Edge Structure Preserving Image Denoising

Peihua Qiu and Partha Sarathi Mukherjee

School of Statistics

University of Minnesota

Minneapolis, MN 55455

Abstract

Image denoising is important in image analysis. It is often used for pre-processing images so that subsequent image analysis is more reliable. Besides noise removal, one important requirement for image denoising procedures is that they should preserve true image structures, such as edges. This paper proposes a novel denoising procedure which can preserve edges and major edge features (e.g., angles of the edges). Our method is based on nonparametric estimation of a discontinuous surface from noisy data, in the framework of jump regression analysis, because a monochrome image can be regarded as a surface of the image intensity function and such a surface has discontinuities at the outlines of objects. Numerical studies show that this method works well in applications, compared to some existing image denoising procedures.

Key Words: Angles, curvature, edges, jump-preserving surface estimation, local smoothing, nonparametric regression, surface estimation.

1 Introduction

Image denoising is often used for pre-processing images so that subsequent image analysis is more reliable (Gonzalez and Woods 1992). Besides noise removal ability, another important requirement for image denoising procedures is that true image structures, such as edges, should be preserved in the denoising process. In this paper, we handle the image denoising problem in the framework of jump regression analysis (JRA), which is a research area handling regression models involving jumps and discontinuities (Qiu 2005). In this framework, image denoising can be accomplished by estimating a discontinuous surface from noisy data, because a monochrome image can be regarded as a surface of the image intensity function and such a surface has discontinuities at the outlines of objects. A novel procedure is suggested in this paper for estimating discontinuous surfaces from noisy data, which can preserve edges and major edge features (e.g., angles of the edges).

In the literature, there are some existing procedures for image denoising and restoration. One group of methods are based on Bayesian estimation, using Markov random field (MRF) modeling and maximum *a posteriori* (MAP) algorithms (e.g., Besag 1974, 1986, Fessler et al. 2000, Geman and Geman 1984, Godtliebsen and Sebastiani 1994, Marroquin et al. 2001, Moussouris 1974). Some closely related methods use the regularization approach, by minimizing certain objective function that enforces a roughness penalty in addition to a term measuring fidelity of an estimator to the data (e.g., Li 1995, Rivera and Marroquin 2002). Image denoising by median filtering and robust estimation is a popular pre-smoothing tool in image processing, because it has certain ability of preserving edges when removing noise (e.g., Brownrigg 1984, Gallagher and Wise 1981, Hillebrand and Müller 2007, Sun et al. 1994). Other image restoration procedures include adaptive smoothing filters (e.g., Polzehl and Spokoiny 2000, Saint-Marc et al. 1991), bilateral filtering procedures (e.g., Chu et al. 1998, Tomasi and Manduchi 1998), diffusion filtering procedures (e.g., Barash 2002, Perona and Malik 1990), wavelet transformation procedures (e.g., Chan et al. 2000, Figueiredo and Nowak 2001, Nason and Silverman 1994, Portilla et al. 2003), discontinuity-preserving surface estimation procedures (e.g., Gijbels et al. 2006, Qiu 1998, 2004, 2009, Sinha and Schunck 1992, Yi and Chelberg 1995), among some others. See Qiu (2007) for a more detailed discussion on this topic.

Most image denoising and jump surface estimation procedures mentioned above have ability in preserving edges at places where the edge curvature is not large. At places where the edges have angles or where their curvature is large, however, such edges are often blurred or rounded by these existing methods (cf., some numerical examples in Section

3). One major reason why this would happen is that the edge structures (e.g., angles) are hidden in observed image intensities, they are not easy to describe and measure (cf., Chabat et al. 1999, Yang et al. 1996), and they are even more difficult to accommodate in the image denoising process (cf., Gijbels et al. 2006). In our opinion, edge structures are an important part of images, because they often denote major characteristics of image objects, and are easier to capture our visual attention than the parts of the edges with relatively small curvature. Therefore, they should be preserved during image denoising. In other words, a good image denoising procedure should preserve not only the parts of the edges with small curvature but also certain major edge structures, such as angles, corners, and other places on the edges with large curvature, although the latter goal is much more challenging than the former.

In this paper, an image denoising procedure is suggested, which can preserve edges and major edge structures well. Our method is based on JRA, and consists of three major steps, outlined below. First, edge pixels are detected in the whole design space by an edge detector. Second, in a neighborhood of a given pixel, a piecewise linear curve is estimated from the detected edge pixels by a simple but efficient algorithm, to approximate the underlying edge segment in that neighborhood. Finally, observed image intensities on the same side of the estimated edge segment, as the given pixel, are averaged by the local linear kernel smoothing procedure (cf., Fan and Gijbels 1996), for estimating the true image intensity at the given pixel. This proposed image denoising procedure is described in detail in Section 2. Some numerical examples are presented in Section 3, for evaluating its numerical performance, in comparison with several existing denoising procedures. Some remarks conclude the article in Section 4.

2 Methodology

We present our proposed methodology in three parts. In Section 2.1, 2-D local quadratic kernel (LQK) smoothing and a corresponding edge detection procedure is introduced. Local approximation to edge segments and edge-structure-preserving local denoising are described in Section 2.2. Data driven parameter selection is discussed in Section 2.3.

2.1 Edge detection by LQK smoothing

As discussed in Section 1, the first step of the proposed image denoising procedure is to detect edge pixels using an edge detector. Theoretically speaking, any reasonable edge detector can be used here. In the literature, most existing edge detectors are based on estimation of the first-order derivatives (e.g., Canny 1986, Fleck 1992, Qiu 2002, Qiu and Bhandarkar 1996) or the second-order derivatives (e.g., Clark 1989, Torre and Poggio 1986) of the image intensity function. Recently, Sun and Qiu (2007) propose an edge detector that combines the major strengths of the two types of edge detectors, by using both the first-order and the second-order derivatives of the image intensity function. This edge detector will be used in all numerical examples of this paper, and it is briefly described below.

Assume that observed image intensities $\{Z_{ij}, i, j = 1, 2, \ldots, n\}$ follow the following 2-D regression model:

$$Z_{ij} = f(x_i, y_j) + \varepsilon_{ij}, \text{ for } i, j = 1, 2, \ldots, n, \qquad (1)$$

where $\{(x_i, y_j), i, j = 1, 2, \ldots, n\}$ are equally spaced pixel locations, f is the unknown image intensity function, and $\{\varepsilon_{ij}, i, j = 1, 2, \ldots, n\}$ are independent and identically distributed (i.i.d.) random errors with mean 0 and unknown variance σ^2. At a given pixel (x, y), let us consider a circular neighborhood $O^*(x, y) = \{(u, v) : \sqrt{(u-x)^2 + (v-y)^2} \leq h_n^*\}$, where $h_n^* > 0$ is a bandwidth parameter. Then, LQK smoothing is accomplished by

$$\min_{a,b,c,d,e,f} \sum_{(x_i,y_j)\in O^*(x,y)} \Big\{ Z_{ij} - \big[a + b(x_i - x) + c(y_j - y) + d(x_i - x)(y_j - y) + e(x_i - x)^2 + f(y_j - y)^2 \big] \Big\}^2 K\left(\frac{x_i - x}{h_n^*}, \frac{y_j - y}{h_n^*}\right), \qquad (2)$$

where K is a radially symmetric, bivariate density kernel function with support $\{(x, y) : x^2 + y^2 \leq 1\}$. The solution to a of the minimization problem (2) can be used as an estimator of the intensity $f(x,y)$, the solution to $(b,c)'$ as an estimator of the gradient vector $G(x,y) = (f_x(x,y), f_y(x,y))'$, and the solution to $(e,f)'$ as an estimator of $(f_{xx}(x,y), f_{yy}(x,y))'$. These estimators, denoted as $\widehat{f}(x,y)$, $\widehat{f_x}(x,y)$, $\widehat{f_y}(x,y)$, $\widehat{f_{xx}}(x,y)$, and $\widehat{f_{yy}}(x,y)$, are called LQK estimators in the literature (e.g., Fan and Gijbels 1996). Along the estimated gradient direction $\widehat{G}(x,y) = (\widehat{f_x}(x,y), \widehat{f_y}(x,y))'$, if (x,y) is on an edge segment, then $\widehat{G}(x,y)$ would have large Euclidean length and $(\widehat{f_{xx}}(x,y), \widehat{f_{yy}}(x,y))'$ would have the *zero-crossing* properties that they are zero at (x,y) and change signs on two different sides of the edge segment. See Figure 1 for a demonstration in one-dimensional cases. Then, a point (x,y) is flagged as a detected edge pixel if $\sqrt{\widehat{f_x}^2(x,y) + \widehat{f_y}^2(x,y)}$ is larger than a threshold value u_n and $\widehat{f_{xx}} + \widehat{f_{yy}}$ demonstrates the zero-crossing properties in $O^*(x,y)$. A formula for the threshold value is derived in Sun and Qiu (2007), which depends on a significance level α_n. In all our numerical examples presented in Section 3, α_n is fixed at 0.01.

