

Some Recent Developments On Edge Detection And Image Reconstruction Based On Local Smoothing And Nonparametric Regression

Peihua Qiu

School of Statistics
University of Minnesota
313 Ford Hall
224 Church St. S.E.
Minneapolis, MN 55455

ABSTRACT

An image could be regarded as a surface which may have jumps at the outlines of the objects. Therefore edge detection and edge-preserving image reconstruction in image processing are essentially the same problems as jump detection and jump-preserving surface recovery in nonparametric statistics. Recently statistical analysis of jump surfaces recovery is under rapid development. In this paper we introduce some recent methodologies on jump detection and jump-preserving surface recovery in the statistical literature. Most of these methodologies are based on local smoothing and statistical nonparametric regression techniques such as local polynomial kernel estimation, local least squares estimation and local principal component analysis.

1 INTRODUCTION

An image could be regarded as a surface of the image intensity function. That is, the height of the surface at each pixel represents the brightness of the image at that pixel. In many situations the image concerned is noisy in the sense that the true image intensity function is contaminated by noise. For example, satellite images and the images transmitted by a cable picture transmission system are often noisy. A noisy image could be described by the following statistical model:

$$z_{ij} = f(x_i, y_j) + \epsilon_{ij}, \quad i, j = 1, 2, \dots, n, \quad (1)$$

where $\{(x_i, y_j)\}$ denote pixels, $f(x_i, y_j)$ is the true value of the image intensity function at pixel (x_i, y_j) , ϵ_{ij} is the noise at that pixel, z_{ij} is the observed value of the image intensity function at pixel (x_i, y_j) . In statistics, $\{(x_i, y_j)\}$ are often called design points, f is called a regression surface, $\{\epsilon_{ij}\}$ are random errors and $\{z_{ij}\}$ are observations. For simplicity of presentation, we assume that $\{(x_i, y_j) = (i/n, j/n), i, j = 1, 2, \dots, n\}$. That is, the design points are equally spaced in design space $[0, 1] \times [0, 1]$. We further assume that $\{\epsilon_{ij}\}$ are independent and identically distributed (i.i.d.) random errors with mean 0 and variance σ^2 . The total number of pixels (design points) in model (1) is $N = n^2$.

The image intensity function has step edges at the outlines of the objects^[1]. Due to the fact that much of the image information is conveyed by the edges and our eye-brain system has evolved to extract edges by preprocessing that begins right at the retina, edge detection and edge-preserving image reconstruction are important topics in image processing^[2]. Edges of images correspond to jump location curves (JLCs) of regression surfaces. The corresponding problems of edge detection and edge-preserving image reconstruction in image processing are jump detection and jump-preserving surface recovery (or surface fitting) in statistics. There are several types of edges discussed in the image processing literature. In this paper we mainly focus on the step edges which correspond to jumps in the regression surface itself (namely, in the zeroth order derivatives of f).

There are dozens of edge detection and edge-preserving image reconstruction proposals in the image processing literature (cf. the references given in the following sections). These proposals have the strength that they are intuitive and simple to use in applications. But some of them do not have enough theory to support them. A direct consequence of this fact is that it is often hard to select a proposal from dozens of existing ones for a specific application problem. Recently statistical analysis of regression models with jumps is under rapid development. As introduced above, the research problems in this area are essentially the same as those in edge detection and edge-preserving image reconstruction although different terminologies are used in these two areas. In this paper, we introduce some recent methodologies on jump detection (Section 2) and jump-preserving surface recovery (Section 4) in the presence of noisy data based on local smoothing and nonparametric regression analysis. These methodologies might be helpful for us to understand the strengths and limitations of some existing edge detection and edge-preserving image reconstruction proposals such that they can be further improved. The problem to evaluate the performance of jump detection is discussed in Section 3.

2 JUMP DETECTION BASED ON LOCAL SMOOTHING

Jump detection in regression surfaces is essentially the same problem as edge detection in image processing. Edge detectors (sometimes called filters in image processing) based on gradient estimation are “classic”. They make use of the property that estimators of the first order derivatives are large or infinite at edge pixels. Because these methods are intuitive and simple to use, they are included in almost all text books and softwares of edge detection (see e.g., [1,3]). More recent edge detection techniques are based on optimal filtering^[4], random field models^[5], surface fitting^[6], anisotropic diffusion^[7], local smoothing and hypothesis testing^[8], residual analysis^[9] and global cost minimization using hill-climbing search^[10], simulated annealing^[11] and the genetic algorithm^[12].

