A NONPARAMETRIC PROCEDURE TO DETECT JUMPS IN REGRESSION SURFACES

Peihua Qiu
School of Statistics
University of Minnesota
313 Ford Hall
224 Church Street SE
Minneapolis, MN 55455

Abstract

A local smoothing procedure is proposed for detecting jump location curves of regression surfaces. This procedure simplifies the computation of some existing jump detectors in the statistical literature. It also generalizes the Sobel edge detector in the image processing literature such that more observations can be used to smooth away random noise in the data. The problem to evaluate the performance of jump detectors is discussed and a new measurement of jump detection performance is suggested.

Key Words: Edge detection; Hausdorff distance; Jump regression surface; Kernel estimation; Performance measurement; Singular points; Sobel edge detector; Strong consistency.

1 Introduction

This paper proposes a methodology for detecting jump locations of regression surfaces. Surface fitting is a fundamental problem in many applications. For example, meteorologists are interested in fitting the equi-temperature surfaces in high sky or deep ocean. Geologists often need to recover the mine surfaces from mineral samples. In some situations, the related surface is discontinuous and the locations of discontinuity are curves in the design space. We call these curves the jump location curves (JLCs) hereafter. An important example is that the image intensity function of an image has step discontinuities (called *step edges* in image processing) at the outlines of the objects. It is important to detect JLCs for surface fitting (surfaces can be fitted as usual in regions separated by the detected JLCs, see e.g., Müller and Song 1994) and for understanding the surface structure

(edges are often regarded as important image structures because much of the image information is conveyed by the edges, see e.g., Gonzalez and Woods 1992).

In the statistical literature, several proposals have been suggested for detecting JLCs. For example, Korostelev and Tsybakov (1993) suggested using piecewise polynomials to approximate the JLCs and the polynomial coefficients were estimated by the maximum likelihood procedure. O'Sullivan and Qian (1994) defined a contrast statistic to detect object boundaries. The boundary curves were assumed to be "smooth simple closed" curves. Hall and Raimondo (1997) provided an almost sure convergence rate for approximating a line that separated an entire image into two regions with two different colors. Qiu and Yandell (1997) proposed a jump detection algorithm based on local least squares estimation. Wang (1998) suggested estimating "change curves" via wavelets. Müller and Song (1994) proposed "maximin" estimators of the jump boundaries of the d-dimensional ($d \geq 1$) jump surfaces. Qiu (1997) suggested a so-called *Rotational Difference Kernel Estimator (RDKE)* of the JLCs. For jump-preserving surface fitting methods, see Chu *et al.* (1998), Qiu (1998) and the references cited there.

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". These detectors are based on the property that the first order derivatives of the image intensity function are large or infinite at edge pixels. Because they are intuitive and simple to use, they are included in most text books on edge detection (see e.g., Marr and Hildreth 1980; Gonzalez and Woods 1992). More recent edge detection techniques are based on optimal filtering (Canny 1986), random field models (Geman and Geman 1984), surface fitting (Haralick 1984), anisotropic diffusion (Perona and Malik 1990), local smoothing and hypothesis testing (Qiu and Bhandarkar 1996), residual analysis (Chen et al. 1991) and global cost minimization using hill-climbing search (Tan *et al.* 1989), simulated annealing (Tan *et al.* 1991) and the genetic algorithm (Bhandarkar *et al.* 1994).

The two groups of methods mentioned above have their own limitations. Most jump detection procedures in the statistical literature require extensive computation because they are based on a number of maximization/minimization procedures to search for the JLCs. For example, the maximization procedures used in some kernel-type methods (Müller and Song 1994; Qiu 1997) requires a great amount of computation. It is not easy to compute the maximum likelihood estimators

of the piecewise polynomial coefficients either in the procedure suggested by Korostelev and Tsybakov (1993). These procedures also require some restrictive model assumptions. A commonly used assumption is that the number of JLCs is known beforehand, which is hard to be satisfied in real applications. Part of the reason behind these limitations, I think, is that (1) the JLCs are regarded as and estimated by curves instead of point sets in these methods; and (2) all directions are searched at each design point for a possible jump direction. Nevertheless, these procedures often have their own theory to support them, which is important for them to be further improved.

