effect, we consider two neighboring design points (x_{N1}, y_{N1}, z_{N1}) and (x_{N2}, y_{N2}, z_{N2}) along the direction of $\widehat{\boldsymbol{\beta}}(x, y, z)$. Their spherical neighborhoods of size h_n^* do not overlap with $O^*(x, y, z)$, but are adjacent to $O^*(x, y, z)$ on either side. Intuitively, if (x, y, z) is on a JLS, then (x_{N1}, y_{N1}, z_{N1}) and (x_{N2}, y_{N2}, z_{N2}) would be on two different sides of the JLS. Based on this intuition, we define the following jump detection criterion:

$$\delta(x, y, z) = \min\{\|\widehat{\boldsymbol{\beta}}(x, y, z) - \widehat{\boldsymbol{\beta}}_{N_1}(x, y, z)\|, \|\widehat{\boldsymbol{\beta}}(x, y, z) - \widehat{\boldsymbol{\beta}}_{N_2}(x, y, z)\|\},$$

where $\widehat{\boldsymbol{\beta}}_{N_1}(x, y, z)$ and $\widehat{\boldsymbol{\beta}}_{N_2}(x, y, z)$ are the estimated gradients in the neighborhoods of (x_{N1}, y_{N1}, z_{N1}) and (x_{N2}, y_{N2}, z_{N2}), respectively, and $\|\cdot\|$ is the Euclidean norm. If there is no jump in the three neighborhoods, then $\widehat{\boldsymbol{\beta}}(x, y, z)$, $\widehat{\boldsymbol{\beta}}_{N_1}(x, y, z)$ and $\widehat{\boldsymbol{\beta}}_{N_2}(x, y, z)$ should be close to each other. Hence, $\delta(x, y, z)$ is small. On the other hand, if (x, y, z) is on a JLS, then $\delta(x, y, z)$ would be relatively large, due to the jump. Therefore, $\delta(x, y, z)$ can be used for detecting jumps. The point (x, y, z) is then detected as a jump point if

$$\delta(x, y, z) > u_n, \tag{4}$$

where u_n is a threshold parameter.

2.2 Local Approximation to the Underlying Jump Location Surfaces

As discussed in Section 1, JLSs of a 3-D image could have a number of different structures (cf., Figure 1). To preserve the major edge structures, we consider estimation of the underlying JLS in the following neighborhood of a given point $(x, y, z) \in \Omega$:

$$O(x, y, z) = \{(u, v, w) : (u, v, w) \in \Omega, \sqrt{(u - x)^2 + (v - y)^2 + (w - z)^2} \leq h_n\},$$

where h_n is a bandwidth that could be different from h_n^* used in edge detection (2). To this end, the following three cases are considered. (i) The underlying JLS in $O(x, y, z)$ is planar, and can be well approximated by a local plane. (ii) The JLS in $O(x, y, z)$ contains a ridge or valley, and can be well approximated by two crossing half-planes. (iii) The JLS contains a pointed part in $O(x, y, z)$, and can be approximated reasonably well by a cone. See Figure 2 for a demonstration of these three basic edge structures. In reality, the JLSs may have more complicated structures than the ones considered here. For instance, they may contain a pyramid, saddle points, or an elliptically pointed part. However, with the three basic structures, the major edge features can be preserved well, which is supported by the numerical examples presented in Section 4. In the next four parts, we describe our proposals to estimate the three basic edge structures from the detected edge pixels, and to choose one of them for estimating the underlying JLS.

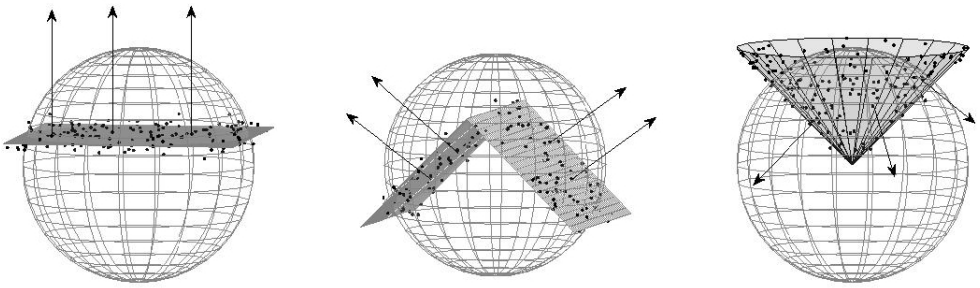

Figure 2: Three basic edge structures used for approximating the underlying JLS in a spherical neighborhood of a given design point. In each plot, the dots denote the detected edge pixels, shaded surface denotes approximation of surface and the arrows point to its normal direction.

2.2.1 Approximation to the JLS in $O(x, y, z)$ by a local plane

In cases when the underlying JLS is planar in $O(x, y, z)$ (cf., the left panel of Figure 1), we suggest approximating it by a local plane constructed as follows.

(i) The plane passes the center (c_x, c_y, c_z) of the detected edge pixels in $O(x, y, z)$.

(ii) Its normal direction is determined by the eigenvector of the largest eigenvalue of

$$G = \frac{1}{m}\Psi\Psi^T, \tag{5}$$

where $\Psi = (\widehat{\boldsymbol{\beta}}_1^*, \widehat{\boldsymbol{\beta}}_2^*, \ldots, \widehat{\boldsymbol{\beta}}_m^*)$, and $\{\widehat{\boldsymbol{\beta}}_l^*, l = 1, 2, \ldots, m\}$ denote estimated gradient directions (with unit lengths) at all m detected edge pixels in $O(x, y, z)$.

See the left panel of Figure 2 for a demonstration, where the dots denote the detected edge pixels, the arrows denote the direction determined by the eigenvector of the largest eigenvalue of G, and the shaded plane denotes the plane for approximation. Lemma 3 in the supplementary file shows that the approximation plane converges almost surely to the true tangent plane of f at (x, y, z) under some regularity conditions.

2.2.2 Approximation to the JLS in $O(x, y, z)$ by two crossing half-planes

In cases when the underlying JLS has a ridge or valley in $O(x, y, z)$ (cf., the middle panel of Figure 1), we suggest approximating it by two crossing half-planes as follows.

(i) Calculate the eigenvector corresponding to the smallest eigenvalue of G, which should be a good estimate of the direction of the ridge/valley contained in the underlying JLS, because the JLS has the smallest variation along that direction.

(ii) Determine the plane P that passes (c_x, c_y, c_z) along the estimated ridge/valley direction obtained in step (i) and along $\overline{\boldsymbol{\beta}}^*$ as well, where $\overline{\boldsymbol{\beta}}^*$ is the average of $\{\widehat{\boldsymbol{\beta}}_l^*, l = 1, 2, \ldots, m\}$.

(iii) Divide all detected edge pixels in $O(x, y, z)$ into two groups as follows. We first determine the direction that is orthogonal to the plane P, and then determine the two groups of the detected edge pixels based on the signs of their inner products with the orthogonal direction. Those detected edge pixels with non-negative inner products belong to one group, and those with negative inner products belong to the other group.

(iv) For each group of the detected edge pixels, determine a plane that passes the center of that group of pixels in the direction orthogonal to the averaged gradient direction within that group of pixels.

(v) The two crossing half-planes are then obtained from the two planes computed in step (iv). The subspace of $O(x, y, z)$ formed by them should contain (c_x, c_y, c_z).

2.2.3 Approximation to the JLS in $O(x, y, z)$ by a cone

In cases when the underlying JLS contains a pointed part in $O(x, y, z)$ (cf., the right panel of Figure 1), we suggest approximating it by a cone, which can be uniquely determined by specifying its central axis, its vertex, and the half cone angle (i.e., the angle between the central axis and a straight line on the lateral surface), as follows.

(i) The direction of the central axis of the cone is estimated by the direction $(1, d_2, d_3)^T$ that minimizes the sample variance of its inner products with $\{\widehat{\boldsymbol{\beta}}_l^*, l = 1, 2, \ldots, m\}$. Since the angle between the central axis of the cone and the normal direction at any point on the lateral surface of the cone is a constant, this estimate should be reasonable to use. Simple calculations show that $d_2 = (\Psi_{23}\Psi_{13} - \Psi_{33}\Psi_{12})/(\Psi_{22}\Psi_{33} - \Psi_{23}^2)$ and $d_3 = (\Psi_{12}\Psi_{23} - \Psi_{22}\Psi_{13})/(\Psi_{22}\Psi_{33} - \Psi_{23}^2)$, where $\Psi_{j_1 j_2}$ is the sample covariance of the (j_1, j_2)th components of $\{\widehat{\boldsymbol{\beta}}_l^*, l = 1, 2, \ldots, m\}$, for $j_1, j_2 = 1, 2, 3$.

(ii) The half cone angle is estimated by the complement of the averaged angle between the direction of the central axis specified in step (i) and $\{\widehat{\boldsymbol{\beta}}_l^*, l = 1, 2, \ldots, m\}$, which is denoted as $\widehat{\theta}$.