In the statistical literature there are several ways to define jump detection criterion (JDC) based on local smoothing. At design point (x_i, y_j) , Müller and Song^[13] and Qiu^[14] considered its two one-sided neighborhoods along a direction θ as shown by Figure

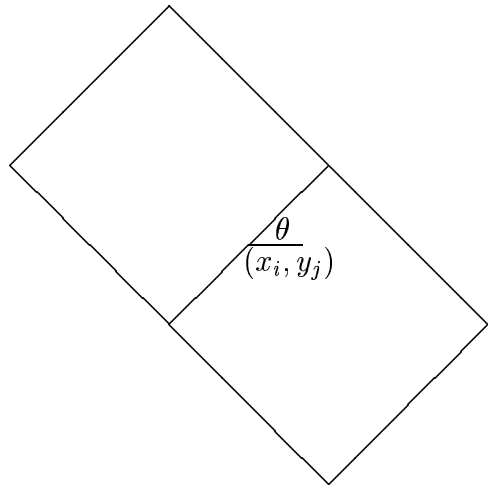


Figure 1: Two one-sided neighborhoods of design point (x_i, y_j) along direction θ .

1. Then weighted averages of the observations in the two neighborhoods were calculated, respectively, and the absolute difference of these two weighted averages was maximized with respect to θ . This maximized difference was used to detect jump at (x_i, y_j) with large value indicating a possible jump. Qiu^[14] called this proposal the *Rotational Difference Kernel Estimation* (RDKE) method. The corresponding jump detection criterion was defined by:

$$|JDC_{RDKE}(x, y)| = \max_{-\pi/2 \leq \theta \leq \pi/2} |JDC_{RDKE}(\theta, x, y)|, \quad (2)$$

where

$$JDC_{RDKE}(\theta, x, y) = \frac{1}{nh_n p_n} \sum_{i=1}^n \sum_{j=1}^n z_{ij} \left[K_2\left(\theta, \frac{x_i - x}{h_n}, \frac{y_j - y}{p_n}\right) - K_1\left(\theta, \frac{x_i - x}{h_n}, \frac{y_j - y}{p_n}\right) \right],$$

$K_i(\theta, x, y)$, $i = 1$ and 2 , were kernel functions obtained by rotating two one-sided kernel functions $K_i(x, y)$, $i = 1$ and 2 , an angle θ counterclockwise, $K_1(x, y)$ was upper-sided with support $[-1/2, 1/2] \times [0, 1]$ and $K_2(x, y)$ was lower-sided with support $[-1/2, 1/2] \times [-1, 0]$, h_n and p_n were two window widths.

If it is assumed that there is only one JLC which has a mathematical expression $y = \phi(x)$ for $0 \leq A \leq x \leq B \leq 1$, then an estimator of the JLC is given by $\hat{\phi}(\cdot)$ which is defined by

$$|JDC_{RDKE}(x, \hat{\phi}(x))| = \max_{0 \leq y \leq 1} |JDC_{RDKE}(x, y)|$$

for $A \leq x \leq B$. It was proved that the detected JLC based on the above jump detection criterion was

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Figure 2: Two Sobel masks.

almost surely consistent when the true JLC had tangent line at each point and the model (1) satisfied some other regularity conditions.

The above criterion (2) requires extensive computation because it searches all possible directions for a possible jump direction at each design point. To simplify its computation, two alternative approaches have been proposed recently. Qiu^[15] suggested a simplified version of criterion (2) which was defined by:

$$|JDC_{RDKE}^*(x, y)| = \max\{|JDC_{RDKE}(0, x, y)|, |JDC_{RDKE}(\pi/2, x, y)|\}. \quad (3)$$

In (3), only two directions (the x and y directions) were searched to detect a possible jump at (x, y) . It was shown that as long as h_n and p_n were *asynchronous* in the sense that $\lim_{n \rightarrow \infty} h_n/p_n = 0$, the detected jumps by criterion (3) were almost surely consistent under some regularity conditions. Qiu^[15] also discussed the trade-off between the amount of computation and the accuracy of the detected jumps by searching different numbers of directions at each design point.

The criterion (3) is a generalization of the Sobel operator which is based on two Sobel masks as shown by Figure 2 (see e.g., [16]). For a given design point (x, y) , a 3×3 neighborhood is considered. Convolution of the first mask with the observations in the neighborhood is used to estimate the partial derivative of the image intensity function with respect to y . This estimator is denoted as $\hat{f}_y(x, y)$. Similarly, the second mask is used to obtain an estimator of the partial derivative with respect to x , which is denoted as $\hat{f}_x(x, y)$. Then $[(\hat{f}_x(x, y))^2 + (\hat{f}_y(x, y))^2]^{1/2}$ is used as an edge detection criterion with large values indicating possible edges.