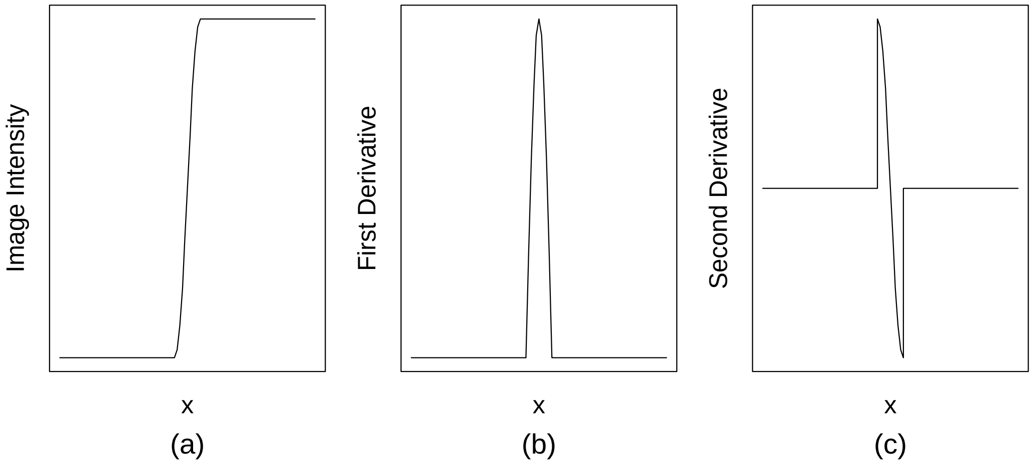

Figure 1: (a): 1-D profile of the image intensity surface around an edge segment. (b): First-order derivative of the 1-D profile. (c): Second-order derivative of the 1-D profile.

2.2 Edge structure preserving image denoising

Detected edge pixels are identified after the edge detection step discussed in the previous part. In this part, we describe the remaining two steps of the proposed image denoising procedure. At a given pixel (x, y), we consider its circular neighborhood

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

where $h_n > 0$ is a bandwidth parameter which could be different from h_n^* used in (2). Detected edge pixels in $O(x, y)$ are denoted by $\{(w_k, v_k), k = 1, 2, \ldots, m\}$. Our major goal here is to estimate $f(x, y)$ from observations in $O(x, y)$ with possible edges preserved.

From (2), the estimated gradient at a detected edge pixel (w_k, v_k) is $\widehat{G}(w_k, v_k)$, for $k = 1, 2, \ldots, m$. Intuitively, if the underlying edge curve in $O(x, y)$ contains an angle, then gradients of f along its two rays would point to two different directions. Therefore, in such cases, we would expect that the estimated gradients $\{\widehat{G}(w_k, v_k), k = 1, 2, \ldots, m\}$ can be divided into two groups, each group corresponds to a ray of the angle, and the estimated gradients in each group would vary around the perpendicular direction of the corresponding ray. See Figure 2 for a demonstration.

To estimate the edge curve in $O(x, y)$, we propose an algorithm with the following steps.

1. Compute the simple average of $\{\widehat{G}(w_k, v_k), k = 1, 2, \ldots, m\}$, denoted as $\overline{G}(x, y)$.

2. Divide $\{(w_k, v_k), k = 1, 2, \ldots, m\}$ into two groups $E = \{(w_k, v_k) : \theta(\widehat{G}(w_k, v_k), \overline{G}(x, y)) \leq 0\}$ and $E^c = \{(w_k, v_k) : \theta(\widehat{G}(w_k, v_k), \overline{G}(x, y)) > 0\}$, where $\theta(\vec{u}, \vec{v}) \in [-\pi/2, \pi/2]$ denotes the angle from vector $\vec{u}$ to vector $\vec{v}$.

3. Compute the line L that goes through the center of E in the perpendicular direction of $\overline{G}_E(x, y)$, where $\overline{G}_E(x, y)$ is the average of the estimated gradients in E.

4. Compute the line L^c that goes through the center of E^c in the perpendicular direction of $\overline{G}_{E^c}(x, y)$, where $\overline{G}_{E^c}(x, y)$ is the average of the estimated gradients in E^c.

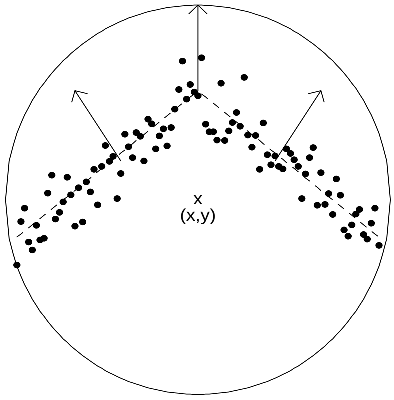

Figure 2: The small dots denote detected edge pixels in $O(x, y)$, and the dashed line denotes the true edge curve which contains an angle. The vertical arrow in the middle denotes $\overline{G}(x, y)$, and the two arrows on its two sides denote two typical estimated gradients of f along the two rays of the angle.

5. Assume that L and L^c cross at point A. If A is located outside of $O(x, y)$, then the two line segments of L and L^c inside $O(x, y)$ are used for estimating the edge segments in $O(x, y)$. In the case when A is located inside of $O(x, y)$, L and L^c is each divided into two half lines by A, and $O(x, y)$ is divided into four parts by the half lines. The two half lines, one on each of L and L^c, that contain the centers of E and E^c, respectively, are selected for estimating the edge curve in $O(x, y)$. Obviously, the two selected half lines form an angle with vertex A.

The estimated edge segment(s) by the above algorithm divides $O(x, y)$ into two or three parts, depending on whether or not the point A defined in step 5 is located inside $O(x, y)$. The part containing the given pixel (x, y) is denoted as $B(x, y)$. Then, the estimator of $f(x, y)$, denoted as $\widehat{f}(x, y)$, can be defined by the solution to a of the following minimization problem:

$$\min_{a,b,c \in R} \sum_{(x_i, y_j) \in B(x,y)} \{Z_{ij} - [a + b(x_i - x) + c(y_j - y)]\}^2 \, K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right). \qquad (3)$$

Obviously, $\widehat{f}(x, y)$ is the local linear kernel (LLK) estimator of $f(x, y)$, constructed from observations in the one-sided neighborhood $B(x, y)$.

Next, we propose a modification to improve the performance of the proposed denoising procedure (3) and to simplify its computation as well. In regions where the true image intensity function f is continuous, it is desirable to use a larger bandwidth to construct an estimator of f, compared to the bandwidth used around true edges, so that the estimator is better in removing noise. To this end, at a given pixel (x, y), we first consider a neighborhood $\tilde{O}(x, y) = \{(u, v) : \sqrt{(u-x)^2 + (v-y)^2} \leq \tilde{h}_n\}$ with a larger bandwidth $\tilde{h}_n$ $(> h_n)$. In $\tilde{O}(x, y)$, if the number of detected edge pixels is so small that it is unlikely to have a true edge segment in it, then we do not implement the last two steps of the proposed denoising procedure described above. In such cases, we can simply define $\widehat{f}(x, y)$ to be the conventional LLK estimator in $\tilde{O}(x, y)$, which is the solution to a of the minimization problem (3), after $B(x, y)$ is replaced by $\tilde{O}(x, y)$. To do so, there are at least two major benefits. One is that $\widehat{f}(x, y)$ is defined using all pixels in $\tilde{O}(x, y)$ in such cases; its denoising ability is thus greatly improved, compared to the estimator constructed in $B(x, y)$. The second benefit is that computation involved is greatly simplified. For a typical observed image, there are many pixels at which no edge segments exist in their neighborhoods. Therefore, the above two benefits are substantial. In all numerical examples presented in next section, if the number of detected edge pixels is smaller than or equal to $[n\tilde{h}_n]$, where $[x]$ denotes the integer part of x, then we define $\widehat{f}(x, y)$ to be the conventional LLK estimator in $\tilde{O}(x, y)$.