(iii) To specify the location of the central axis, let us consider a sphere $\widetilde{O}(x, y, z)$ that is centered at (x, y, z) and has radius $\widetilde{h}_n > h_n$. The plane $\widetilde{P}$ passing (c_x, c_y, c_z) and having the normal direction $(1, d_2, d_3)^T$ would divide $\widetilde{O}(x, y, z)$ into two parts. Centers of the detected edge pixels in these parts are then calculated, and the one closer to $\widetilde{P}$ is denoted as (c_x^*, c_y^*, c_z^*). Then, the line passing (c_x^*, c_y^*, c_z^*) along the direction $(1, d_2, d_3)^T$ is used as the central axis of the cone. In all numerical examples presented in Section 4, we choose $\widetilde{h}_n = 3h_n$.

(iv) The vertex location (v_x, v_y, v_z) of the cone is estimated by minimizing the orthogonal distance between the cone and the detected edge pixels in $O(x, y, z)$. After some algebraic manipulations, the estimator can be calculated by

$$(\widehat{v}_x, \widehat{v}_y, \widehat{v}_z) = (c_x^* + \overline{\beta}_1^* t, c_y^* + \overline{\beta}_2^* t, c_z^* + \overline{\beta}_3^* t),$$

where $\overline{\beta}_1^*$, $\overline{\beta}_2^*$, and $\overline{\beta}_3^*$ are three components of $\overline{\boldsymbol{\beta}}^*$, t is one of $[\sum d_l t_l/(\tan\widehat{\theta}||\overline{\boldsymbol{\beta}}^*||) + \sum t_l^2]/\sum t_l$ and $[\sum d_l t_l/(\tan\widehat{\theta}||\overline{\boldsymbol{\beta}}^*||) - \sum t_l^2]/(-\sum t_l)$ that minimizes $\sum(d_l \cos\widehat{\theta} - |t -$

$t_l|||\overline{\boldsymbol{\beta}}^*|| \sin\widehat{\theta})^2$,

$$
\begin{aligned}
t_l &= \frac{\overline{\beta}_1^*(x_l^* - c_x) + \overline{\beta}_2^*(y_l^* - c_y) + \overline{\beta}_3^*(z_l^* - c_z)}{||\overline{\boldsymbol{\beta}}^*||^2}, \\
d_l &= \sqrt{(x_l^* - c_x^* - \overline{\beta}_1^* t_l)^2 + (y_l^* - c_y^* - \overline{\beta}_2^* t_l)^2 + (z_l^* - c_z^* - \overline{\beta}_3^* t_l)^2},
\end{aligned}
$$

(x_l^*, y_l^*, z_l^*) denotes the l-th detected edge pixel in $O(x, y, z)$, and $\sum$ is over all detected edge pixels in $O(x, y, z)$.

2.2.4 Selection of the local surface

In the previous three parts, we have discussed how to approximate the underlying JLS in $O(x, y, z)$ using one of the three basic surfaces shown in Figure 2. In practice, we need to choose one of the three surfaces based on observations in $O(x, y, z)$ for estimating $f(x, y, z)$. To this end, various model selection criteria, including the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and so forth, have been considered. For instance, by the BIC, the fitted surface minimizing the following BIC criterion should be chosen:

$$
BIC(x, y, z) = m \log\left(\frac{RSS(x, y, z)}{m}\right) + k \log(m), \tag{6}
$$

where $RSS(x, y, z)$ denotes the sum of squares of the orthogonal distances from the individual detected edge pixels to the fitted surface in question, m is the number of detected edge pixels in $O(x, y, z)$, and k is the total number of parameters in the model of the surface. For the local plane, the two crossing half-planes, and the cone, their numbers of parameters are 3, 6, and 7, respectively. By the AIC, the second term on the right side of (6) should be replaced by $2k$.

In our numerical studies, we tried both AIC and BIC methods, and found that results with the BIC method are usually better in terms of the mean squared error (MSE) of the estimated f. This might be consistent with the findings in Nishii (1984) that the probability of choosing the true model by the BIC tends to 1 when the sample size increases, as long as the true model is included in the candidate set. In the current problem, most part of the true JLSs should be well approximated by one of the three basic surfaces considered. For this reason, we use the BIC in all numerical examples presented in Section 4.

2.3 Proposed 3-D Image Denoising Procedure

After the underlying JLS is approximated by a local surface in the neighborhood $O(x, y, z)$, $O(x, y, z)$ can be divided into two parts by the approximation surface, denoted as $O_1(x, y, z)$ and $O_2(x, y, z)$. Without loss of generality, we assume that (x, y, z) is contained in $O_1(x, y, z)$. Then, $f(x, y, z)$ can be estimated by the solution to a, denoted as $\widehat{f}(x, y, z)$, of the following minimization problem:

$$\min_{a,b,c,d} \sum_{(x_i,y_j,z_k)\in O_1(x,y,z)} \{\xi_{ijk} - [a + b(x_i - x) + c(y_j - y) + d(z_k - z)]\}^2 K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}, \frac{z_k - z}{h_n}\right). \tag{7}$$

From (2) and (7), $\widehat{f}(x, y, z)$ is a weighted average of the observations whose design points are located on the same side of the approximation surface in $O(x, y, z)$ as the given point (x, y, z). Intuitively, as long as the the approximation surface estimates the underlying JLS well, $\widehat{f}(x, y, z)$ should preserve edges and major edge features well.

For a real image, there are regions where f is smooth. In these regions, the number of detected edge pixels should be small. Also, in such regions, a relatively larger bandwidth is prefered to increase the noise removal ability of the procedure. So, before estimating f using (7), we suggest counting the number of detected edge pixels in $O(x, y, z)$. If the number is large so that a potential JLS is likely in $O(x, y, z)$ (e.g., larger than or equal to $(nh_n)^2/4$), then estimate f using (7). Otherwise, consider a larger spherical neighborhood $\breve{O}(x, y, z)$ with bandwidth $\breve{h}_n > h_n$. In $\breve{O}(x, y, z)$, if the number of detected edge pixels is still smaller than $(n\breve{h}_n)^2/4$, then it is unlikely to have a JLS in $\breve{O}(x, y, z)$ and we suggest estimating $f(x, y, z)$ simply by the conventional LLK estimator constructed from all observations in $\breve{O}(x, y, z)$. If the number of detected edge pixels in $\breve{O}(x, y, z)$ is larger than or equal to $(n\breve{h}_n)^2/4$, then it is possible to have a JLS in $\breve{O}(x, y, z)$, but the JLS is unlikely to be in $O(x, y, z)$. In such cases, we suggest estimating $f(x, y, z)$ by the conventional LLK estimator constructed in $O(x, y, z)$. By this modification, procedure (7) is used only when the number of detected edge pixels in $O(x, y, z)$ is relatively large. To do so, there are at least two benefits. One is that much computation is saved, because the conventional LLK estimator is much easier to compute, compared to the one-sided estimator obtained by (7). The other benefit is that the estimated f would be more efficient, because it is constructed from all observations

in $O(x,y,z)$ or $\breve{O}(x,y,z)$, instead of from part of observations in $O(x,y,z)$, in cases when the number of detected edge pixels in $O(x,y,z)$ is small. Considering the fact that most points in the design space are continuity points of f, these benefits are substantial. Regarding $\breve{h}_n$, based on our numerical experience, we can simply choose $\breve{h}_n = ch_n$ with $c \in [1.25, 2]$. In all numerical examples considered in Section 4, we choose $c = 1.75$ and the results are satisfactory. Now, we can summarize the proposed 3-D image denoising procedure as follows.

Proposed 3-D Image Denoising Procedure

- Detect edge pixels by the procedure described in Section 2.1.
- For a given pixel (x,y,z), count the number of detected edge pixels in $O(x,y,z)$. If this number is smaller than $(nh_n)^2/4$, then consider a larger spherical neighborhood $\breve{O}(x,y,z)$ with bandwidth $\breve{h}_n$. If the number of detected edge pixels in $\breve{O}(x,y,z)$ is still smaller than $(n\breve{h}_n)^2/4$, then define $\widehat{f}(x,y,z)$ to be the conventional LLK estimate in $\breve{O}(x,y,z)$. Otherwise, define $\widehat{f}(x,y,z)$ to be the conventional LLK estimate in $O(x,y,z)$. Continue the denoising procedure for the next pixel.
- If the number of detected edge pixels in $O(x,y,z)$ is larger than or equal to $(nh_n)^2/4$, then define $\widehat{f}(x,y,z)$ by (7), after the underlying JLS in $O(x,y,z)$ is approximated by one of the three basic local surfaces, as discussed in Sections 2.2.1–2.2.4. Continue the denoising procedure for the next pixel.