The quantity $\hat{f}_y(x, y)$ is similar to our $JDC_{RDKE}(0, x, y)$. It is also a difference of two weighted averages. Similarly, $\hat{f}_x(x, y)$ is related to $JDC_{RDKE}(\pi/2, x, y)$. The Sobel edge detector used the Euclidean length of the estimated gradient as its edge detection criterion. In (3), we suggested

using the maximum of $|JDC_{RDKE}(0, x, y)|$ and $|JDC_{RDKE}(\pi/2, x, y)|$ to detect jumps based on the following consideration. When (x, y) is on a JLC, the jump structure of the regression surface contaminates some of the four kernel averages used in constructing $JDC_{RDKE}^*(x, y)$ (cf. (3)) as estimators of the regression surface at (x, y) . For example, if the JLC is parallel to the x -axis, then the two kernel averages in $JDC_{RDKE}(0, x, y)$ estimate the surface well. Consequently, $JDC_{RDKE}(0, x, y)$ is a good estimator of the jump magnitude at (x, y) . But the two kernel averages in $JDC_{RDKE}(\pi/2, x, y)$ do not provide much helpful information for jump detection. It could only have negative effect if they are included in the jump detection criterion since the criterion will become noisier. This negative effect is mostly eliminated by using $JDC_{RDKE}^*(x, y)$.

Qiu and Yandell^[17] suggested estimating a possible jump direction at each design point (x_i, y_j) by its gradient direction which could be estimated by fitting a local least squares (LLS) plane in a square neighborhood $N(x_i, y_j)$ with width h_n . Then two neighboring design points (x_{P_1}, y_{P_1}) and (x_{P_2}, y_{P_2}) of the design point (x_i, y_j) along the estimated gradient direction \vec{v}_{ij} were defined as shown by Figure 3. A jump detection criterion was then defined as follows:

$$JDC_{LLS}(x_i, y_j) = \min\{\|\vec{v}_{ij} - \vec{v}_{P_1}\|, \|\vec{v}_{ij} - \vec{v}_{P_2}\|\}, \quad (4)$$

where \vec{v}_{P_1} and \vec{v}_{P_2} were the estimated gradient directions at (x_{P_1}, y_{P_1}) and (x_{P_2}, y_{P_2}) , respectively. Some variants of (4) are possible. For example, Qiu^[18] replaced $\vec{v}_{ij}, \vec{v}_{P_1}$ and \vec{v}_{P_2} in (4) by three weighted averages of the observations in the three neighborhoods, respectively.

There are several other proposals to define jump detection criterion in the literature. For example, O'Sullivan and Qian^[19] defined a ‘‘contrast statistic’’ to detect jumps which was a difference of two averages of the observations located inside and outside of a candidate JLC, respectively. (They assumed that there was only one JLC which was a ‘‘smooth, simple and closed’’ curve.) Hall and Rau^[20] suggested a sequential, or ‘‘tracking’’, algorithm for estimating a JLC. Wang^[21] suggested estimating a JLC based on wavelet transformations.

After defining a jump detection criterion, the next step is to use it for detecting jumps. In the statistical literature, there are two major strategies for this purpose. The first strategy is to search for a candidate JLC from a population of all possible candidates (e.g., [13,14,19,22,23]). This idea is natural. But it often

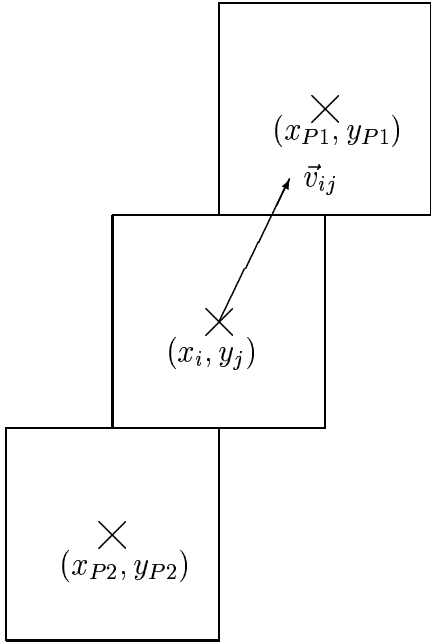


Figure 3: Two neighboring design points (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}) of (x_i, y_j) were defined by the design points which neighborhoods were adjacent to $N(x_i, y_j)$ on either side along the direction of \vec{v}_{ij} .