When the number of detected edge pixels is larger than $[n\tilde{h}_n]$ in $\tilde{O}(x, y)$, the chance is high that there is a true edge segment in the neighborhood. In such cases, we consider using a smaller bandwidth h_n $(< \tilde{h}_n)$ to deal with the potential edges. In neighborhood $O(x, y)$ with bandwidth h_n, if the number of detected edge pixels is smaller than or equal to $[nh_n]$, then $\widehat{f}(x, y)$ is defined by the conventional LLK estimator in $O(x, y)$. Otherwise, $\widehat{f}(x, y)$ is defined by procedure (3). The entire proposed image denoising procedure can now be summarized as follows.

Proposed Image Denoising Procedure

- Detect edge pixels using an edge detector such as the one described in Section 2.1.

- For a given pixel (x, y), count the number of detected edge pixels in neighborhood $\tilde{O}(x, y)$. If this number is smaller than or equal to $[n\tilde{h}_n]$, then define $\widehat{f}(x, y)$ to be the conventional LLK estimator in $\tilde{O}(x, y)$, and continue the denoising procedure for the next pixel. If this number is larger than $[n\tilde{h}_n]$, then consider a smaller neighborhood $O(x, y)$ and count the number of detected edge pixels in that neighborhood. If the number is smaller than or equal to $[nh_n]$, then $\widehat{f}(x, y)$ is defined to be the conventional LLK estimator in $O(x, y)$. Otherwise, estimate the edge segment in $O(x, y)$ using the algorithm described three paragraphs above, and proceed to the next step.
- Compute $\widehat{f}(x, y)$ using (3), and continue the denoising procedure for the next pixel.

2.3 Selection of procedure parameters

In the proposed image denoising procedure, there are four parameters u_n, h_n^*, $\tilde{h}_n$ and h_n (cf., expressions (2), (3), and the related discussion). They should be chosen properly because performance of the proposed procedure depends on their values. For instance, if u_n is chosen too large, then some real edge pixels would be missed by the edge detector discussed in Section 2.1. Consequently, some jumps in f would be blurred in the denoising process. Similarly, the bandwidths h_n^*, $\tilde{h}_n$ and h_n also play an important role in image denoising. Theoretically speaking, we can choose these parameters by minimizing the Mean Integrated Squared Error (MISE) of the surface estimator, defined by:

$$MISE(\widehat{f}, f) = E\left[\int_0^1 \int_0^1 \left(\widehat{f}(x, y) - f(x, y)\right)^2 \, dxdy\right], \tag{4}$$

where E denotes the expectation with respect to the probability distribution of $\widehat{f}(x, y)$. In practice, because f is unknown, this method can not be actually used. In this paper, we suggest using a modified version of the conventional cross-validation (CV) procedure. Remember that, in the proposed denoising procedure, neighborhoods of two different sizes (i.e., $\tilde{h}_n$ and h_n) are used. For pixels that are quite far away from true edges, their neighborhoods have width $\tilde{h}_n$. For the other pixels, their neighborhoods have width h_n. Let I

be the set of pixels whose neighborhoods have width h_n, $|I|$ be the number of pixels in I, and

$$CV(u_n, h_n^*, \tilde{h}_n, h_n) = \\ \frac{\lambda}{|I|} \sum_{(x_i,y_j)\in I} \left[Z_{ij} - \widehat{f}_{-i,-j}(x_i,y_j)\right]^2 + \frac{1-\lambda}{n^2-|I|} \sum_{(x_i,y_j)\in I^c} \left[Z_{ij} - \widehat{f}_{-i,-j}(x_i,y_j)\right]^2, \quad (5)$$

where I^c denotes the complementary set of I, $\widehat{f}_{-i,-j}(x_i, y_j)$ is the "leave-one-out" estimator of $f(x_i, y_j)$ obtained by (3) when the (i, j)th observation is not used (cf., Section 2.4.4 in Qiu (2005)), and λ is a weighting parameter. It should be pointed out that both I and I^c depend on u_n, h_n^*, $\tilde{h}_n$ and h_n, although it is not explicit in notation. From (5), we can see that λ controls the trade-off between edge-preservation and noise removal when choosing the parameters. In the extreme case when $\lambda = 1$, the second term on the right hand side of (5) would disappear. In such cases, the parameters are chosen to best preserve the edges, and the performance of the denoised image in continuity regions of f are actually not taken into account. In the other extreme case when $\lambda = 0$, the first term on the right hand side of (5) would disappear and the parameters are actually chosen to best remove noise in the continuity regions of f. In practice, one natural choice for λ is $|I|/n^2$, in which case procedure (5) becomes the conventional CV procedure. By (5), selected parameter values are those minimizing $CV(u_n, h_n^*, \tilde{h}_n, h_n)$ under the constraint that $\tilde{h}_n > h_n$. It should be pointed out that, in all our numerical studies presented in next section, we actually did not put the above constraint when searching for the parameter values by CV. From the results (cf., Tables 1–3 in Section 3), we can see that searched values of $\tilde{h}_n$ and h_n satisfy the condition that $\tilde{h}_n > h_n$ in all cases, which implies that this intuitively reasonable constraint is indeed reasonable in practice.

3 Numerical Examples

In this section, we present some numerical results regarding the performance of the proposed image denoising procedure (denoted as NEW), in comparison with several existing image denoising methods. Four existing methods are considered here, including a recent denoising procedure based on gradient estimation and one-sided surface estimation (denoted as GE, Gijbels et al. 2006), a denoising method based on MRF modeling (denoted as MRF, Godtliebsen and Sebastiani 1994), a wavelet transformation method (denoted as WT, Portilla et al. 2003), and the conventional local median filter (denoted as MED).

First, we present some simulation results when the true image intensity function is one of the following three functions:

$$\begin{aligned} f_1(x,y) &= -2(x-0.5)^2 - 2(y-0.5)^2 + \phi(x \geq 0.4)\phi(y \geq 0.3)\phi(2x+y \leq 1.6), \\ f_2(x,y) &= -2(x-0.5)^2 - 2(y-0.5)^2 + \\ &\phi\Big(\phi(y \geq 0.3)\phi(y-\sqrt{3}x \leq 0.8-0.5\sqrt{3})\phi(y+\sqrt{3}x \leq 0.8+0.5\sqrt{3}) + \\ &\phi(y \leq 0.7)\phi(y-\sqrt{3}x \geq 0.2-0.5\sqrt{3})\phi(y+\sqrt{3}x \geq 0.2+0.5\sqrt{3}) > 0\Big), \\ f_3(x,y) &= 0.5(1-x)y + (1-0.5(1-x)y)\phi(y \leq 3(0.25-(x-0.5)^2))\phi((x-0.5)^2+y^2 \geq 0.3) \\ &+(1-0.5(1-x)y)\phi(0.48 \leq x \leq 0.52)\phi(0.25 \leq y \leq 0.5), \end{aligned}$$

where $\phi(a)$ is the indicator function which equals 1 when $a =$"True" and 0 otherwise. One realization of these three functions from model (1) when $n = 256$ and $\sigma = 0.5$ is presented in the first column of Figure 3. From the plots, we can see that edges of f_1 have three angles of different sizes, edges of f_2 have twelve angles, and edges of f_3 have different curvature at different places. In model (1), random errors are generated from distribution $N(0, \sigma^2)$. We consider three σ values 0.25, 0.5, and 0.75, representing low, medium, and high noise levels, and two n values 128 or 256, denoting two different image resolutions.