2.4 Selection of Procedure Parameters

In the proposed 3-D image denoising procedure, there are three parameters h_n^*, u_n, and h_n (cf., expressions (2), (4), and (7)). They should be chosen properly because performance of the proposed

image denoising procedure would depend on their values. To choose u_n, we notice that

$$\begin{aligned}
& P(\delta(x,y,z) > u_n) \\
\leq\ & P(\|\widehat{\boldsymbol{\beta}}(x,y,z) - \widehat{\boldsymbol{\beta}}_{N1}(x,y,z)\| > u_n) \\
=\ & P\left((\widehat{b}(x,y,z) - \widehat{b}_{N1}(x,y,z))^2 + (\widehat{c}(x,y,z) - \widehat{c}_{N1}(x,y,z))^2 + (\widehat{d}(x,y,z) - \widehat{d}_{N1}(x,y,z))^2 > u_n^2\right) \\
=\ & E\left\{P\left((\widehat{b}(x,y,z) - \widehat{b}_{N1}(x,y,z))^2 + (\widehat{c}(x,y,z) - \widehat{c}_{N1}(x,y,z))^2 + \right.\right. \\
& \left.\left.(\widehat{d}(x,y,z) - \widehat{d}_{N1}(x,y,z))^2 > u_n^2 \,\middle|\, \widehat{b}(x,y,z), \widehat{c}(x,y,z), \widehat{d}(x,y,z)\right)\right\}.
\end{aligned}$$

For fixed $\widehat{b}(x,y,z)$, $\widehat{c}(x,y,z)$, and $\widehat{d}(x,y,z)$, $((\widehat{b}(x,y,z) - \widehat{b}_{N1}(x,y,z))^2 + (\widehat{c}(x,y,z) - \widehat{c}_{N1}(x,y,z))^2 + (\widehat{d}(x,y,z) - \widehat{d}_{N1}(x,y,z))^2)/\sigma_{N1}^2$ approximately follows the χ_3^2 distribution, under the assumption that there are no jumps in $O^*(x,y,z) \bigcup O^*(x_{N1}, y_{N1}, z_{N1}) \bigcup O^*(x_{N2}, y_{N2}, z_{N2})$, where $\sigma_{N1}^2 = \text{Var}(\widehat{b}_{N1}(x,y,z))$. From expressions in (3), we have

$$\sigma_{N1}^2 = \sigma^2 \frac{\sum (x_i - x_{N1})^2 K_{N1}^2}{\{(\sum (x_i - x_{N1}) K_{N1}\}^2},$$

where $K_{N1} = K\left(\frac{x_i - x_{N_1}}{h_n^*}, \frac{y_j - y_{N_1}}{h_n^*}, \frac{z_k - z_{N_1}}{h_n^*}\right)$. Therefore, a natural choice for u_n is

$$u_n = \widehat{\sigma} \sqrt{\frac{\chi_{3,\alpha_n}^2 \sum (x_i - x_{N1})^2 K_{N1}^2}{(\sum (x_i - x_{N1})^2 K_{N1})^2}}, \tag{8}$$

where χ_{3,α_n}^2 is the $1 - \alpha_n$ quantile of the χ_3^2 distribution and $\widehat{\sigma}$ is a consistent estimator of σ. For simplicity, we can define $\widehat{\sigma}$ to be the residual mean squares of the conventional LLK estimator of f defined in $O^*(x,y,z)$, and α_n can be specified beforehand to be a small number, say, $\alpha_n = 0.001$.

To choose h_n^* and h_n, we suggest using the cross-validation (CV) procedure with the following CV score:

$$CV(h_n^*, h_n) = \frac{1}{n^3} \sum_{i,j,k=1}^{n} \left(\xi_{ijk} - \widehat{f}_{-i,-j,-k}(x_i, y_j, z_k)\right)^2, \tag{9}$$

where $\widehat{f}_{-i,-j,-k}(x_i, y_j, z_k)$ is the estimate of $f(x_i, y_j, z_k)$ when we do not include the (i,j,k)-th pixel (x_i, y_j, z_k) in all subsequent steps of the proposed image denoising procedure after edge detection. Then, h_n^* and h_n are chosen to be the minimizer of $CV(h_n^*, h_n)$. Note that, when computing $\widehat{f}_{-i,-j,-k}(x_i, y_j, z_k)$, the (i,j,k)-th pixel is still used in edge detection, which is for simplifying computation. By (9), we only need to detect edges once in the whole design space. Based on our numerical experience, it would not change the value of $\widehat{f}_{-i,-j,-k}(x_i, y_j, z_k)$ much

to include the (i, j, k)-th pixel or not in edge detection. In the statistical literature, there are some alternative methods for choosing smoothing parameters like h_n^* and h_n, including the Mallow's C_p, plug-in algorithms, bootstrap, and so forth (e.g., Marron 1988, Loader 1999, Hall and Robinson 2009). The CV method is used here mainly because of its simplicity, which is especially relevant for 3-D image analysis where computation is generally extensive.

3 Some Statistical Properties

In this section, we discuss some statistical properties of the estimated image intensity function $\widehat{f}$. In our description, a point (x, y, z) on the JLSs is called a *singular point* if one of the following conditions is satisfied. (i) There exists some constant $\nu > 0$ such that, for any $0 < \tilde{\nu} < \nu$, the spherical neighborhood of (x, y, z) with diameter $\tilde{\nu}$ is divided into more than two connected regions by the JLSs. (ii) There exists a direction along which the JLSs do not have the one-sided directional tangent line at (x, y, z). (iii) The jump magnitude of f at (x, y, z) is 0. (iv) f does not have a tangent plane at (x, y, z), and (x, y, z) is neither on a ridge/valley of the JLSs nor a vertex of a circular cone (cf., the middle and right panels of Figure 2). All other points on the JLSs are called nonsingular points. Also, we define

$$
\begin{aligned}
\Omega_\epsilon &= [\epsilon, 1-\epsilon] \times [\epsilon, 1-\epsilon] \times [\epsilon, 1-\epsilon], \\
J_\epsilon &= \{(x, y, z) : (x, y, z) \in \Omega, d_E((x, y, z), (x^*, y^*, z^*)) \le \epsilon, \text{ for some } (x^*, y^*, z^*) \in D\}, \\
S_\epsilon &= \{(x, y, z) : (x, y, z) \in \Omega, d_E((x, y, z), (x^*, y^*, z^*)) \le \epsilon, \text{ for a singular point } (x^*, y^*, z^*) \text{ in } D\}, \\
\Omega_{\bar{J},\epsilon} &= \Omega_\epsilon \backslash J_\epsilon, \\
\Omega_{\bar{S},\epsilon} &= \Omega_\epsilon \backslash S_\epsilon,
\end{aligned}
$$

where ϵ is a small positive constant, d_E denotes the Euclidean distance, and D denotes the set of points on the JLSs. Let $\widehat{D}_n = \{(x_i, y_j, z_k) : \delta(x_i, y_j, z_k) > u_n\}$ be the set of all detected edge pixels. Then, we have

Theorem 3.1 Assume that f has continuous first order partial derivatives over $(0, 1) \times (0, 1) \times (0, 1)$ except on the JLSs, and its first order partial derivatives have one-sided limits at non-singular points of the JLSs on both sides of a given JLS; $h_n^* = o(1)$, $1/(nh_n^*) = o(1)$, and

$\log^2(n)/n^3(h_n^*)^6 = O(1)$; $E|\varepsilon_{111}^4| < \infty$; the kernel function K is a Lipschitz-1 continuous, isotropic, trivariate density function; and α_n is chosen such that (i) $\alpha_n = o(1)$, (ii) $(nh_n^*)^{7/2}/\{n^2\sqrt{-\log \alpha_n}\} = o(1)$ and (iii)$\sqrt{-\log \alpha_n/(nh_n^*)^3} = o(1)$. Then, for any small constant $\epsilon > 0$, $d_H(D \bigcap \Omega_{\bar{S},\epsilon}, \widehat{D}_n \bigcap \Omega_{\bar{S},\epsilon}) = O(h_n^*)$ a.s., where $d_H(A, B)$ denotes the Hausdorff distance between two point sets A and B, defined by

$$d_H(A,B) = \max\left\{ \sup_{(x,y,z)\in A} \inf_{(x',y',z')\in B} d_E((x,y,z),(x',y',z')), \sup_{(x,y,z)\in B} \inf_{(x',y',z')\in A} d_E((x,y,z),(x',y',z')) \right\} \blacksquare$$

Theorem 3.2: Besides the conditions in Theorem 3.1, if we further assume that $h_n = o(1)$, $1/(nh_n) = o(1)$, $h_n^*/h_n^3 = o(1)$, and $\breve{h}_n = ch_n$ where $c > 0$ is a constant, then $\|\widehat{f} - f\|_{\Omega_{\bar{J},h_n}} = \max_{(x,y,z)\in\Omega_{\bar{J},h_n}} |\widehat{f}(x,y,z) - f(x,y,z)| = O(h_n^2)$ a.s. For any small constant $\epsilon > 0$, when $(x,y,z) \in J_{h_n}\backslash S_\epsilon$, $\widetilde{h}_n = o(1)$ and $\frac{h_n}{\breve{h}_n} = o(1)$, we have $\widehat{f}(x,y,z) = f(x,y,z) + O(h_n)$ a.s. $\blacksquare$

Theorem 3.1 establishes the almost sure consistency of the detected edge pixels. If we choose $h_n^* = O(n^{-1/4})$, $\alpha_n = O(e^{-n^2})$, and all other conditions in the theorem are satisfied, then we have $d_H(D \bigcap D_\epsilon, \widehat{D}_n \bigcap D_\epsilon) = O(n^{-1/4})$ a.s. Theorem 3.2 says that the estimated image intensity function is uniformly strong consistent in the whole design space excluding a small region around the design border and the true JLSs. At a given point around the true JLSs, it is pointwise strong consistent, as long as the point in question is a small distance away from any singular points of the JLSs. Proofs of the above two theorems are given in a supplementary file.