requires restrictive assumptions on the JLCs. For example, a common assumption is that the number of JLCs is known beforehand, which is hard to be satisfied in applications. Another difficulty with this idea is the extensive computation involved. The second strategy is to regard the JLCs as a *point set* in the design space and estimate them by another point set in the same design space^[15,17,18]. For example, the criterion $JDC_{RDKE}^*(x, y)$ in (3) should be close to zero if (x, y) is a continuous point of the regression surface. On the other hand, if (x, y) is on a JLC, then one of $JDC_{RDKE}(0, x, y)$ and $JDC_{RDKE}(\pi/2, x, y)$ should be close to the jump magnitude at (x, y) . Consequently, $JDC_{RDKE}^*(x, y)$ is relatively large. It is then natural to use the point set

$$\widehat{D}_n := \{(x_i, y_i) : JDC_{RDKE}^*(x_i, y_i) \geq u_n\}, \quad (5)$$

to estimate the point set of the JLCs which is defined by $D := \{(x, y) : (x, y) \text{ is some point on the JLCs}\}$ where u_n is a positive threshold value. Qiu^[15] provided a formula for calculating u_n by which u_n has the property that when (x, y) is a continuous point and n is large enough,

$$Prob(JDC_{RDKE}^*(x, y) > u_n) \leq \alpha_n,$$

where α_n is a significance level. It was demonstrated that \widehat{D}_n in (5) provided a good estimator of D except around the so-called *singular points* of the JLCs.

Comparing the jump detection procedures in non-parametric regression and the edge detection procedures in image processing, we should say that both of these two groups of procedures have their strengths and limitations. Most jump detection procedures have their own theory to support them. The conditions required on the JLCs and on model (1), under which these procedures can successfully detect the jumps, are often explicitly given. Therefore we usually know in which circumstances a specific procedure can work well. On the other hand, many jump detection procedures in the statistical literature require extensive computation because they are based on numerous maximization/minimization procedures to search for the JLCs. For example, the maximization procedures used in some kernel-type methods^[13,14] need great amount of computation. It is not easy to compute the maximum likelihood estimators of the piecewise polynomial coefficients either in the proposal suggested by Korostelev and Tsybakov^[22]. These procedures also impose various model assumptions, many of which are hard to be satisfied in applications. For example, Müller and Song^[13] and Qiu^[14] assumed that the number of JLCs was known beforehand.

Most edge detection methods in the image processing literature are ready to use in applications. But some of them do not have enough theory to support them. For example, the Sobel edge detector described above often uses a 3×3 window (called *mask* in image processing) at each pixel to obtain estimators of the first-order partial derivatives. If the window size increases to $k_1 \times k_2$ with $k_1, k_2 > 3$, how can the Sobel operator be modified accordingly such that the detected edges are statistically consistent (namely, the detected edges converge to the true edges when k 's and the image resolution tend to infinity)? Should k_1 and k_2 equal to each other? We have not seen much discussion of this type in the image processing literature yet. Although edge detectors can be evaluated by numerical experiments based on visual impression, we believe that theoretical justifications can help us understand their strengths and limitations such that they can be further improved.

3 EVALUATION OF JUMP DETECTION PERFORMANCE

To compare different jump detection methods by simulations, it is necessary to suggest some measure-

ments to evaluate jump detection performance. Because both the true JLCs and the estimated JLCs can be regarded as point sets in the design space as mentioned in Section 2 (see equation (5)), a natural choice of the measurement is the Hausdorff distance between two point sets \mathcal{A} and \mathcal{B} used in mathematics, which is defined by:

$$d_H(\mathcal{A}, \mathcal{B}) := \max(\sup_{x \in \mathcal{A}} \inf_{y \in \mathcal{B}} \|x - y\|, \sup_{x \in \mathcal{B}} \inf_{y \in \mathcal{A}} \|x - y\|).$$

Let us assume that \mathcal{A} denotes the set of true jump points and \mathcal{B} denotes the set of detected jump points. In simulation studies, \mathcal{A} is often specified beforehand and in most situations it can be expressed by some mathematical equations. Therefore it is not hard to compute $\inf_{y \in \mathcal{A}} \|x - y\|$ for each $x \in \mathcal{B}$. However, it is often hard to compute $\inf_{y \in \mathcal{B}} \|x - y\|$ for each $x \in \mathcal{A}$ since \mathcal{B} does not have any mathematical expressions in most situations and therefore we need to make a judgement for each design point to see if it is a detected jump point, which makes the computational complexity of the Hausdorff distance be $O(N^{3/2})$. Another limitation of the Hausdorff distance is explained by the following example^[15].