Most edge detection procedures in the image processing literature are simple to use due to the fact that (1) they treat edges of an image as a point set and consequently the detected edges of these procedures consist of individual pixels; and (2) only a few directions (most often the x and y directions) are searched at each pixel for a possible jump. However, a major limitation of these edge detectors is the fact that they often do not have enough theory to support them. For example, the conventional Sobel edge detector (see e.g., Rosenfeld and Kak 1982) uses a 3×3 window (called *mask* in the image processing literature) at each pixel to obtain estimators of the first-order partial derivatives of the image intensity function. If the window size increases to $k_1 \times k_2$ with k_1 and $k_2 > 3$, 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_1, k_2 and the image resolution tend to infinity)? Should k_1 and k_2 equal 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 impession, we believe that theoretical justifications can help us understand their strengths and limitations such that they can be further improved.

This article represents part of our research effort to connect these two groups of jump detection methods in the statistical literature and image processing literature. A jump detection procedure is suggested based on kernel smoothing techniques. Unlike some existing kernel-type methods, the current procedure avoids using the maximization procedure with respect to the direction parameter at each design point. Instead it searches the x and y directions only for jump detection, making its computation simple. The number of JLCs could be unknown by using this procedure. Other model assumptions are also flexible. This procedure can be regarded as a generalization of the conventional Sobel edge detector in the sense that its window widths are not restricted to 3×3 any more although they need to satisfy some regularity conditions such that the detected edges

are statistically consistent. The conditions on the JLCs required by this procedure are related explicitly to the way it detects jumps, which is helpful for users to know the potential places in the design space at which the procedure may fail to detect jumps.

The remaining part of the article is organized as follows. In Section 2, our jump detection procedure is introduced in detail along with the connection between the current procedure and the Sobel edge detector. A generalization of the procedure is also discussed in this section. In Section 3, statistical consistency of the detected jumps is discussed. A performance measurement of the detected jumps is defined in Section 4. In Section 5, several numerical examples are presented regarding the accuracy of the detected jumps and the selection of the procedure parameters. Finally, some remarks conclude the article in Section 6.

2 The Jump Detection Procedure

Suppose that the regression model concerned is

$$Z_i = f(x_i, y_i) + \varepsilon_i, \quad i = 1, 2, \cdots, n \tag{2.1}$$

where $\{Z_i\}$ are observations, $\{(x_i, y_i)\}$ are design points in design space Ω which is a connected region in R^2, $f(x, y)$ is a bivariate regression function which is continuous in Ω except at the JLCs, and $\{\varepsilon_i\}$ are i.i.d. errors with mean 0 and variance σ^2.

Let $K_1(x, y)$ and $K_2(x, y)$ be two non-negative kernel functions satisfying: (i) the support of $K_1(x, y)$ is $[-1/2, 1/2] \times [-1, 0]$ and the support of $K_2(x, y)$ is $[-1/2, 1/2] \times [0, 1]$; and (ii) $\int_{-1}^{1}\int_{-1}^{1} K_i(x, y)\, dxdy = 1$ for $i = 1$ and 2. Namely, the two kernel functions are one-sided density functions: $K_1(x, y)$ is lower-sided and $K_2(x, y)$ is upper-sided. Then we define

$$M_n^{(1)}(x, y) = \frac{1}{nh_n p_n} \sum_{i=1}^{n} Z_i \left[K_2(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) - K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) \right], \tag{2.2}$$

where $(x, y) \in \Omega \backslash O(\partial\Omega, b_n)$, $\partial\Omega$ denotes the boundary point set of Ω, $O(\partial\Omega, b_n)$ is the border region of Ω defined by $O(\partial\Omega, b_n) = \{s : d(s, \partial\Omega) \leq b_n \text{ for all } s \in \Omega\}$, $d(\cdot, \cdot)$ is the Euclidean distance, $b_n = \sqrt{h_n^2/4 + p_n^2}$, h_n and p_n are two window widths. Obviously, $M_n^{(1)}(x, y)$ is a difference of two weighted averages of the observations in the upper and lower sided neighborhoods of (x, y), respectively (cf. Figure 2.1 (a)).