4 Numerical Examples

In this section, we present some numerical results regarding the performance of the proposed 3-D image denoising procedure (denoted as NEW), in comparison with four existing methods that are commonly used in practice, including an anisotropic diffusion method (denoted as AD, cf., Lopes 2007), a method based on total variation minimization (denoted as TV, cf., Rudin et al. 1992, Getreuer 2007), a method based on optimized non-local means (denoted as ONLM, cf., Coupe et al. 2008b), and the conventional local median filtering method (denoted as MED). For the TV

method, the code by Getreuer (2007) interprets a 3-D image as a 2-D image with vector-valued image intensities. Here, we have modified it by minimizing a TV criterion constructed from a 3-D image directly for 3-D image denoising. In this method, there is a regularization parameter to choose. The AD method is accomplished by an iterative algorithm that contains two parameters: the diffusion parameter κ and the number of iterations of the algorithm. The ONLM method has two bandwidth parameters to choose. The median filtering method denoises the image by using the sample median of the observations in a spherical window to estimate the true image intensity at the central pixel of the window. It has one parameter to choose, which is the bandwidth of the spherical window. Our proposed denoising method NEW has three parameters h_n^*, u_n and h_n to choose.

The numerical study includes two artificial 3-D images and a real 3-D MRI test image. First, we consider the following two true image intensity functions

$$\begin{aligned} f_1(x,y,z) &= -(x-0.5)^2-(y-0.5)^2-(z-0.5)^2+ \\ &\quad I\left(\max(I(|x-0.5|\le 0.25)I(|y-0.5|\le 0.25)I(|z-0.5|\le 0.25),\right. \\ &\quad \left. I((x-0.5)^2+(y-0.5)^2\le 0.15^2)I(|z-0.5|\le 0.35))=1\right), \end{aligned}$$

$$\begin{aligned} f_2(x,y,z) &= \frac{1}{4}\sin(2\pi(x+y+z)+1)+\frac{1}{4}+I\left(\max(I((x-0.5)^2+(y-0.5)^2\le\frac{1}{4}(z-0.5)^2)\right. \\ &\quad I(z\le 0.5)I(z\ge 0.2), I((x-0.5)^2+(y-0.5)^2+(z-0.5)^2\le 0.4^2) \\ &\quad \left. I((x-0.5)^2+(y-0.5)^2+(z-0.5)^2>0.2^2)I(z<0.45))=1\right). \end{aligned}$$

Their edge surfaces are shown in the two plots of Figure 3. Basically, f_1 contains a cube with a solid cylinder in the middle, and its background and foreground are variable but smooth. From the left panel of Figure 3, we can see that the JLSs of f_1 contain planes, intersections of two planes, intersections of three planes, smooth non-linear surfaces, and intersections of a plane and a smooth non-linear surface. From the right panel of Figure 3, we can see that f_2 contains a half hollow hemisphere and a cone. The two objects are separated, although they are close to each other. The foreground of f_2 has a fixed intensity level, and its background has a variable intensity level. Therefore, jump sizes are different at different places of the JLSs.

We then apply the five image denoising procedures to the 3-D noisy images generated from

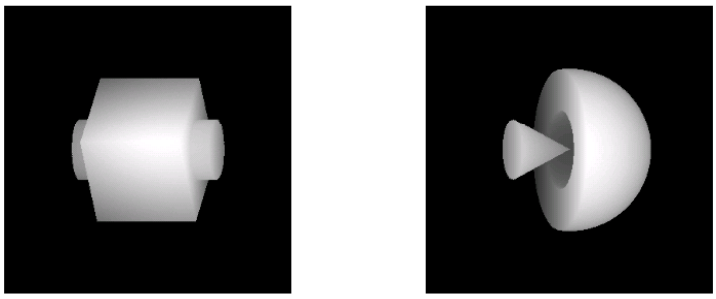

Figure 3: The left and right panels show the jump location surfaces of f_1 and f_2, respectively.

model (1) when f equals f_1 or f_2, $n = 64$ or 128 (denoting two different image resolutions), $\varepsilon_{111} \sim N(0, \sigma^2)$ with $\sigma = 0.1$, 0.2 or 0.3 (representing low, medium and high noise levels). Because procedures ONLM, TV, AD and MED do not have corresponding data-driven parameter selection algorithms yet, to make a fair comparison, we search their procedure parameters by minimizing the MISE value estimated by the sample mean of

$$ISE(\widehat{f}, f) = \frac{1}{n^3} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \Big(\widehat{f}(x_i, y_j, z_k) - f(x_i, y_j, z_k) \Big)^2$$

over 100 replications, for each combination of f, σ and n, where $\widehat{f}$ denotes the denoised image of the related denoising method.

The MISE criterion provides us a measure of the overall performance of a denoising procedure. But, it can not tell us how well the JLSs are preserved by the denoising procedure. Hall and Qiu (2007) defined a measure of the jump size (JS) of a 2-D image. A discretized version of its 3-D generalization can be written as follows. For the true image intensity function f, its JS can be measured by

$$JS(f) = \frac{1}{(n-2)^3} \sum_{i=2}^{n-1} \sum_{j=2}^{n-1} \sum_{k=2}^{n-1} |f(x_i', y_j', z_k') - f(x_i'', y_j'', z_k'')|,$$

where (x_i', y_j', z_k') and (x_i'', y_j'', z_k'') are two pixels on two different sides of the pixel (x_i, y_j, z_k) along the $x-$, $y-$, $z-$direction, or a diagonal direction that is closest to the gradient direction of f at (x_i, y_j, z_k). Obviously, if (x_i, y_j, z_k) is an edge pixel, then $|f(x_i', y_j', z_k') - f(x_i'', y_j'', z_k'')|$ is close to the jump magnitude of f at (x_i, y_j, z_k). Otherwise, $|f(x_i', y_j', z_k') - f(x_i'', y_j'', z_k'')|$ is close to 0. So, $JS(f)$ is a reasonable measure of the accumulative jump magnitude of f along the JLSs. For a denoised image $\widehat{f}$, we can compute $JS(\widehat{f})$ similarly, by using the estimated gradient

directions of f. Then, $EP(\widehat{f}) = (JS(f) - JS(\widehat{f}))/JS(f)$ would be a reasonable measure of edge-preservation (EP) for the denoising method in question. In the numerical examples, after we denoise images by the NEW method and all other competing methods, we calculate this measure using the neighborhood size 0.0234 when $n = 64$ and 0.0117 when $n = 128$ to calculate the gradient directions. For MRI images we used it to be 0.0117. The choices of this neighborhood size seem reasonable if we compare them with the choice of h_n^* in different images listed in Tables 1–3 below.

The numerical results for f_1 and f_2 are presented in Tables 1 and 2, respectively. From Table 1, it can be seen that the proposed method NEW is uniformly better than the competing methods TV, AD, and MED in both MISE and EP, although the MISE values of the methods NEW and TV are almost the same in cases when $\sigma = 0.1$. Compared to the method ONLM, the method NEW performs better in terms of MISE in cases when $\sigma = 0.2$ and 0.3, and performs similarly in terms of MISE in cases when $\sigma = 0.1$. In terms of EP, it seems that ONLM is better when σ is small (i.e., $\sigma = 0.1$ when $n = 64$, and $\sigma = 0.1$ or 0.2 when $n = 128$). Similar conclusions can be made from the results presented in Table 2, except that the methods NEW, TV and ONLM perform similarly in this example in terms of MISE, but the method NEW is much better than the other two methods in terms of EP.