Example 1 Suppose that the true JLC is a line $\mathcal{A} = \{(x, y) : y = .5, 0 \leq x \leq 1\}$ and there are two sets of detected jumps. The first set consists of the design points on line $y = .5$ and a point $(.8, .1)$ (Figure 4(a)). The second set consists of the design points on line $y = .2$ (Figure 4(b)). The Hausdorff distance between the true JLC and the first set of detected jumps is .4 while it is .3 for the second set. Therefore the performance of the second set is better than the performance of the first set by the Hausdorff distance, which might be the opposite to what we would expect. A main reason behind this phenomenon is that the Hausdorff distance is sensitive to individual points (point $(.8, .1)$ in the case of Figure 4(a)).

Qiu^[15] suggested an alternative performance measurement defined by

$$d_Q(\mathcal{A}, \mathcal{B}) := .5 \frac{|\mathcal{B} \setminus \mathcal{A}|}{|\mathcal{A}|} + .5 \frac{|\mathcal{A} \setminus \mathcal{B}|}{|\mathcal{A}|},$$

where $\bar{\mathcal{A}}$ denotes the complementary set of \mathcal{A} and $|\mathcal{A}|$ denotes the number of points in set \mathcal{A} . Clearly, $\frac{|\mathcal{B} \setminus \mathcal{A}|}{|\mathcal{A}|}$ is the proportion of all false jump points detected by the jump detection procedure and $\frac{|\mathcal{A} \setminus \mathcal{B}|}{|\mathcal{A}|}$ is the percentage of all true jump points missed by the procedure. The measurement $d_Q(\mathcal{A}, \mathcal{B})$ is their average. Apparently, we could also use the weighted

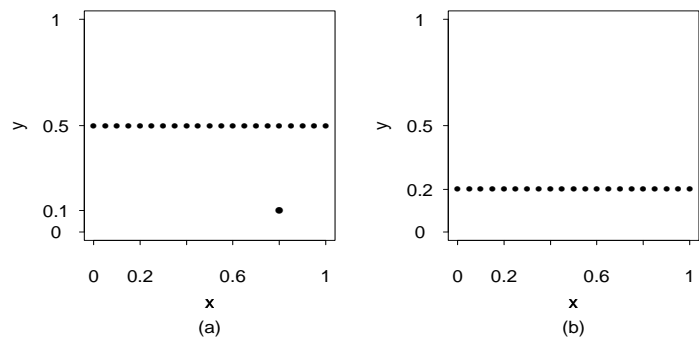


Figure 4: (a) The detected jumps are the design points on line $y = .5$ and a point $(.8, .1)$; (b) the detected jumps include the design points on line $y = .2$ only. The true JLC is the line $y = .5$.

average:

$$w \frac{|\mathcal{B} \setminus \mathcal{A}|}{|\mathcal{A}|} + (1 - w) \frac{|\mathcal{A} \setminus \mathcal{B}|}{|\mathcal{A}|},$$

where the weight $0 \leq w \leq 1$ represents the relative importance of the first percentage and it needs to be specified by users.

We first notice that the computation of $d_Q(\mathcal{A}, \mathcal{B})$ is simple. For each point in \mathcal{B} , we can first make a judgement to see if it also belongs to \mathcal{A} . Then $d_Q(\mathcal{A}, \mathcal{B})$ can be computed from the number of points in both \mathcal{A} and \mathcal{B} , the number of points in \mathcal{B} , and the number of points in \mathcal{A} . So its computational complexity is $O(N)$. Secondly, $d_Q(\mathcal{A}, \mathcal{B})$ is more robust to individual points since it is based on proportions instead of maximum/minimum values. It is not hard to check that $d_Q(\mathcal{A}, \mathcal{B})$ is close to 0 in the case of Figure 4(a) and larger than .5 in the case of Figure 4(b).

However, $d_Q(\mathcal{A}, \mathcal{B})$ is not perfect for applications. In the example of Figure 4, if the detected jumps are the design points on line $y = .5 + 1/n$, then probably we would expect that the jump detection procedure performs better when n gets larger. By $d_Q(\mathcal{A}, \mathcal{B})$, it will not happen. We realize that to measure the jump detection performance is an important issue because it is directly related to comparisons of different jump detection procedures. Therefore more research on this topic is needed in the future.

4 JUMP-PRESERVING SURFACE RECOVERY

It is still a problem how to recover the entire surface from noisy data with jumps preserved after the possible jump points being detected. The surface recovery

problem is essentially the same problem as image reconstruction in image processing except that the image intensity function often takes discrete values (e.g., black and white images or gray-scale images with 256 gray levels; cf. Switzer^[24] for discussion about types of images) while the response variable in regression setup is often continuous. When the true image has only several possible colors, the Markov Random Field (MRF) methods are commonly used in the literature to recover the true image from noisy data^[5,25]. These methods usually assume that the true image is a MRF and the observations have a given conditional distribution conditional on the true image. Then the true image is estimated by maximizing *a posteriori*. The Markov chain Monte Carlo (MCMC) techniques are often used in the related computation^[26]. The image reconstruction problem can also be formulated in a regularization framework^[27] which is based on minimizing a target function, a trade-off between a fidelity measurement to the data and a smoothness measurement of a candidate image (see also [28,29]). Green^[30], O'Sullivan^[31] and the references cited there discussed image reconstruction in emission tomography.