In method NEW, to save some computation, parameter α_n used for determining the threshold value u_n for edge detection is fixed at 0.01. In such cases, detected edges look reasonably well. Based on our numerical experience, as long as detected edges are reasonably good, their effect on denoised image is minimal. Besides u_n, procedure NEW has another three parameters, h_n^*, $\tilde{h}_n$ and h_n, to choose. The method GE has three parameters: a bandwidth parameter, and two threshold parameters used in surface estimation and corner

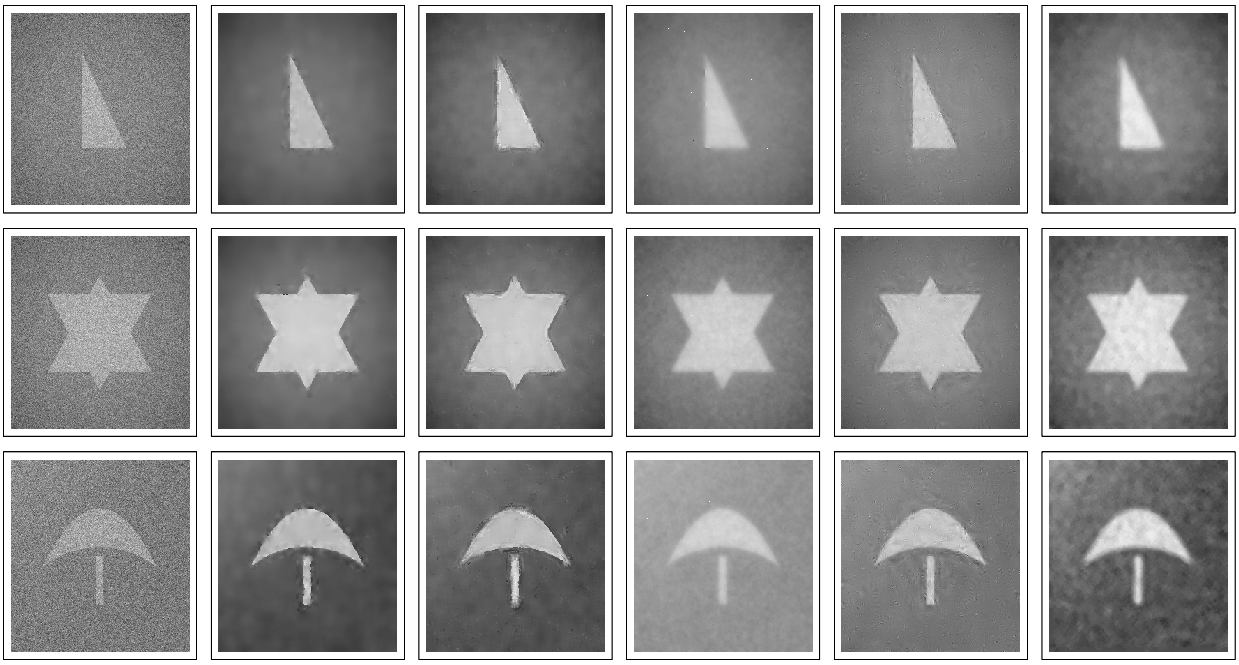

Figure 3: The first column denotes three noisy images when f equals f_1, f_2, or f_3, $n = 256$, and $\sigma = 0.5$. The remaining five columns denote denoised images by the methods NEW, GE, MRF, WT, and MED, respectively, when their procedure parameters are chosen to be the corresponding ones listed in Tables 1–3.

preservation, respectively. The MRF method combines the ideas of using a discontinuity labeling process (Geman and Geman 1984) and the iterated conditional modes algorithm (Besag 1986). This procedure assumes that a binary line component exists between any two vertically or horizontally neighboring pixels, with 1 denoting an existing edge between the two pixels and 0 denoting no edge. In a 3×3 neighborhood of a given pixel, there are 12 line components and 2^{12} possible configurations of these components. To use this procedure, probabilities of the 2^{12} possible line configurations need to be specified. In this section, these probabilities are estimated from the true image intensity function values at the design points, which is in favor of this procedure. Besides the line configurations, it has three positive procedure parameters α, β and λ to determine. In the method WT, the default family of wavelets (which is Daubechies' "extremal phase" wavelet), the "full steerable pyramid" image decomposition procedure, the "Bayes least square (BLS)" solution, and the "symmetric" boundary handling condition are used. Other parameters are chosen to be the ones suggested by Portilla et al. (2003). The method MED defines the surface estimator by the sample median of the observed image intensities in a circular neighborhood of a given pixel. So, it has one parameter (i.e., the bandwidth) to choose. Because it is simple to use and has certain ability in preserving edges while removing noise, it is widely used as a pre-smoothing procedure (cf., Gonzalez and Woods 1992, Chapter 4).

For each denoising method considered, 100 replications are performed in each combination of f, σ, and n. Their parameter values are searched so that the MISE value (cf., expression (4)), estimated by the sample mean of

$$ISE(\widehat{f}, f) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} [\widehat{f}(x_i, y_j) - f(x_i, y_j)]^2$$

over 100 replications, reaches the minimum. The estimated MISE values and the corresponding standard errors of ISE of various methods are presented in Tables 1–3, along with searched parameter values, for cases when f equals f_1, f_2, and f_3, respectively. For investigating ability of various methods in preserving edge structures, their estimated local MISE values, computed in circular neighborhoods of radius 0.1 when $n = 128$ and radius 0.05 when $n = 256$ of edge angles, along with the corresponding standard errors of local ISE are also presented in the tables. From the tables, it can be seen that the proposed method NEW is uniformly better in quite large margins than the remaining methods, in terms of both estimated MISE and estimated local MISE, when noise level is medium to high (i.e., $\sigma = 0.5$ or 0.75). When the noise level is low (i.e., $\sigma = 0.25$), it seems that procedure MRF always performs the best, which is consistent to the findings in Gijbels et al. (2006). For the observed images presented in the first column of Figure 3 when $n = 256$ and $\sigma = 0.5$, the denoised images by various methods when their parameters are chosen to be the ones corresponding to the results presented in Tables 1–3 (cf., columns corresponding to cases when $n = 256$ and $\sigma = 0.5$) are presented in columns 2–6 of Figure 3. From the plots, we can see that certain methods (e.g., GE, MRF, WT) either do not preserve angular edges well, or do not remove noise sufficiently. Method MED blurs all edges to a certain degree when removing noise. As a comparison, the denoised images by the proposed method (cf., the 2nd column in Figure 3) preserve all edges reasonably well when most noise has been removed. As a side note, for the observed triangle image shown in the (1,1)-th panel of Figure 3, the estimated gradients at the detected edge pixels around the lower-left angle by the proposed method are shown in Figure 4. From the figure, it can be seen that the detected edge pixels around that angle can indeed be divided into two groups using the estimated gradients for approximating the true edge curve, as demonstrated in Figure 2.

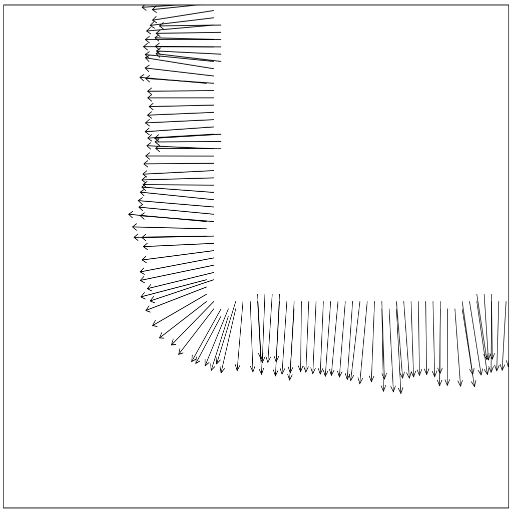

Figure 4: The estimated gradients at the detected edge pixels around the lower-left angle of the triangle image shown in the (1,1)-th panel of Figure 3.