One realization of f_1 is generated from model (1) with $n = 128$, $\varepsilon_{111} \sim N(0, \sigma^2)$, and $\sigma = 0.2$. Three cross sections of this realization at $z = 0.5$, $z = 0.1875$, and $x = 0.4688$ are shown in the first column of Figure 4. The corresponding cross sections of the denoised images by TV, AD, ONLM, MED and NEW are shown in the 2–6 columns. In the denoising methods, their parameters are chosen to be the ones shown in the part with $n = 128$ and $\sigma = 0.2$ of Table 1. Similarly, three cross sections of a realization of f_2 at $x = 0.5$, $x = 0.8125$, and $x = 0.6562$, from model (1) with $n = 128$, $\varepsilon_{111} \sim N(0, \sigma^2)$, and $\sigma = 0.2$, along with the cross sections of the denoised images of the five methods, are shown in Figure 5. From the two figures, it can be seen that TV, ONLM and MED tend to blur the edges at some places, the denoised image by AD seems to contain some scattered noise, and the denoised image by NEW preserves the edges well and removes noise well. To better see the results, for the three slices of the denoised images shown in Figure 5, we present the differences between them and the corresponding slices of the true image in Figure 6. From the

Table 1: In each entry, the first line presents the estimated MISE value from 100 simulations and the corresponding standard error of ISE (in parenthesis), the second line presents the measure of edge-preservation EP and its standard error, and the third line presents the searched procedure parameter values. This table considers the case when $f = f_1$.

	n=64			n=128		
Method	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
TV	.0003 (.0000) .1011 (.0023) 11.0	.0009 (.0000) .1909 (.0037) 6.0	.0018 (.0000) .2480 (.0050) 4.5	.0001 (.0000) .1053 (.0013) 11.0	.0004 (.0000) .1930 (.0019) 6.0	.0008 (.0000) .2737 (.0024) 4.0
AD	.0006 (.0000) -.0310 (.0022) .18 , 6	.0035 (.0000) -.0457 (.0050) .4 , 4	.0082 (.0001) .1109 (.0089) .6 , 4	.0003 (.0000) -.0246 (.0009) .18 , 8	.0020 (.0000) -.0632 (.0021) .38 , 5	.0048 (.0000) .1719 (.0040) .6 , 5
ONLM	.0003 (.0000) -.0082 (.0025) 10 , 2	.0012 (.0000) .0408 (.0054) 10 , 2	.0032 (.0001) .1191 (.0075) 10 , 2	.0001 (.0000) -.0070 (.0013) 10 , 2	.0006 (.0000) .0140 (.0026) 10 , 2	.0015 (.0000) .0421 (.0035) 10 , 2
MED	.0022 (.0000) .1615 (.0023) .0313	.0033 (.0000) .2757 (.0039) .0359	.0055 (.0001) .3214 (.0051) .0359	.0008 (.0000) .1579 (.0010) .0189	.0019 (.0000) .3789 (.0014) .0250	.0033 (.0000) .4667 (.0016) .0250
NEW	.0003 (.0000) .0224 (.0025) .023 , 13 , .037	.0008 (.0000) .0369 (.0049) .023 , 22 , .050	.0012 (.0001) .0464 (.0075) .028 , 22 , .053	.0001 (.0000) .0199 (.0015) .012 , 24 , .025	.0003 (.0000) .0251 (.0027) .012 , 44 , .028	.0005 (.0000) .0277 (.0038) .014 , 44 , .028

Table 2: In each entry, the first line presents the estimated MISE value from 100 simulations and the corresponding standard error of ISE (in parenthesis), the second line presents the measure of edge-preservation EP and its standard error, and the third line presents the searched procedure parameter values. This table considers the case when $f = f_2$.

	n=64			n=128		
Method	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
TV	.0006 (.0000) .1648 (.0025) 13.0	.0015 (.0000) .2375 (.0046) 7.5	.0023 (.0000) .3563 (.0058) 4.5	.0003 (.0000) .1795 (.0013) 12.0	.0006 (.0000) .3077 (.0022) 6.0	.0010 (.0000) .3504 (.0029) 4.5
AD	.0007 (.0000) .0153 (.0031) .22 , 4	.0038 (.0000) .1111 (.0067) .4 , 4	.0087 (.0001) .1418 (.0066) .75 , 3	.0004 (.0000) .0085 (.0017) .18 , 7	.0022 (.0000) .1709 (.0041) .4 , 5	.0052 (.0000) .1880 (.0033) .72 , 4
ONLM	.0006 (.0000) .1264 (.0029) 3 , 2	.0014 (.0000) .2146 (.0048) 4 , 2	.0024 (.0000) .2529 (.0067) 5 , 2	.0003 (.0000) .1795 (.0017) 5 , 3	.0006 (.0000) .2308 (.0031) 5 , 3	.0009 (.0000) .2564 (.0045) 6 , 3
MED	.0011 (.0000) .1918 (.0028) .0313	.0026 (.0000) .3793 (.0042) .0406	.0038 (.0000) .5019 (.0049) .0500	.0006 (.0000) .2735 (.0015) .0203	.0014 (.0000) .4530 (.0021) .0250	.0022 (.0000) .5726 (.0020) .0297
NEW	.0006 (.0000) .0625 (.0035) .023 , 9 , .031	.0013 (.0000) .1016 (.0066) .023 , 16 , .038	.0019 (.0000) .2007 (.0099) .028 , 18 , .053	.0002 (.0000) .0477 (.0022) .012 , 20 , .019	.0006 (.0000) .0873 (.0045) .014 , 22 , .028	.0008 (.0000) .1568 (.0065) .014 , 34 , .033

images in this figure, we can see that the method NEW is indeed better in preserving edges than the four competing methods, which is consistent with the results of EP shown in Table 2.

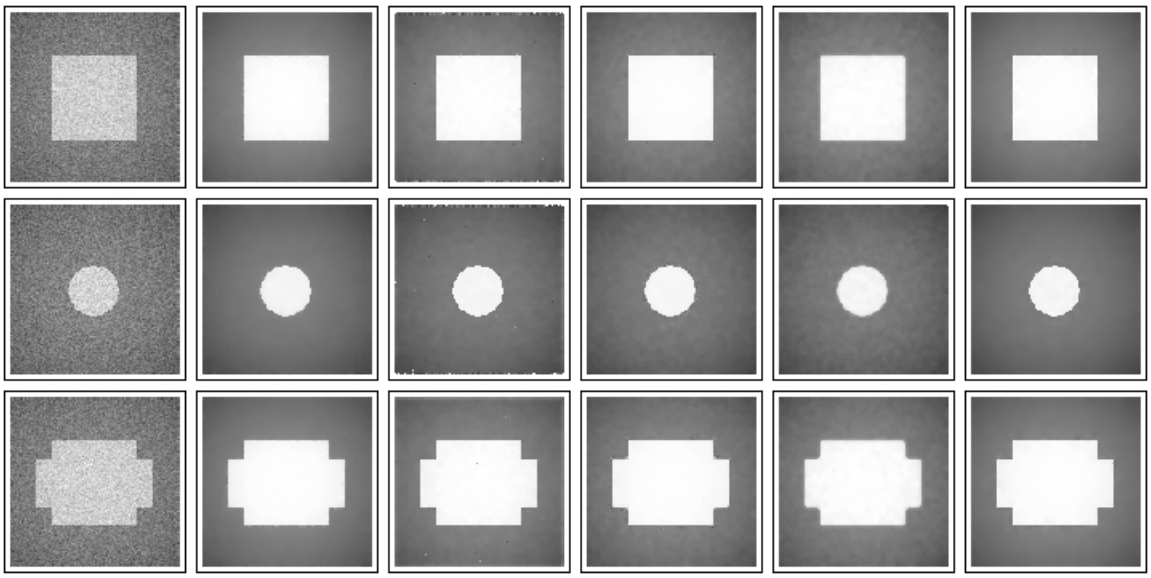

Figure 4: The first column presents three cross sections of an image generated from model (1) with $f = f_1$, $n = 128$, $\varepsilon_{111} \sim N(0, \sigma^2)$, and $\sigma = 0.2$. Columns 2–6 show denoised images by TV, AD, ONLM, MED and NEW, respectively, when their procedure parameters are chosen to be the corresponding ones listed in Table 1.