In the statistical literature there are several recent methodologies for image reconstruction based on local smoothing. Chu *et al.*^[32] discussed two types of edge-preserving smoothers: the sigma filter and the M smoother. The estimator of the regression surface at point (x_{i_0}, y_{j_0}) by the sigma filter is defined by

$$\hat{f}(x_{i_0}, y_{j_0}) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} z_{ij}, \quad (6)$$

where

$$w_{ij} = \frac{K_1\left(\frac{x_i - x_{i_0}}{h_{11}}, \frac{y_j - y_{j_0}}{h_{12}}\right) K_2\left(\frac{z_{ij} - z_{i_0 j_0}}{h_2}\right)}{\sum_{i=1}^n \sum_{j=1}^n K_1\left(\frac{x_i - x_{i_0}}{h_{11}}, \frac{y_j - y_{j_0}}{h_{12}}\right) K_2\left(\frac{z_{ij} - z_{i_0 j_0}}{h_2}\right)},$$

$K_1(\cdot, \cdot)$ and $K_2(\cdot, \cdot)$ are two kernel functions (often taken as probability densities), h_{11} , h_{12} and h_2 are window widths. The surface estimator $\hat{f}(x_{i_0}, y_{j_0})$ defined in (6) is a weighted average of the observations in a neighborhood of a given design point. The weights, however, are put on both the design points and the observations. Observations which design points are closer to (x_{i_0}, y_{j_0}) receive more weights. Similarly they receive more weights if their magnitudes are closer to $z_{i_0 j_0}$.

The surface estimator at a given design point (x_{i_0}, y_{j_0}) by the M smoother is defined by the local

minimizer of the following M function

$$S(\theta) = \sum_{i=1}^n \sum_{j=1}^n \rho(z_{ij} - \theta) K\left(\frac{x_i - x_{i_0}}{h}, \frac{y_j - y_{j_0}}{g}\right),$$

where $\rho(\cdot)$ is a robust function which has the property that $\rho(y)$ does not grow too rapidly when $|y|$ increases, $K(\cdot, \cdot)$ is a kernel function and h and g are window widths. Chu *et al.*^[32] also provided an algorithm for finding this local minimizer. Based on a simulation study and some asymptotic analysis, they found that the M smoother often had better performance than the sigma filter.

In Section 2, we introduced the idea to estimate the JLCs by a point set in the design space. Based on that idea, Qiu^[18] suggested the following three-stage algorithm to fit surfaces with jumps preserved. The first step is to detect jump points by some jump detectors as discussed in Section 2. In the second step, we first define a neighborhood of design point (x_i, y_j) by $N^*(x_i, y_j) := \{(x_{i+s}, y_{j+t}) : s, t = -\ell^*, \dots, 0, \dots, \ell^*\}$ where $k^* = 2\ell^* + 1$ is a window width which could be different from the window width used in the first step. Suppose that the detected jump points in $N^*(x_i, y_j)$ are $\{(w_r, v_r), r = 1, 2, \dots, m\}$. Then a local principal component (PC) line is fitted through these jump candidate points, which turns out to be

$$\sigma_{wv}(x - \bar{w}) + (\lambda_1 - \sigma_{wv})(y - \bar{v}) = 0,$$

where

$$\lambda_1 := \frac{1}{2} \left(\sigma_{wv} + \sigma_{vv} - \sqrt{(\sigma_{wv} - \sigma_{vv})^2 + 4\sigma_{wv}^2} \right),$$

\bar{w} , \bar{v} , σ_{wv} and σ_{vv} are the sample means and variances of w 's and v 's, and σ_{wv} is their covariance. Qiu^[18] proved that the fitted PC line provided a first order approximation to the true JLC around (x_i, y_j) in the sense that if (x_i, y_j) was the closest design point to a given point P on the JLC at which the JLC had a unique tangent line, then the PC line in $N^*(x_i, y_j)$ converged to the tangent line both pointwisely and in direction with probability 1. Finally in the third step, observations in $N^*(x_i, y_j)$ which are located on the same side of the PC line as (x_i, y_j) are combined as a weighted average to fit the surface at (x_i, y_j) . The computational complexity of this algorithm is $O(N(k^*)^2)$. This property makes it appropriate to handle large data sets. Its assumptions on the JLCs and the underlying regression surface are flexible. We proved that the fitted surface was L^2 consistent in the continuous regions with the optimal convergence rate

of nonparametric regression and L^2 consistent in the neighborhood of the JLCs as well but with a slower convergence rate.