Next, we consider a real test image including a maple leaf in the middle. The image intensities are in the range $[0, 255]$, and the image has 160×160 pixels. A noisy version of this image with i.i.d. noise from $N(0, 100^2)$ is presented in the (1,1)-th plot of Figure 5, from which we can see that the boundary of the maple leaf has a number of quite sharp angles. We then apply the methods NEW, GE, MRF, WT, and MED to this example. Their parameters are searched by minimizing the estimated MISE values obtained from 100 replications, as in Tables 1–3. The estimated MISE values, their standard errors, and the search parameters values are presented in the first column of Table 4, from which we can

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 estimated local MISE value and the corresponding standard error of local ISE computed in circular neighborhoods of true edge angles with width 0.1 when $n = 128$ and width 0.05 when $n = 256$, and the third line presents the searched procedure parameter values. This table considers the case when $f = f_1$.

	n=128			n=256		
Method	$\sigma = .25$	$\sigma = .5$	$\sigma = .75$	$\sigma = .25$	$\sigma = .5$	$\sigma = .75$
NEW	.0029 (.0003) .0118 (.0013) .023,.070,.023	.0043 (.0005) .0172 (.0032) .023,.086,.031	.0061 (.0006) .0246 (.0028) .031,.102,.047	.0016 (.0001) .0128 (.0018) .012,.047,.012	.0025 (.0003) .0202 (.0036) .012,.051,.020	.0034 (.0003) .0303 (.0035) .016,.055,.027
GE	.0022 (.0002) .0108 (.0011) .031, .04, .4	.0080 (.0007) .0311 (.0033) .047, .06, .4	.0134 (.0006) .0475 (.0028) .055, .14, .8	.0011 (.0001) .0201 (.0014) .020, .04, .6	.0043 (.0003) .0355 (.0032) .027, .06, .6	.0081 (.0002) .0494 (.0030) .027, .16, .6
MRF	.0013 (.0002) .0035 (.0015) 1, 40, 1	.0100 (.0004) .0329 (.0013) 15, 5, 15	.0137 (.0008) .0431 (.0025) 15, 4, 20	.0009 (.0001) .0040 (.0019) .5, 35, 1	.0067 (.0003) .0417 (.0039) .5, 10, .5	.0091 (.0004) .0482 (.0034) 10, 5, 10
WT	.0030 (.0001) .0084 (.0007) –	.0098 (.0005) .0213 (.0018) –	.0200 (.0013) .0356 (.0037) –	.0020 (.0001) .0089 (.0001) –	.0069 (.0002) .0214 (.0016) –	.0149 (.0006) .0354 (.0032) –
MED	.0050 (.0002) .0169 (.0010) .023	.0112 (.0005) .0338 (.0019) .039	.0160 (.0008) .0495 (.0025) .055	.0028 (.0001) .0255 (.0010) .020	.0067 (.0002) .0464 (.0019) .027	.0098 (.0003) .0566 (.0028) .031

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 estimated local MISE value and the corresponding standard error of local ISE computed in circular neighborhoods of true edge angles with width 0.1 when $n = 128$ and width 0.05 when $n = 256$, and the third line presents the searched procedure parameter values. This table considers the case when $f = f_2$.

	n=128			n=256		
Method	$\sigma = .25$	$\sigma = .5$	$\sigma = .75$	$\sigma = .25$	$\sigma = .5$	$\sigma = .75$
NEW	.0052 (.0004)	.0085 (.0006)	.0122 (.0008)	.0027 (.0001)	.0043 (.0003)	.0062 (.0003)
	.0131 (.0011)	.0209 (.0015)	.0290 (.0021)	.0137 (.0009)	.0227 (.0013)	.0310 (.0016)
	.023,.063,.023	.023,.086,.039	.031,.094,.055	.012,.043,.012	.012,.109,.023	.016,.055,.027
GE	.0050 (.0002)	.0128 (.0007)	.0197 (.0007)	.0020 (.0001)	.0064 (.0003)	.0118 (.0003)
	.0104 (.0006)	.0280 (.0018)	.0413 (.0014)	.0115 (.0006)	.0257 (.0016)	.0404 (.0013)
	.023, .04, .4	.047, .06, .4	.047, .14, .8	.020, .04, .6	.027, .06, .6	.027, .16, .6
MRF	.0021 (.0003)	.0152 (.0004)	.0221 (.0009)	.0012 (.0001)	.0096 (.0002)	.0131 (.0004)
	.0042 (.0008)	.0349 (.0010)	.0509 (.0019)	.0043 (.0008)	.0358 (.0011)	.0519 (.0015)
	.5, 45, 1	.5, 5, 15	1, 5, 20	.5, 35, 1	2, 5, .5	10, 5, 5
WT	.0048 (.0002)	.0132 (.0006)	.0247 (.0014)	.0029 (.0001)	.0089 (.0002)	.0178 (.0006)
	.0101 (.0004)	.0235 (.0011)	.0382 (.0023)	.0105 (.0004)	.0239 (.0010)	.0387 (.0019)
	–	–	–	–	–	–
MED	.0071 (.0002)	.0161 (.0005)	.0229 (.0009)	.0042 (.0001)	.0097 (.0002)	.0140 (.0004)
	.0156 (.0005)	.0316 (.0012)	.0462 (.0020)	.0189 (.0005)	.0397 (.0010)	.0530 (.0015)
	.023	.031	.047	.016	.023	.027

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 estimated local MISE value and the corresponding standard error of local ISE computed in circular neighborhoods of true edge angles with width 0.1 when $n = 128$ and width 0.05 when $n = 256$, and the third line presents the searched procedure parameter values. This table considers the case when $f = f_3$.

	n=128			n=256		
Method	$\sigma = .25$	$\sigma = .5$	$\sigma = .75$	$\sigma = .25$	$\sigma = .5$	$\sigma = .75$
NEW	.0034 (.0002) .0115 (.0007) .023,.070,.023	.0055 (.0004) .0190 (.0016) .023,.102,.039	.0085 (.0007) .0272 (.0028) .031,.109,.047	.0018 (.0006) .0110 (.0006) .012,.047,.012	.0030 (.0002) .0183 (.0015) .012,.051,.020	.0048 (.0003) .0296 (.0022) .016,.055,.031
GE	.0046 (.0002) .0106 (.0010) .023, .04, .4	.0118 (.0006) .0283 (.0020) .039, .08, .4	.0165 (.0007) .0381 (.0020) .039, .22, .6	.0018 (.0001) .0126 (.0007) .023, .02, .4	.0057 (.0002) .0217 (.0019) .031, .04, .4	.0100 (.0003) .0284 (.0022) .035, .09, .4
MRF	.0020 (.0003) .0050 (.0011) 1, 45, .5	.0122 (.0004) .0325 (.0013) .5, 4, 5	.0165 (.0008) .0421 (.0025) .5, 3, 1	.0014 (.0001) .0048 (.0012) .5, 30, .5	.0077 (.0002) .0339 (.0014) .1, 6, 10	.0105 (.0004) .0425 (.0025) .1, 4, 15
WT	.0042 (.0001) .0098 (.0005) –	.0116 (.0006) .0224 (.0016) –	.0224 (.0013) .0363 (.0034) –	.0026 (.0001) .0094 (.0006) –	.0080 (.0003) .0214 (.0015) –	.0163 (.0006) .0346 (.0029) –
MED	.0062 (.0002) .0185 (.0009) .023	.0135 (.0005) .0347 (.0019) .031	.0190 (.0008) .0518 (.0023) .047	.0037 (.0001) .0192 (.0007) .016	.0080 (.0002) .0367 (.0017) .023	.0113 (.0003) .0470 (.0027) .027

see that the method NEW has the smallest MISE value. The denoised images by these methods with the parameters chosen to be the ones in Table 4 are presented in Figure 5. We can see that method NEW preserves the edge structure and removes the noise well, compared to its peers.