Next, we consider a magnetic resonance image (MRI) of a human brain with $128 \times 128 \times 52$ pixels. Its image intensity levels range from 0 to 809. I.i.d. noise from the distribution $N(0, \sigma^2)$ is added to the image, where σ is chosen to be 40, 70, or 100. The parameters of the four denoising procedures are chosen in the same way as those in the examples of Tables 1 and 2. The results are shown in left-side portion of Table 3 and in Figure 7. From Table 3, we can see that the proposed procedure NEW outperforms all three competing methods in all cases in terms of both the MISE and the edge preservation measure EP. In Figure 7, the first two rows present two cross-sections of a noisy version of the 3-D MRI image when $\sigma = 40$ (1st column), and their denoised versions by procedures TV, AD, ONLM, MED, and NEW (columns 2–6). The third and fourth rows and the fifth and sixth rows present the corresponding results when $\sigma = 70$ and 100, respectively. In order to see the detail better, in the seventh row, we zoom out the upper-middle portion of the images in

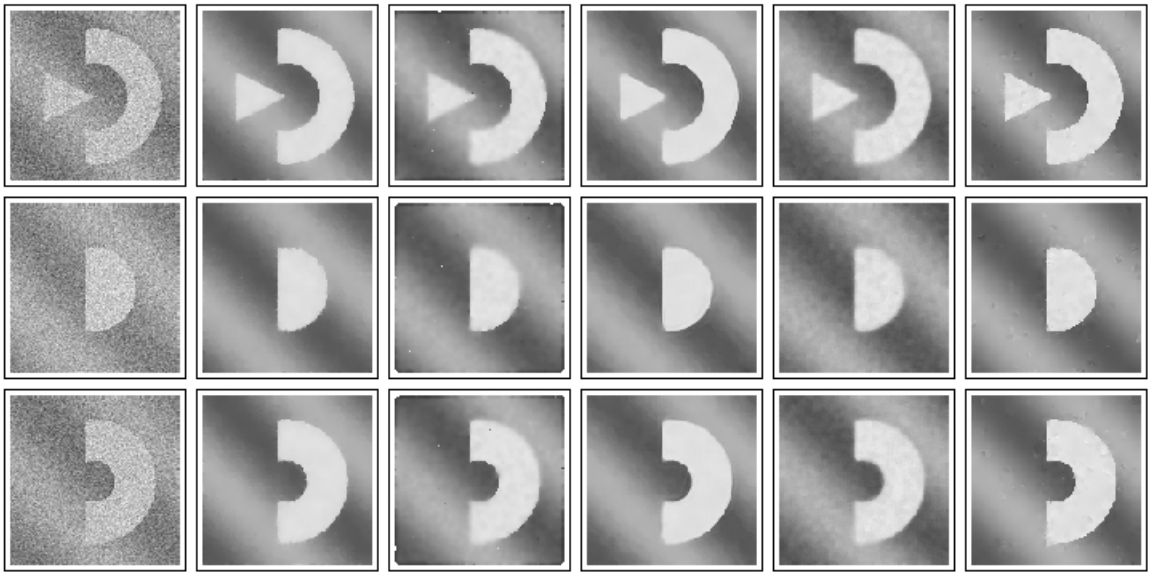

Figure 5: The first column presents three cross sections of an image generated from model (1) with $f = f_2$, $n = 128$, $\varepsilon_{111} \sim N(0, \sigma^2)$, and $\sigma = 0.2$. Columns 2–6 show denoised images by TV, AD, ONLM, MED and NEW, respectively, when their procedure parameters are chosen to be the corresponding ones listed in Table 2.

the fourth row. From the images in this figure, it seems that procedure NEW preserves edges better than the competing methods, which is consistent with results in Table 3.

We also consider the case when the noise level is different at different places of the image. In practice, the noise level is often higher in the foreground (i.e., the central part of the image with the image objects) and lower in the background. To mimic this situation, the following variable noise level is considered:

$$\sigma(x, y, z) = 25 + 50 \exp\left[-\frac{(x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2}{4}\right].$$

The denoising methods are executed in the same way as before, and the results are presented in the right-side portion of Table 3 and in Figure 8. From the table and the figure, we can see that the proposed procedure NEW performs relatively well in this case too.

In Section 2.4, a cross-validation (CV) procedure is proposed for choosing the parameters of the proposed procedure NEW. Next, we apply NEW to all examples discussed above in the same as described, except that u_n is chosen by (8) with $\alpha_n = 0.001$ and h_n^* and h_n are chosen by minimizing

Table 3: In each entry, the first line presents the estimated MISE value from 100 simulations and the corresponding standard error of ISE (in parenthesis), the second line presents the measure of edge-preservation EP and the corresponding standard error, and the third line presents the searched procedure parameter values.

Method	Gaussian Noise $\sigma = 40$	Gaussian Noise $\sigma = 70$	Gaussian Noise $\sigma = 100$	Variable Noise variable σ
TV	374.0 (1.0) .3021 (.0015) .05	737.5 (2.2) .4788 (.0020) .025	1063.6 (3.4) .5350 (.0025) .018	756.4 (2.5) .4647 (.0021) .025
AD	456.2 (1.3) .2574 (.0016) 110 , 2	971.8 (3.0) .4042 (.0021) 190 , 2	1525.0 (4.8) .4440 (.0018) 700 , 1	997.5 (3.1) .4223 (.0022) 200 , 2
ONLM	403.8 (1.2) .2264 (.0013) 6 , 1	764.5 (2.8) .3158 (.0019) 10 , 1	1129.1 (4.6) .3494 (.0024) 16 , 1	785.2 (2.8) .3289 (.0019) 11 , 1
MED	642.7 (1.6) .2120 (.0013) .0078	1065.9 (2.7) .4241 (.0018) .0117	1472.1 (4.1) .4727 (.0023) .0141	1093.9 (2.8) .4200 (.0019) .0117
NEW	371.3 (1.1) .1899 (.0017) .012,8000,.012	669.7 (2.2) .2529 (.0026) .012,12000,.014	949.5 (3.9) .3128 (.0038) .012,16000,.017	700.0 (2.5) .2574 (.0029) .012,12500,.014

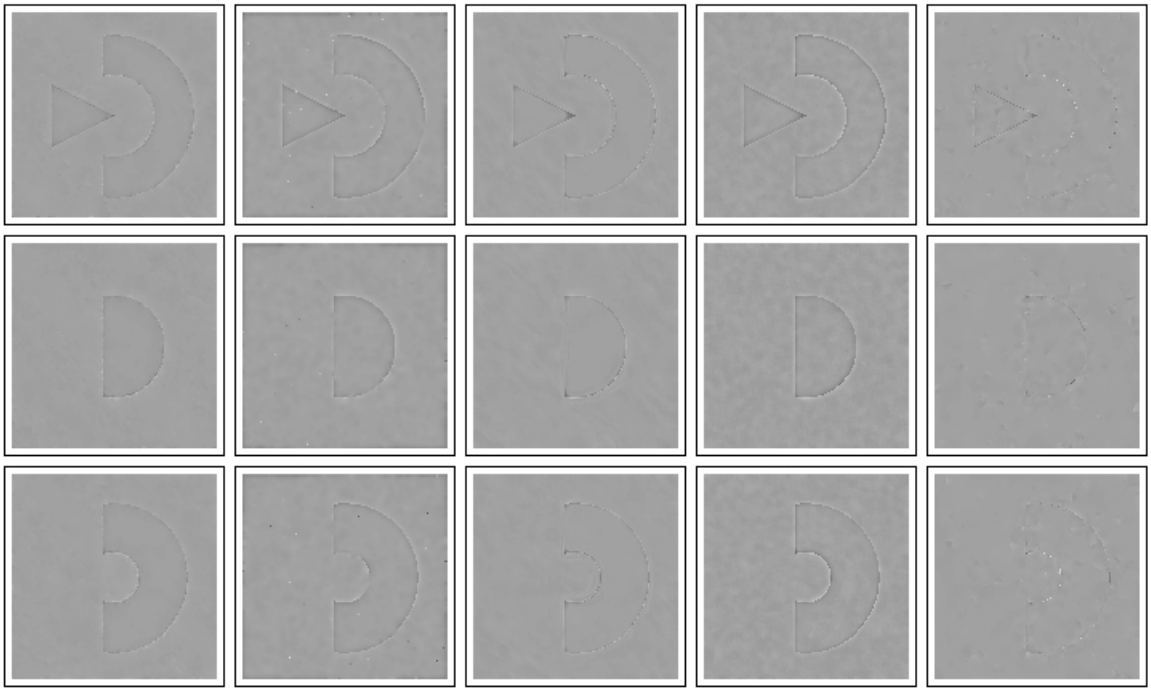

Figure 6: The five columns show the differences between the three slices of the denoised images shown in Figure 5 and the corresponding slices of the true image, for methods TV, AD, ONLM, MED and NEW, respectively. Pixels at which the estimation is greater than (less than) the true intensity value are darker (lighter), and the majority of errors are small (gray).

the CV score defined in (9). The results based on 100 replications are presented in Table 4. From this table and Tables 1–3, we can see that (i) results of NEW with its parameters selected by CV are close to results of NEW with its parameters selected optimally by minimizing the estimated MISE in all cases, and (ii) when its parameters chosen by CV, NEW still outperforms TV and MED in all cases, outperforms AD when $\sigma = 0.2$ or 0.3, and is comparable with AD when $\sigma = 0.1$. The three cross sections of its denoised image in the case of Figure 4 are shown in the first column of Figure 9, the three cross sections of its denoised image in the case of Figure 5 are shown in the second column, and the cross sections of its denoised images corresponding to rows 2, 4 and 6 of Figure 7 are shown in the third column. Comparing Figure 9 with Figures 4, 5, and 7, we can see that the denoised images of NEW when its parameters are chosen by CV are indeed similar to its denoised images when its parameters are chosen optimally by minimizing the estimated MISE.