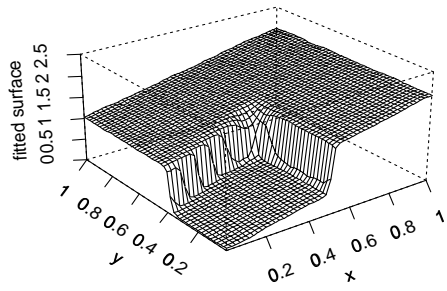
Example Consider a regression function $f(x, y) = x + I_{(y > .375 + .25x \text{ or } x > .25)}$ which has jumps at JLC: $\{(x, y) : y = .375 + .25x \text{ and } 0 \leq x \leq .5\} \cup \{(x, y) : x = .5 \text{ and } 0 \leq y \leq .5\}$ with constant jump magnitude 1. A sample is generated from model (1) with $n = 100$ and $\epsilon_{11} \sim N(0, .5^2)$. After the possible jump points being detected by the criterion suggested by Qiu^[18], the surface is fitted by the above three-stage algorithm. The fitted surface with $k^* = 15$ is presented in Figure 5(a). It is an average of 100 replications (averaged results are presented here to remove some randomness such that systematic drawbacks of the method could be visually revealed). As indicated by the plot, the three-stage algorithm works well except around the singular point (.5, .5) (see Qiu^[18] for definition of a singular point). The 2.5 and 97.5 percentiles of 100 replications of the surface fit in the cross section of $x = .25$ are shown in Figure 5(b). The conventional local smoothing techniques are built in most geographic softwares such as ARC/INFO to (1) remove the noise from an image and (2) change an image from one resolution to another. As a comparison, Figure 5(c) shows the fitted surface by the conventional kernel smoothing technique with the same window width and kernel function as those used in Figure 5(a). Figure 5(d) presents 2.5 and 97.5 percentiles of 100 replications of the kernel smoothing surface fit in the cross section of $x = .25$. We can see that “blurring” happens around the true JLC.

As mentioned in Section 2, several authors suggested estimating the JLCs by curves with compact forms (e.g., [13,14,19]). In such case, the regression surface can be fitted as usual in design subspaces separated by the estimated JLCs. But it is often difficult to estimate the JLCs in that way in applications due to the extensive computation involved and the restrictive assumptions imposed on the JLCs.

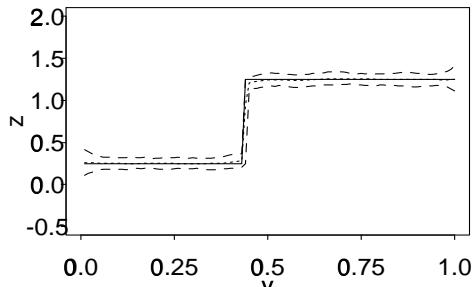
REFERENCES

- [1] Gonzalez, R.C., and Woods, R.E. (1992), *Digital Image Processing*, Addison-Wesley Publishing Company, Inc.
- [2] Bracewell, R.N. (1995), *Two-Dimensional Imaging*, Prentice-Hall, Inc.: New Jersey.
- [3] Marr, D., and Hildreth, E. (1980), “Theory of edge detection,” *Proceedings of the Royal Society of London 207*, 187-217.
- [4] Canny, J. (1986), “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 679-698.
- [5] 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.
- [6] Haralick, R.M. (1984), “Digital step edges from zero crossing of second directional derivatives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 58-68.
- [7] 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.
- [8] Qiu, P., and Bhandarkar, S.M. (1996), “An edge detection technique using local smoothing and statistical hypothesis testing,” *Pattern Recognition Letters 17*, 849-872.
- [9] Chen, M.H., Lee, D., and Pavlidis, T. (1991), “Residual analysis for feature detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 30-40.
- [10] Tan, H.L., Gelfand, S.B., and Delp, E.J. (1989), “A comparative cost function approach to edge detection,” *IEEE Transaction on Systems, Man, and Cybernetics 19*, 1337-1349.
- [11] Tan, H.L., Gelfand, S.B., and Delp, E.J. (1991), “A cost minimization approach to edge detection using simulated annealing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 3-18.
- [12] Bhandarkar, S.M., Zhang, Y., and Potter, W.D. (1994), “An edge detection technique using genetic algorithm-based optimization,” *Pattern Recognition 27*, 1159-1180.