Intuition tells us that the quantity $M_n^{(1)}(x,y)$ alone can detect jumps that are parallel to the x-axis well but it can not detect jumps that are perpendicular to the x-axis. To overcome this difficulty, Qiu (1997) introduced the rotational kernel (RK) functions

$$RK_i(\theta,x,y) = K_i\left(\delta(x)\sqrt{x^2+y^2}\cos(\arctan(y/x)-\theta), \delta(x)\sqrt{x^2+y^2}\sin(\arctan(y/x)-\theta)\right)$$

and then defined

$$RM_n(\theta,x,y) = \frac{1}{nh_np_n}\sum_{i=1}^{n} Z_i\left[RK_2(\theta,\frac{y_i-y}{h_n},\frac{x_i-x}{p_n}) - RK_1(\theta,\frac{y_i-y}{h_n},\frac{x_i-x}{p_n})\right], \tag{2.3}$$

where $\theta \in [-\pi/2,\pi/2]$ and $\delta(x)=1$ if $x \geq 0$ and 0 otherwise. The support of $RK_i(\theta,x,y)$ can be obtained by rotating the support of $K_i(x,y)$ an angle θ counterclockwise. The quantity $RM_n(\theta,x,y)$ is actually a difference of two weighted averages of the observations located in two neighborhoods on two different sides of the point (x,y) along the direction of $(\cos(\theta),\sin(\theta))$ (cf. Figure 2.1(b)). Then he suggested using

$$\max_{-\pi/2\leq\theta\leq\pi/2} |RM_n(\theta,x,y)| \tag{2.4}$$

as a jump detection criterion and the detected jumps by the criterion (2.4) were called the rotational difference kernel estimators (RDKE) of the JLCs.

The criterion (2.4) searches all directions at (x,y) for detecting a possible jump point. Theoretically it is appealing. But it is computationally intensive and hard to use in applications. To simplify its computation, we consider the x-axis and y-axis directions only at each point (x,y) in this paper. Similar to $M_n^{(1)}(x,y)$, we define

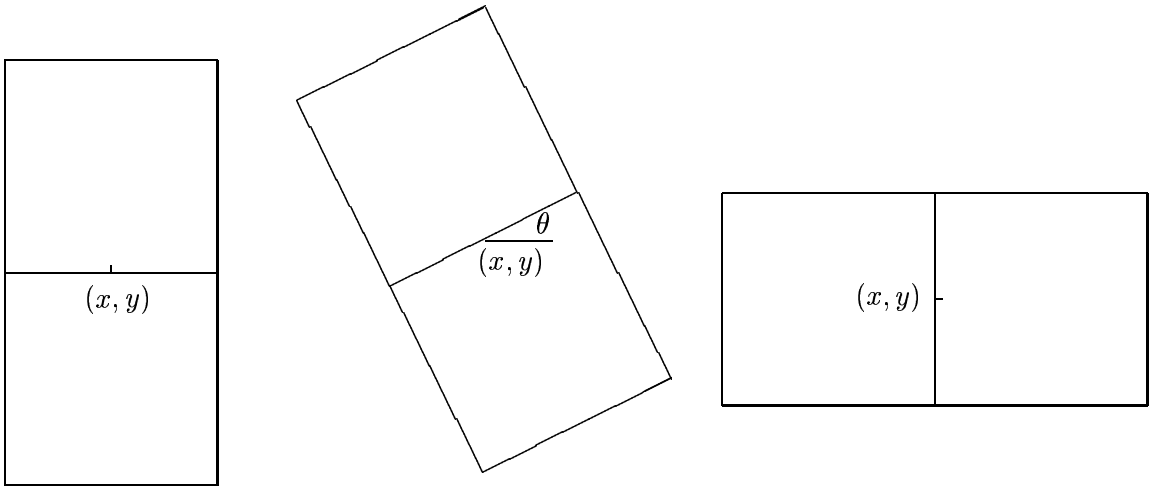
$$M_n^{(2)}(x,y) = \frac{1}{nh_np_n}\sum_{i=1}^{n} Z_i\left[K_2(\frac{y_i-y}{h_n},\frac{x_i-x}{p_n}) - K_1(\frac{y_i-y}{h_n},\frac{x_i-x}{p_n})\right]. \tag{2.5}$$