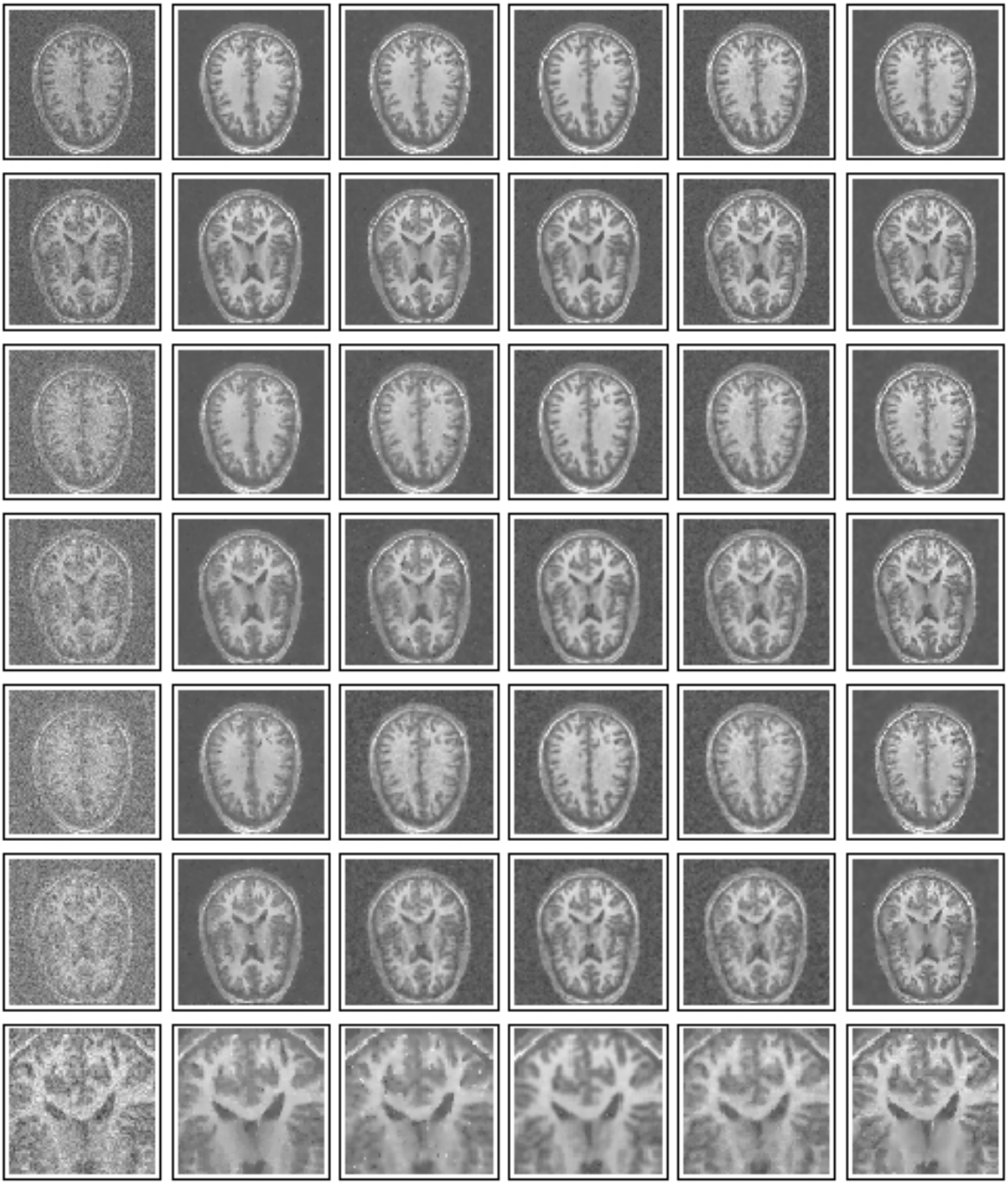

Figure 7: The first two rows present two cross-sections of a noisy version of the 3-D MRI image when $\sigma = 40$ (1st column), and their denoised versions by procedures TV, AD, ONLM, MED, and NEW (columns 2–6). The third and fourth rows and the fifth and sixth rows present the corresponding results when $\sigma = 70$ and 100, respectively. Images in the seventh row zoom out the upper-middle portion of the images in the fourth row. Procedure parameters are chosen to be the ones listed in Table 3.

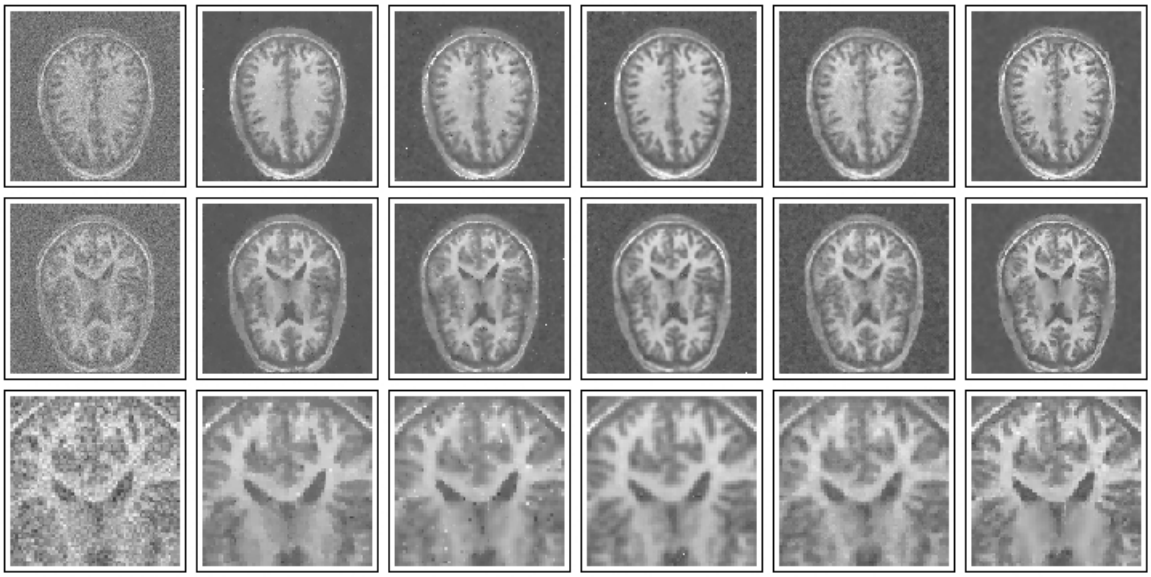

Figure 8: The first two rows present two cross-sections of a noisy version of the 3-D MRI image when $\sigma = 25 + 50\exp\{-[(x-0.5)^2+(y-0.5)^2+(z-0.5)^2]/4\}$ (1st column), and their denoised versions by procedures TV, AD, ONLM, MED, and NEW (columns 2–6). Images in the third row zoom out the upper-middle portion of the images in the second row. Procedure parameters are chosen to be the ones listed in Table 3.

5 Concluding Remarks

We have presented an image denoising procedure in the framework of jump regression analysis. The new procedure can preserve edges and major edge structures. Numerical examples show that it performs well in various cases. However, 3-D image denoising is a challenging task. From the construction of the proposed method, it can be seen that this method would not work well at places where two or more edge surfaces cross or at other singular points of the JLSs. It would not efficiently preserve edge structures that are more complicated than the ones shown in Figure 2 either, although such structures might be less important in describing image objects. The current denoising procedure requires explicit detection of JLSs before estimating the true image intensity function. In practice, it might be more convenient to use a 3-D denoising procedure without explicit detection of the JLSs. All these issues need to be addressed in our future research.

Table 4: Simulation results of the procedure NEW when its parameters are chosen by (8) with $\alpha_n = 0.001$ and by minimizing the CV score defined in (9).

	n=64			n=128		
	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
f_1	.0003 (.0000)	.0008 (.0000)	.0013 (.0001)	.0002 (.0000)	.0004 (.0000)	.0006 (.0000)
	-.0096 (.0028)	-.0136 (.0055)	.0132 (.0073)	.0129 (.0016)	.0124 (.0027)	.0161 (.0037)
	.034 , .041	.034 , .038	.034 , .056	.014 , .027	.014 , .027	.014 , .028
f_2	.0006 (.0000)	.0013 (.0000)	.0020 (.0000)	.0002 (.0000)	.0007 (.0000)	.0008 (.0000)
	.0650 (.0036)	.1332 (.0061)	.1956 (.0103)	.0479 (.0024)	.0812 (.0049)	.1723 (.0065)
	.023 , .034	.031 , .047	.031 , .053	.016 , .020	.012 , .025	.017 , .031
	$\sigma = 40$	$\sigma = 70$	$\sigma = 100$	variable σ		
MRI	435.9 (1.3)	729.0 (2.2)	1024.6 (4.4)	763.6 (2.6)		
	.1752 (.0018)	.1921 (.0031)	.2727 (.0039)	.1877 (.0033)		
	.012 , .014	.012 , .014	.012 , .017	.012 , .014		

Supplemental Materials

Proofs of the two theorems in Section 3 and R-packages for executing the proposed method are provided online as supplemental materials. They are briefly described below.