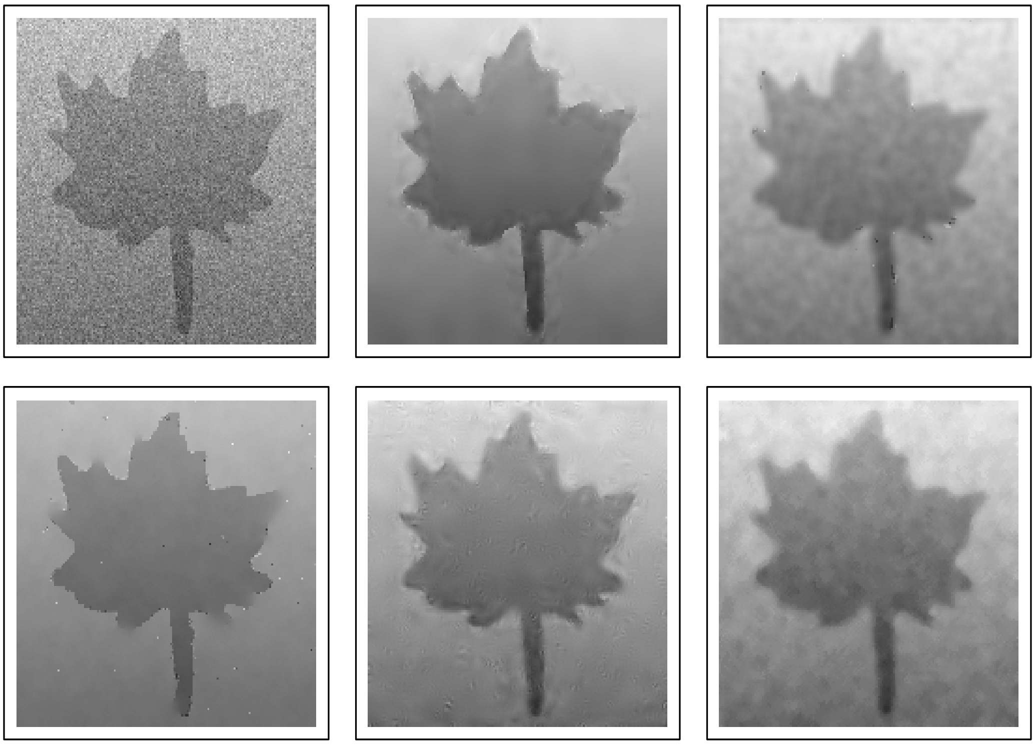

Figure 5: The noisy maple leaf image and its denoised images by methods NEW, GE, MRF, WT, and MED. The noise is from $N(0, 100^2)$.

We next consider a magnetic resonance image (MRI) of a knee part of human body with 128×128 pixels. The image intensity levels range from 0 to 255 as usual. A noisy version with i.i.d. noise from $N(0, 100^2)$ is presented in the (1,1)-th plot of Figure 6. As in the previous example, the searched parameters and the estimated MISE values of the methods NEW, GE, MRF, WT, and MED, based on 100 replications, are presented in the second column of Table 4. From the table, it can be seen that method WT performs the best, and method NEW is better than the remaining three methods. The denoised images by these methods from the one shown in the (1,1)-th plot are presented in the next five

Table 4: The first line in each entry presents the estimated MISE value from 100 simulations and the corresponding standard error of ISE (in parenthesis), and the second line presents the searched procedure parameter values. The five columns are for cases of the leaf image, the knee image with Gaussian noise (GN), the knee image with double exponential noise (DEN), the knee image with variable noise (VN) level, and the knee image with 30% salt-and-pepper noise (SPN), respectively.

Method	Leaf	Knee (GN)	Knee (DEN)	Knee (VN)	Knee (SPN)
NEW	431.52 (13.65) .019,.063,.025	855.90 (21.89) .032,.070,.032	762.40 (19.44) .032,.070,.032	846.07 (29.80) .032,.070,.032	1314.68 (40.38) .032,.078,.032
GE	713.54 (13.47) .019,20000,0.8	941.48 (17.71) .023,30000,1	872.88 (14.04) .023,10000,1	869.67 (16.64) .023,30000,1	3041.21 (80.12) .023,10000,1
MRF	576.51 (38.84) 1,.0002,.16	920.75 (18.23) 1,.0001,20	1469.46 (81.40) .25,.0001,20	1213.01 (111.00) .5,.0001,20	1813.47 (76.29) 1,0.0001,20
WT	586.58 (18.66) –	840.14 (28.88) –	825.54 (39.05) –	1233.66 (104.63) –	1375.81 (52.12) –
MED	783.92 (19.32) .025	1151.27 (21.52) .023	763.72 (14.23) .016	989.26 (19.57) .023	556.82 (19.74) .016

plots of Figure 6, from which we can see that both methods NEW and WT preserve edges well, compared to the remaining three methods. It seems that method NEW preserves edge structures a little better than method WT (cf., the edge segment surrounding the central white part in the image).

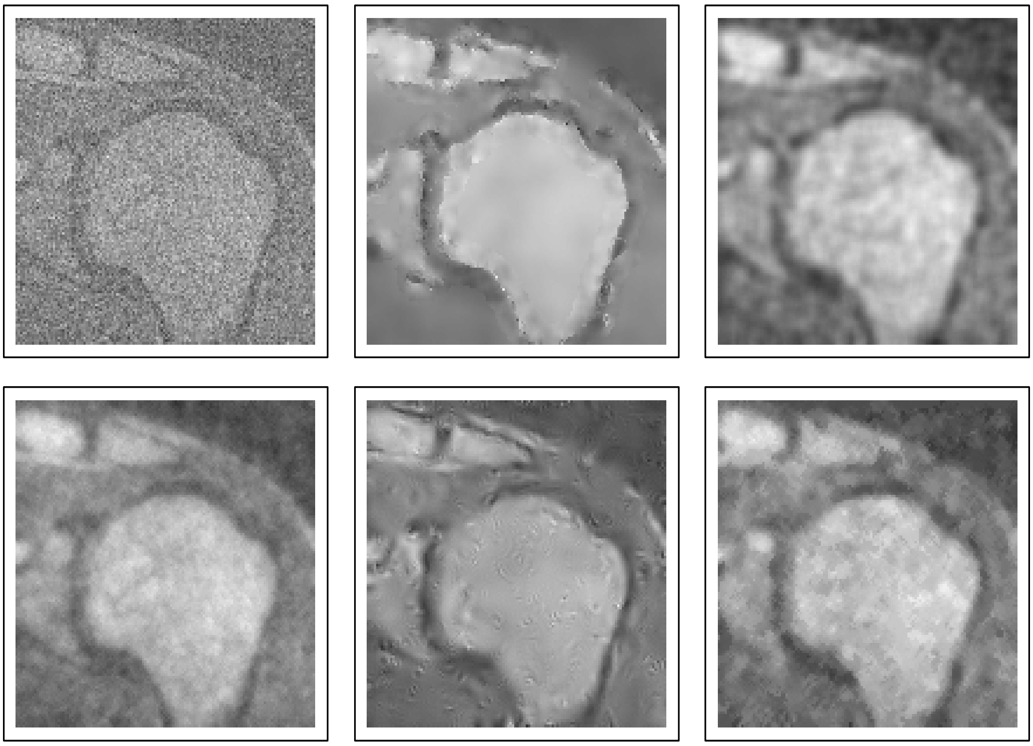

Figure 6: The noisy knee magnetic resonance image and its denoised images by methods NEW, GE, MRF, WT, and MED. The noise is from $N(0, 100^2)$.

The WT method considered here is designed for handling Gaussian noise. So, it is not surprising that it would perform well in the example of Figure 6 where noise is from a Gaussian distribution. Next, we consider the same MRI image, but add noise from a Double Exponential distribution with location parameter 0 and scale parameter 60. The corresponding results are shown in the third column of Table 4 and in Figure 7. We can see that WT does not perform well in this case in terms of the estimated MISE.