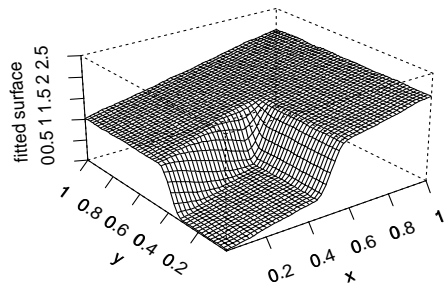
- [13] Müller, H.G., and Song, K.S. (1994), "Maximin estimation of multidimensional boundaries," *Journal of the Multivariate Analysis* 50, 265-281.
- [14] Qiu, P. (1997), "Nonparametric estimation of jump surface," *Sankhyā (A)* 59, 268-294.
- [15] Qiu, P. (1998), "A nonparametric procedure to detect jumps in regression surfaces," *Technical Report 626*, School of Statistics, University of Minnesota.
- [16] Rosenfeld, A., and Kak, A.C. (1982), *Digital Picture Processing*, Vols. 1 and 2, Academic Press, New York.
- [17] Qiu, P., and Yandell, B. (1997), "Jump detection in regression surfaces," *Journal of Computational and Graphical Statistics* 6, 332-354.
- [18] Qiu, P. (1998), "Discontinuous regression surfaces fitting," *The Annals of Statistics* 26, 2218-2245.
- [19] O'Sullivan, F., and Qian, M. (1994), "A regularized contrast statistic for object boundary estimation – implementation and statistical evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16, 561-570.
- [20] Hall, P., and Rau, C. (1998), "Tracking a smooth fault line in a response surface," Manuscript.
- [21] Wang, Y. (1998), "Change curve estimation via wavelets," *Journal of the American Statistical Association* 93, 163-172.
- [22] Korostelev, A.P., and Tsybakov, A.B. (1993), *Minimax Theory of Image Reconstruction*, Lecture Notes in Statistics, Vol. 82, Springer, New York.
- [23] Rudemo, M., and Stryhn, H. (1994), "Approximating the distribution of Maximum likelihood contour estimators in two-region images," *Scandinavian Journal of Statistics* 21, 41-55.
- [24] Switzer, P. (1986), "Statistical image processing", In *Quantitative Analysis of Mineral and Energy Resources*, (C.F. Chung, A.G. Fabbri, and R. Sinding-Larsen, eds.) Reidel, Dordrecht, 271-282.
- [25] Besag, J. (1986), "On the statistical analysis of dirty pictures (with discussion)", *Journal of the Royal Statistical Society B* 48, 259-302.
- [26] Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995), "Bayesian computation and stochastic systems (with discussion)", *Statistical Science* 10, 3-66.
- [27] Titterton, D.M. (1985), "Common structure of smoothing techniques in statistics", *International Statistical Review* 53, 141-170.
- [28] Sinha, S.S., and Schunk, B.G. (1992), "A two-stage algorithm for discontinuity-preserving surface reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 36-55.
- [29] Li, S.Z. (1995), "On discontinuity-adaptive smoothness priors in computer vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, 576-586.
- [30] Green, P.J. (1990), "Bayesian reconstruction from emission tomography data using a modified EM algorithm", *IEEE Transactions on Medical Imaging* 9, 84-93.
- [31] O'Sullivan, F. (1995), "A study of least squares and maximum likelihood for image reconstruction in positron emission tomography", *The Annals of Statistics* 23, 1267-1300.
- [32] Chu, C.K., Glad, I.K., Godtliebsen, F., and Marron, J.S., (1998), "Edge-preserving smoothers for image processing," *Journal of the American Statistical Association* 93, 526-556.



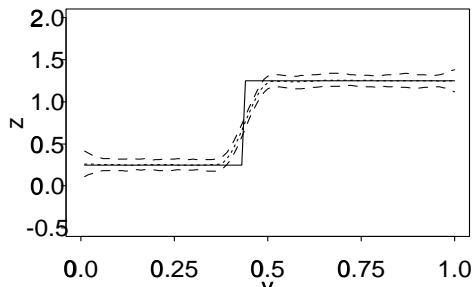
(a)



(b)



(c)



(d)

Figure 5: (a) Fitted surface by the three-stage surface fitting algorithm based on 100 replications; (b) the 2.5 and 97.5 percentiles of 100 replications of the surface fit by the three-stage algorithm in the cross section of $x = .25$; (c) fitted surface by the conventional kernel smoothing algorithm; (d) the 2.5 and 97.5 percentiles of 100 replications of the surface fit by the conventional kernel smoothing technique in the cross section of $x = .25$. In plots (b) and (d), the dashed curves denote the percentiles, the solid curves are the cross section of the true surface and the dotted curves are the averaged fits.