It can be seen that $M_n^{(2)}(x,y)$ is a difference of two weighted averages of the observations in the left and right sided neighborhoods of (x,y), respectively (cf. Figure 2.1 (c)). Then we define

$$M_n(x,y) = \max\left\{|M_n^{(1)}(x,y)|, |M_n^{(2)}(x,y)|\right\} \tag{2.6}$$

as a jump detection criterion and large value of $M_n(x,y)$ indicates a possible jump at (x,y). We then use the point set

$$\hat{D}_n := \{(x_i,y_i) : M_n(x_i,y_i) \geq u_n\} \tag{2.7}$$



(a) (b) (c)

Figure 2.1: (a) The upper and lower sided windows of (x, y) for constructing $M_n^{(1)}(x, y)$; (b) The windows in plot (a) are rotated counterclockwise by an angle θ; (c) the left and right sided windows of (x, y) for constructing $M_n^{(2)}(x, y)$.

to estimate the point set of the JLCs defined by $D := \{(x, y) : (x, y) \text{ is a point on the JLCs}\}$, where u_n is a positive threshold value. In Appendix A, a formula is derived for u_n:

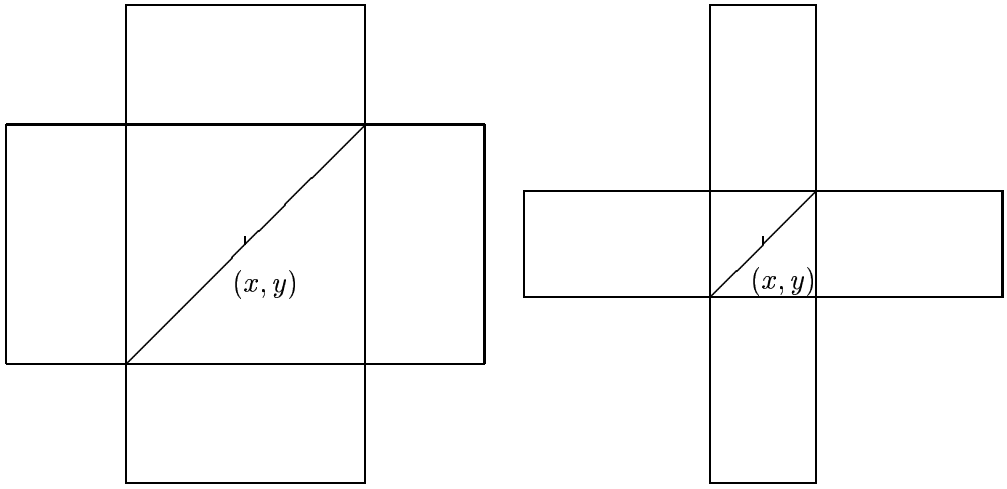
$$u_n = \sqrt{\frac{\sigma^2 \chi^2_{1,\alpha_n/2}}{n h_n p_n} [\int_{-1/2}^{1/2} \int_0^1 (K_2(x, y))^2 \, dxdy + \int_{-1/2}^{1/2} \int_{-1}^0 (K_1(x, y))^2 \, dxdy]}, \tag{2.8}$$

which has the property that when (x, y) is a continuous point of the regression surface and n is large enough,

$$Prob(M_n(x, y) > u_n) \leq \alpha_n,$$

where α_n is a significance level.