Proofs.pdf: A pdf file containing the proofs of the two theorems in Section 3.

readme.txt: Readme file that explains how to use the R-packages provided in supplemental materials.

denoise-1.0.tar.gz: This R-package denoises 3-D images. It also calculates performance measures MISE and JS when the true image is known. In this package, users can specify the values of the three procedure parameters h_n^*, u_n and h_n.

denoiseautothresh-1.0.tar.gz: This R-package denoises 3-D images. It also calculates performance measures MISE and JS when the true image is known. In this package, u_n is chosen to be the one given in (8) of Section 2.4 with $\alpha_n = 0.001$.

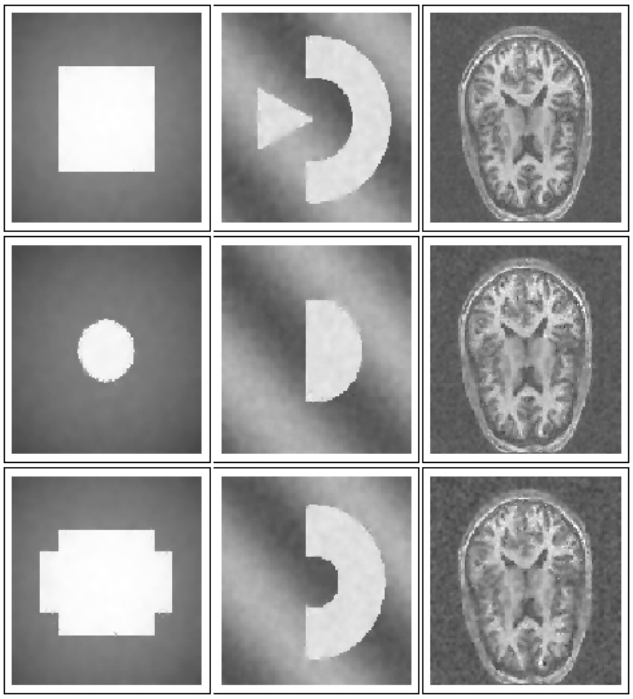

Figure 9: The first column shows the cross sections of the denoised image of NEW in the case of Figure 4, the second column shows the cross sections of the denoised image of NEW in the case of Figure 5, and the third column shows the cross sections of the denoised images of NEW corresponding to rows 2, 4 and 6 of Figure 7. In this example, procedure parameters of NEW are chosen by CV.

denoisecv-1.0.tar.gz: This R-package calculates the CV-score defined by (9) in Section 2.4. It can be used for selecting the parameters h_n^* and h_n by the CV procedure.

References

BARASH, D. (2002). A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **24** 844–847.

BESAG, J. (1986). On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society (Series B)*. **48** 259–302.

BUADES, B., COLL, B. and MOREL, J.M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*. **4(2)** 490–530.

CANNY, J. (1986), A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **8(6)** 679–698.

CHAMBOLLE, A. (2004), An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*. **20(1–2)** 89–97.

CHANG, G.S., YU, B. and VETTERLI, M. (2000). Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Transactions on Image Processing*. **9(9)** 1522–1531.

CHU, C.K., GLAD, I.K., GODTLIEBSEN, F. and MARRON, J.S. (1998). Edge-preserving smoothers for image processing (with discussion). *Journal of the American Statistical Association*. **93** 526–556.

COUPE, P., HELLIER, P., PRIMA, S., KERVRANN, C. and BARILLOT, C. (2008a). 3D wavelet subbands mixing for image denoising. *International Journal of Biomedical Imaging*. **2008** 1–11.

COUPE, P., YGER, P., PRIMA, S., HELLIER, P., KERVRANN, C. and BARILLOT, C. (2008b). An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. *IEEE Transactions on Medical Imaging*. **27** 425–441.

GEMAN, S. and GEMAN, D. (1984). "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6** 721–741.

GETREUER, P. (2007). tvdenoise.m (a MATLAB program). http://www.mathworks.nl/matlabcentral/fileexchange/16236.

GIJBEL, I., LAMBERT, A. and QIU, P. (2006). Edge-preserving image denoising and estimation of discontinuous surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **28(7)** 1075–1087.

GODTLIEBSEN, F. and SEBASTIANI, G. (1994). Statistical methods for noisy images with discontinuities. *Journal of Applied Statistics*. **21** 459–477.

HALL, P. and QIU, P. (2007). Blind deconvolution and deblurring in image analysis. *Statistica Sinica*. **17** 1483–1509.

HALL, P. and ROBINSON, A.P. (2009). Reducing variability of crossvalidation for smoothing-parameter choice. *Biometrika*. **96(1)** 175–186.

HILLEBRAND, M. and MÜLLER, C.H. (2007). Outlier robust corner-preserving methods for reconstructing noisy images. *The Annals of Statistics*. **35** 132–165.

HOSTALKOVA, E., OLDRICH, V. and PROCHAZKA, A. (2007). Multi-dimensional biomedical image de-noising using Haar transform. *Proceedings of the 15th International Conference on Digital Signal Processing*. 175–178, Cardiff, UK.

KEELING, S. (2003). Total variation based convex filters for medical imaging. *Applied Mathematics and Computation*. **139(1)** 101–119.

LOADER, C.R. (1999). Bandwidth selection: classical or plug-in? *Annals of Statistics*. **27(2)** 415–438.

LOPES, D.S. (2007). anisodiff3D.m (a MATLAB program). http://www.mathworks.co.uk/matlabcentral/fileexchange/14995-anisotropic-diffusion-perona-malik.

LU, H., JUI-HSI, C., HAN, G., LI, L. and LIANG, Z. (2001). A 3D distance-weighted Wiener filter for Poisson noise reduction in sinogram space for SPECT imaging. *SPIE proceedings series*. **4320**. 905–913.

MARRON, J.S. (1988). Automatic smoothing parameter selection: A survey. *Empirical Economics*. **13(3-4)** 187–208.

NISHII, R. (1984). Asymptotic properties of criteria for selection of variables in Multiple Regression. *The Annals of Statistics*. **12** 758–765.

PERONA, P. and MALIK, J.(1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **12(7)** 629–639.

POLZEHL, J. and SPOKOINY, V.G. (2000). Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society (Series B)*. **62**, 335–354.

PORTILLA, J., STRELA, V., WAINWRIGHT, M. and SIMONCELLI, E.P. (2003). Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*. **12(11)** 1338–1351.

QIU, P. (1998). Discontinuous regression surfaces fitting. *The Annals of Statistics*. **26** 2218–2245.

QIU, P. (2007). Jump surface estimation, edge detection, and image restoration. *Journal of the American Statistical Association*. **102** 745–756.

QIU, P. and MUKHERJEE, P.S. (2010). Edge structure preserving image denoising. *Signal Processing*. **90** 2851–2862.

QIU, P. and YANDELL, B. (1997). Jump detection in regression surfaces. *Journal of Computational and Graphical Statistics*. **6(3)** 332–354.

RUDIN, L., OSHER, S. and FATEMI, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*. **60** 259–268.

SAINT-MARC, P., CHEN, J. and MEDIONI, G. (1991). Adaptive smoothing: a general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **13(6)** 514–529.

SONKA, M., HLAVAC, V. and BOYLE, R. (2008), *Image Processing, Analysis, and Machine Vision (3rd ed.)*, Toronto, Thomson Learning.

SUN, T., GABBOUJ, M. and NEUVO, Y. (1994). Center weighted median filters: some properties and their applications in image processing. *Signal Processing*. **35(3)** 213–229.

SUN, J. and QIU, P. (2007). Jump detection in regression surfaces using both first-order and second-order derivatives. *Journal of Computational and Graphical Statistics*. **16(2)** 289–311.

TAKEDA, H., FARSIU, S. and MILANFAR, P. (2007). Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*. **16(2)** 349–366.

TAKEDA, H. and MILANFAR, P. (2009). An adaptive nonparametric approach to restoration and interpolation for medical imaging. *Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI): From Nano to Macro*. Boston, MA.

TOMASI, C. and MANDUCHI, R. (1998). Bilateral filtering for gray and color images. *Proceedings of the 1998 IEEE International Conference on Computer Vision*. 839–846. Bombay, India.

WANG, Y. and ZHOU, H. (2006). Total variation wavelet-based medical image denoising. *International Journal of Biomedical Imaging*. **2006** 1–6.

WEICKERT, J., ROMENI, B.M. TER HAAR and VIERGEVER, M.A. (1998). Efficient and reliable schemes for nonlinear filtering. *IEEE Transactions on Image Processing*. **7(3)** 398–410.

WOISELLE, A., STARCK, J.L. and FADILI, J. (2008). 3D image restoration with the curvelet transform. *2008 Proceedings of Astronomical Data Analysis V*. Heraklion (Crete).