In the above examples, noise level is homogeneous in entire observed images, for simplicity. It should be pointed out that the proposed procedure can also handle cases when

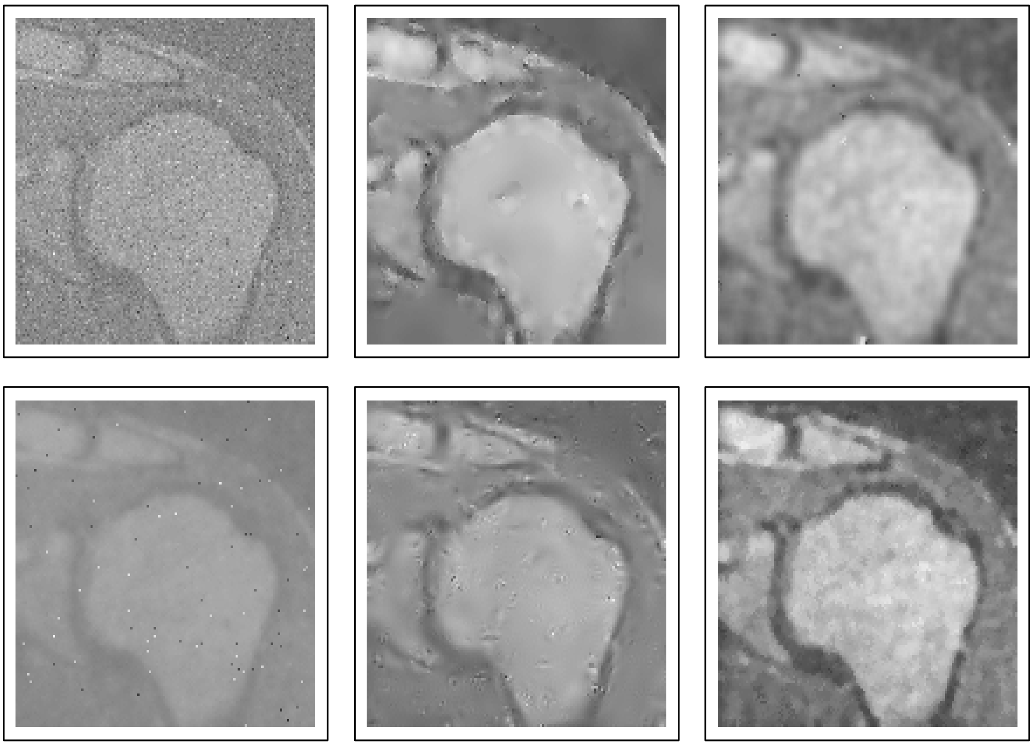

Figure 7: The noisy knee magnetic resonance image and its denoised images by methods NEW, GE, MRF, WT, and MED. The noise is from a Double Exponential distribution with location parameter 0 and scale parameter 60.

the noise level depends on location. As an example, suppose that the noise level has the expression

$$\sigma(x, y) = 40 \exp\left(3.5(x - 0.5)^2 + 3.5(y - 0.5)^2\right).$$

Obviously, $\sigma(x, y)$ is small in the central region of the image and large in the border region. An observed noisy knee image with i.i.d. noise from distribution $N(0, \sigma^2(x, y))$ is shown in the (1,1)-th plot of Figure 8. The estimated MISE values and denoised images of various methods are presented in the fourth column of Table 4 and in Figure 8, respectively. It can be seen that method NEW performs the best in this case.

Finally, we consider adding salt-and-pepper noise to 30% randomly selected pixels of the knee image. The added noise is binary. It equals either the maximum intensity level of the true knee image or the minimum intensity level, by random. The noisy image is shown

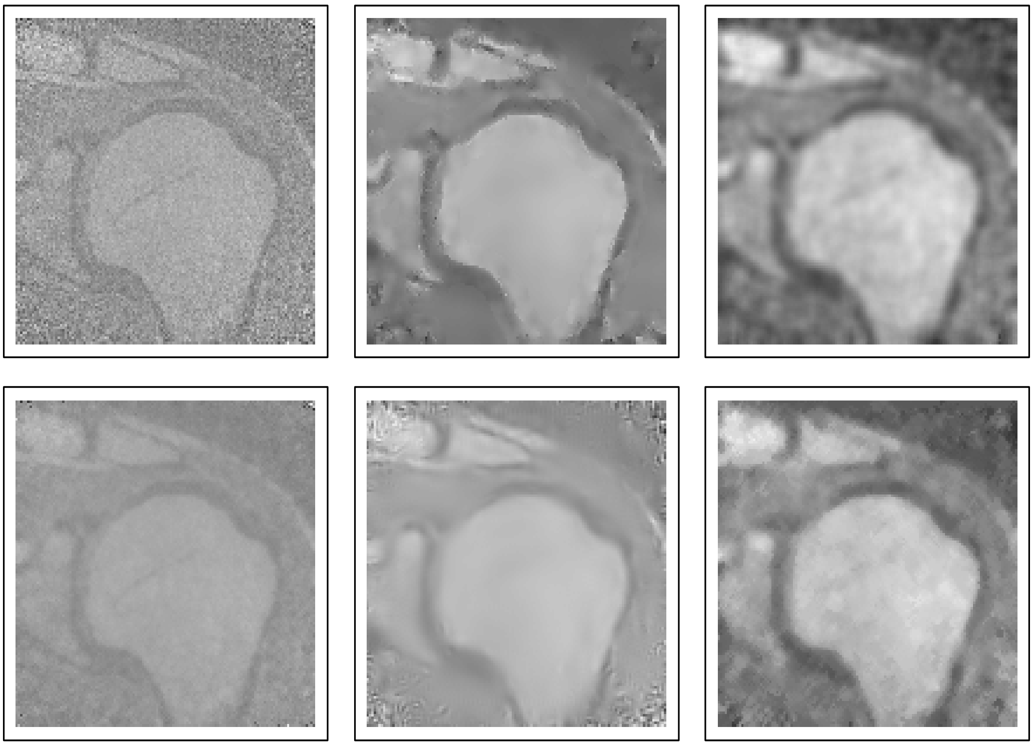

Figure 8: The noisy knee magnetic resonance image and its denoised images by methods NEW, GE, MRF, WT, and MED. The noise level changes with location by $\sigma(x, y) = 40 \exp\left(3.5(x - 0.5)^2 + 3.5(y - 0.5)^2\right)$.

in the (1,1)-th plot of Figure 9. The estimated MISE values and the denoised images of various methods are presented in the last column of Table 4 and in Figure 9, respectively. We can see that the median method MED performs the best in such a case, as expected, because median methods are robust to a small amount of large or small intensity values such as those with salt-and-pepper noise added. When the percentage of pixels with salt-and-pepper noise added increases, its performance becomes worse and worse, which has been numerically confirmed by us, although the results are skipped here. Method NEW performs better than the remaining three methods in this case.

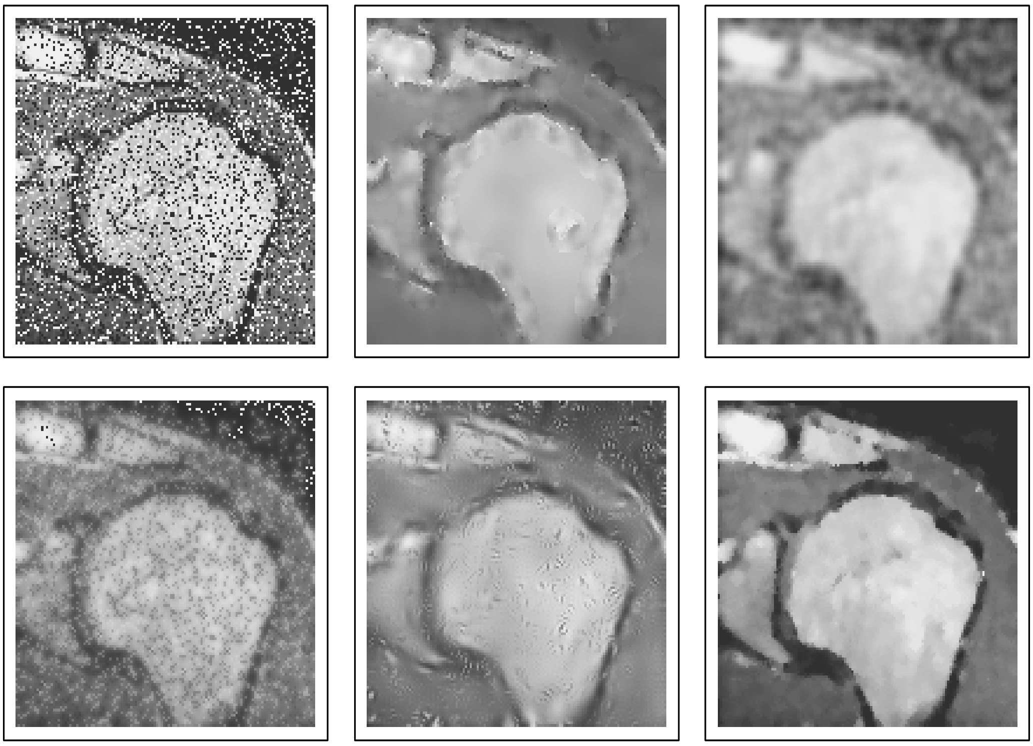

Figure 9: The noisy knee magnetic resonance image and its denoised images by methods NEW, GE, MRF, WT, and MED. Only 30% randomly selected pixels are added salt-and-pepper noise.