As mentioned above, the quantity $M_n^{(1)}(x, y)$ is good for detecting edges that are parallel to the x-axis. Similarly, $M_n^{(2)}(x, y)$ is good for detecting edges that are parallel to the y-axis. But neither one of them could be able to detect an arbitrary edge well as demonstrated by Figure 2.2(a). In Figure 2.2(a), the point (x, y) is on a JLC which is not parallel to the x-axis and the y-axis. If there are no restrictions on the bandwidths h_n and p_n, then both $M_n^{(1)}(x, y)$ and $M_n^{(2)}(x, y)$ could be small because observations on both sides of the JLC are averaged in computing either one of these two quantities and consequently the jump effect is mostly cancelled out. However, when the ratio h_n/p_n of the two window widths gets smaller (namely, the two neighborhoods get relatively narrower), both $M_n^{(1)}(x, y)$ and $M_n^{(2)}(x, y)$ become more capable to detect the JLC because relatively less



(a) (b)

Figure 2.2: (a) The point (x, y) is on a JLC which is not parallel to the x-axis and the y-axis; (b) same as plot (a) except that the neighborhoods are relatively narrower than those in plot (a).

observations on two different sides of the JLC are averaged in computing either one of them, which is demonstrated by Figure 2.2(b). Therefore, in order to detect an arbitrary JLC efficiently, the window widths h_n and p_n need to satisfy the assumption that

(A 2.1) $$\lim_{n\to\infty} h_n/p_n = 0$$

besides some regularity conditions given in Section 3. That is, the neighborhoods need to be narrower and narrower when n increases. This issue will be further demonstrated by numerical examples in Section 5.2.

As mentioned in Section 1, the Sobel operator is a "classic" edge detector in the image processing literature. It is based on two Sobel masks displayed in Figure 2.3. For a given design point (x_i, y_i), 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 by $\hat{f}_y^{(i)}$. Similarly, the second mask is used to obtain an estimator of the partial derivative with respect to x, which is denoted by $\hat{f}_x^{(i)}$. Then $[(\hat{f}_x^{(i)})^2 + (\hat{f}_y^{(i)})^2]^{1/2}$ is used as an edge detection criterion with large values indicating possible edges.

The quantity $\hat{f}_y^{(i)}$ is similar to our $M_n^{(1)}(x_i, y_i)$ defined in (2.2) in the sense that $\hat{f}_y^{(i)}$ is also a difference of two weighted averages along the y-axis direction. Similarly, the quantity $\hat{f}_x^{(i)}$ is related to $M_n^{(2)}(x_i, y_i)$. The Sobel edge detector uses the Euclidean length of the estimated gradient vector as its edge detection criterion. In (2.6), we suggest using $M_n(x_i, y_i)$, which is the maximum of

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

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

Figure 2.3: Sobel masks.

$|M_n^{(1)}(x_i, y_i)|$ and $|M_n^{(2)}(x_i, y_i)|$, to detect jumps based on the following consideration. When the point (x_i, y_i) is on a JLC, the jump structure of the regression surface contaminates some of the four kernel averages used in constructing $M_n(x_i, y_i)$ (cf. (2.6)) as estimators of the regression surface at (x_i, y_i). For example, if the JLC is parallel to the x-axis, then the two kernel averages used in constructing $M_n^{(1)}(x_i, y_i)$ estimate the surface well. Consequently, $M_n^{(1)}(x_i, y_i)$ is a good estimator of the jump magnitude at (x_i, y_i). But the two kernel averages used in constructing $M_n^{(2)}(x_i, y_i)$ do not provide much helpful information for jump detection. If $M_n^{(2)}(x_i, y_i)$ is included in the jump detection criterion as the Sobel edge detector does, it can only make the jump detection criterion noisier and thus have negative impact on jump detection. Such negative impact is mostly eliminated in our criterion (2.6) because most likely $M_n(x_i, y_i)$ equals $M_n^{(1)}(x_i, y_i)$ in the case mentioned above.

The conventional Sobel edge detector uses windows of size 3×3 only. With this small size, its ability to smooth away the noise and enhance the discontinuities is limited. This limitation is overcomed by our procedure because larger windows are allowed now. From the numerical examples in Section 5, we can see that larger windows do improve jump detection.

The jump detection procedure (2.6)-(2.8) searches the x-axis and y-axis directions only at each design point for a possible jump. This idea can be generalized by searching more than two directions as follows. Let $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_m < \pi$ be m directions in $[0, \pi)$, where $m \ge 2$ is an integer. At point (x, y), we define

$$M_n(x, y) := \max \{|RM_n(\theta_i, x, y)|, i = 1, 2, \cdots, m\}, \tag{2.9}$$

where $RM_n(\theta_i, x, y)$ is defined by equation (2.3), which is a difference of two averages of the observations in two one-sided neighborhoods along the direction of $(\cos(\theta_i), \sin(\theta_i))$. In applications, we suggest using equally spaced angles in $[0, \pi)$, namely $\theta_i = \frac{(i-1)\pi}{m}$ for $i = 1, 2, \cdots, m$, unless users have some prior information about the jump directions. By using the same arguments as those in

Appendix A, the threshold value for $M_n(x,y)$ in (2.9) can be calculated by:

$$u_n = \sqrt{\frac{\sigma^2 \chi^2_{1,\alpha_n/m}}{nh_n p_n}[\int_{-1/2}^{1/2}\int_0^1 (K_2(x,y))^2\,dxdy + \int_{-1/2}^{1/2}\int_{-1}^0 (K_1(x,y))^2\,dxdy]}. \tag{2.10}$$

It should be pointed out that formulas (2.8) and (2.10) are based on the Bonferroni correction (cf. (A.1) in Appendix A), which are quite conservative in the sense that the threshold values based on these formulas tend to be larger than what we would expect from the significance level α_n especially when m is large. To compensate for this, we suggest choosing α_n a little larger than its regular value in applications. More accurate threshold formulas might be possible by studying the correlation among $\{RM_n(\theta_i,x,y), i=1,2,\cdots,m\}$. We leave this topic to our future research.

When $m=2$, $\theta_1=0$ and $\theta_2=\pi/2$, (2.9) is identical to (2.6). On the other hand, when m is large enough such that $\{\theta_i\}_{i=1}^m$ are dense in $[0,\pi)$, the generalized jump detection procedure (2.9)-(2.10) is almost equivalent to the RDKE procedure suggested by Qiu (1997) (also see Müller and Song (1994) for a similar procedure). It is apparent that the value of m is related directly to the amount of computation required: the larger the value of m is, the more computation the procedure requires. In the next setion, we will show that m is also related to the conditions on the JLCs required by the procedure. If m is chosen larger, then the conditions for a point on the JLCs to be detected successfully by the procedure will be weaker and consequently more jump points can be detected by the procedure. So m is related to the tradeoff between the computational complexity of the procedure and its power for jump detection. In Section 5, this issue will be further discussed with numerical examples.

3 Strong Consistency of the Detected Jumps

We establish the almost sure consistency of the detected jumps in this section. Since the detected jumps and the true jumps are treated as two point sets in the design space by our procedure, a distance measure between two point sets G_1 and G_2 in R^2 is needed. In this article, the well-known Hausdorff distance is used, which is defined by

$$d_H(G_1,G_2) := \max\{\sup_{x\in G_1}\inf_{y\in G_2}\|x-y\|, \sup_{x\in G_2}\inf_{y\in G_1}\|x-y\|\}. \tag{3.1}$$

Throughout this article, we have the following assumption (A 3.1) on the design points:

(A 3.1) There exists a partition $\Lambda := \{\Delta_i, i=1,2,\cdots,n\}$ of the design space Ω such that

- $\bigcup_{i=1}^{n} \Delta_i = \Omega$, and $\Delta_i \bigcap \Delta_j = \emptyset$ if $i \neq j$;
- $(x_i, y_i) \in \Delta_i$, for $i = 1, 2, \cdots, n$;
- $\max_{1 \leq i \leq n} d_i = O(n^{-1/2})$, where d_i denotes the diameter of Δ_i;
- $\max_{1 \leq i \leq n} |S(\Delta_i) - 1/n| = O(n^{-1-\lambda})$, where $\lambda > 0$ is some constant and $S(\Delta_i)$ denotes the area of Δ_i.