4 Summary and Concluding Remarks

We have presented an image denoising procedure in the framework of jump regression analysis. The new procedure can efficiently preserve both the parts of edges with small curvature and the parts of edges with large curvature. Numerical examples show that it performs well in various cases. 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 segments cross, because the true edge curves can not be approximated well by two half lines around a crossing point of several edge segments (cf., Figure 2). It requires much future research to denoise images properly in such cases. The proposed method consists of three steps. Although each step is based on local smoothing and the corresponding computation is thus fast, it is ideal to simplify the method by skipping or combining certain steps without sacrificing much of its denoising and edge structure preserving ability. In some applications (e.g., MRI, fMRI image analysis), denoising of 3-D images is necessary. In such cases, one possibility is to slide the 3-D image first and then denoise individual 2-D images slice by slice. But this would not be an efficient way of denoising 3-D images, because information in neighboring slices is not shared when individual 2-D slices are analyzed. It may not be straightforward to generalize the current method to 3-D cases due to the fact that edge structure becomes much more complicated in 3-D images. This topic also requires much future research.

As pointed out in Section 1, most existing image denoising procedures in the literature can not preserve certain edge structures well. For instance, the anisotropic diffusion filters control the degree of smoothing around a given pixel by a nonhomogeneous diffusivity which is often chosen to be a decreasing function of an estimated gradient (cf., Section 7.5, Qiu 2005). So, if the given pixel is close to an edge curve, then the estimated gradient would be large and consequently there would be less smoothing around that pixel. However, in order to remove noise, some smoothing is still necessary around edge curves. Therefore, a certain degree of edge blurring is inevitable by the diffusion filters. Because the diffusion filters usually do not take the shape of the edge curves into consideration, they could not preserve certain edge structures (e.g., angles) well. These comments can also be applied to the bilateral filters (e.g., Tomasi and Manduchi 1998), because Barash (2002) showed that bilateral filters were just special diffusion filters. Hillebrand and Müller (2007) recently demonstrated that the M-filter discussed by Chu et al. (1998) cannot handle isolated outliers well when denoising, and they proposed a modification to overcome that limitation by combining the ideas of robust estimation and trimming. Although proper selection of certain parameters needs to be further addressed, this modification makes a good contribution to the image denoising literature. Much future research is required to modify other existing denoising methods so that edge structures can be better preserved while noise and other possible contaminations are better removed.

Acknowledgments: This research is supported in part by an NSF grant. The authors thank the handling editor, Professor Dacheng Tao, for help and encouragement. We also thank the three reviewers for many constructive comments and suggestions that greatly improved the quality of the paper.

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. (1974), "Spatial interaction and the statistical analysis of lattice systems (with discussions)," *Journal of the Royal Statistical Society (Series B)*, **36**, 192–236.

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

Bovik, A.C., Huang, T.S., and Munson, D.C (1987), "The effect of median filtering on edge estimation and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **9**, 181–194.

Brownrigg, D.R.K. (1984), "The weighted median filtering," *Communications of the ACM*, **27**, 807–818.

Canny, J. (1986), "A Computational Approach to Edge Detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **8**, 679–698.

Chabat, F., Yang, G.Z., and Hansell, D.M. (1999), "A corner orientation detector," *Image and Vision Computing*, **17**, 761–769.

Chan, S.G., Yu, B., and Vetterli, M. (2000), "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on Image Processing,* **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.

Clark, J.J. (1988), "Singularity theory and phantom edges in scale space," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **10**, 720–727.

Fan, J., and Gijbels, I. (1996), *Local Polynomial Modelling and Its Applications,* Chapman & Hall: London.

Fessler, J.A., Erdogan, H., and Wu, W.B. (2000), "Exact distribution of edge-preserving MAP estimators for linear signal models with Gaussian measurement noise," *IEEE Transactions on image processing*, **9**, 1049–1055.

Figueiredo, A.A.T., and Nowak, R.D. (2001), "Wavelet-based image estimation: an empirical Bayes approach using Jeffrey's noninformative prior," *IEEE Transactions on Image Processing*, **10**, 1322–1331.

Fleck, M.M. (1992), "Some Defects in Finite-Difference Edge Finders," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **14**, 337–345.

Gallagher, N.C., Jr., and Wise, G.L. (1981), "A theoretical analysis of the properties of median filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **29**, 1136–1141.

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.

Gijbels, 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**, 1075–1087.

Gijbels, I., Lambert, A., and Qiu, P. (2007), "Jump-preserving regression and smoothing using local linear fitting: a compromise," *Annals of the Institute of Statistical Mathematics*, **59**, 235–272.

Godtliebsen, F., and Sebastiani, G. (1994), "Statistical methods for noisy images with discontinuities," *Journal of Applied Statistics*, **21**, 459–476.

Gonzalez, R.C., and Woods, R.E. (1992), *Digital Image Processing*, Addison-Wesley Publishing Company, Inc.

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

Huang, T.S. (1981), *Two-Dimensional Digital Signal Processing,* New York: Springer-Verlag.

Li, S.Z. (1995), "On discontinuity-adaptative smoothness prior in computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 576–586.

Marroquin, J.L., Velasco, F.A., Rivera, M., and Nakamura, M. (2001), "Gauss-Markov measure field models for low-level vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**, 337–347.

Moussouris, J. (1974), "Gibbs and Markov systems with constraints," *Journal of Statistical Physics*, **10**, 11–33.

Nason, G., and Silverman, B. (1994), "The discrete wavelet transform in S," *Journal of Computational and Graphical Statistics 3,* 163-191.

Perona, P., and Malik, J. (1990), "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629-639.

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

Portilla, J., Strela, V., Wainwright, M.J., and Simoncelli, E.P. (2003), "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, **12**, 1338–1351.

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

Qiu, P. (2002), "A nonparametric procedure to detect jumps in regression surfaces," *Journal of Computational and Graphical Statistics*, **11**, 799-822.

Qiu, P. (2003), "A jump-preserving curve fitting procedure based on local piecewise-linear kernel estimation," *Journal of Nonparametric Statistics*, **15**, 437–453.

Qiu, P. (2004), "The local piecewisely linear kernel smoothing procedure for fitting jump regression surfaces," *Technometrics*, **46**, 87–98.

Qiu, P. (2005), *Image Processing and Jump Regression Analysis*, New York: John Wiley & Sons.

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

Qiu, P. (2009), "Jump-preserving surface reconstruction from noisy data," *Annals of the Institute of Statistical Mathematics*, **61**, 715–751.

Qiu, P., and Bhandarkar, S.M. (1996), "An edge detection technique using local smoothing and statistical hypothesis testing," *Pattern Recognition Letters*, **17**, 849–872.

Rivera, M., and Marroquin, J.L. (2002), "Adaptive rest condition potentials: first and second order edge-preserving regularization, *Journal of Computer Vision and Image Understanding*, **88**, 76–93.

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**, 514–529.

Sinha , S.S., and Schunck, B.G. (1992), “A two-stage algorithm for discontinuity-preserving surface reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 36–55.

Sun, T., Gabbouj , M., and Neuvo, Y. (1994), “Center weighted median filters: some properties and their applications in image processing,” *Signal Processing*, **35**, 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**, 289–311.

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.

Torre, V. and Poggio, T.O. (1986), “On edge detection”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **8**, 147–163.

Yang, G.Z., Burger, P., Firmin, D.N., and Underwood, S.R. (1996), “Structure adaptative anisotropic image filtering,” *Image and Vision Computing*, **14**, 135–145.

Yi, J.H., and Chelberg, D.M. (1995), “Discontinuity-preserving and viewpoint invariant reconstruction of visible surfaces using a first order regularization”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 624-629.