Remark 3.1 The assumption (A 3.1) requires that the design points have some homogeneity. This kind of assumption is often used in multivariate nonparametric regression analysis (cf., e.g., Müller 1988, Chapter 6). It can be checked that the assumption (A 3.1) is satisfied automatically when the design points are equally spaced in Ω.

A point (x_0, y_0) is called a *nonsingular point* of the JLCs if it is on a JLC and satisfies the following three conditions:

(i) The jump magnitude of the regression surface is positive at (x_0, y_0);

(ii) (x_0, y_0) is not a cross point of two or more JLCs;

(iii) Suppose that the JLC on which (x_0, y_0) is located has a parametric expression $x = x(t)$ and $y = y(t)$ in a neighborhood of (x_0, y_0), where t denotes the curve length from point (x_0, y_0) to point $(x(t), y(t))$ (obviously, $(x_0, y_0) = (x(t_0), y(t_0))$ with $t_0 = 0$). Then the JLC satisfies the Lipschitz (1) condition at (x_0, y_0) in the sense that: either it can be expressed as $x(t) = \phi(y(t))$ in some neighborhood $N(t_0)$ of t_0 and there exists a constant $M_1 > 0$ such that

$$|x(t_1) - x(t_2)| \leq M_1 |y(t_1) - y(t_2)|, \text{ for any } t_1, t_2 \in N(t_0);$$

or it has an expression $y(t) = \psi(x(t))$ in $N(t_0)$ and there exists another constant $M_2 > 0$ such that

$$|y(t_1) - y(t_2)| \leq M_2 |x(t_1) - x(t_2)|, \text{ for any } t_1, t_2 \in N(t_0), \tag{3.2}$$

where $\phi(\cdot)$ and $\psi(\cdot)$ are two univariate functions.

If (x_0, y_0) is on a JLC and it is not a nonsingular point, then it is called a *singular point* of the JLCs.

Remark 3.2 The above condition (iii) is essentially the Lipschitz (1) condition expressed in a symmetric way with respect to the x and y axes.

From the above definition, if (x_0, y_0) is a nonsingular point of the JLCs, then it is on a single JLC and the JLC satisfies the Lipschitz condition around (x_0, y_0) as stated in condition (iii). Without loss of generality, suppose that the JLC satisfies the Lipschitz condition (3.2). In such case, the two neighborhoods used in constructing $M_n^{(1)}(x_0, y_0)$ (cf. Figure 2.1(a)) are mostly on two different sides of the JLC under the assumption of (A 2.1). The part of the upper-sided neighborhood located below the JLC and the part of the lower-sided neighborhood located above the JLC can both be ignored (cf. the discussion after equation (B.6) in Appendix B). Consequently $M_n^{(1)}(x_0, y_0)$ is a good estimator of the jump magnitude of the regression surface at (x_0, y_0) and this jump point is likely to be detected by our procedure. On the other hand, if (x_0, y_0) is a singular point of the JLCs, then it is either a crosspoint of several JLCs or a point on a single JLC which does not satisfy the Lipschitz condition around (x_0, y_0). In either case, the property of $M_n^{(1)}(x_0, y_0)$ (also $M_n^{(2)}(x_0, y_0)$) mentioned above may not be true. So the procedure (2.6)-(2.8) may not be able to detect the singular points.

Theorem 3.1 In model (2.1), suppose that the regression function $f(x, y)$ is Lipschitz (1) continuous in design space Ω except on the JLCs; the design points satisfy the assumption (A 3.1); $E|\varepsilon_1|^p < \infty$ for some $p \geq 2$. Besides the two conditions given in Section 2, $K_1(x, y)$ and $K_2(x, y)$ are assumed to be Lipschitz (1) continuous in their supports. The window widths h_n and p_n are assumed to satisfy the condition (A 2.1) and the conditions that: (i) $\frac{n^\nu}{\beta_n \log n}[p_n + \frac{1}{n^\lambda h_n p_n} + \frac{1}{\sqrt{n} h_n^2 p_n}] = o(1)$; (ii) $\frac{n^{2\nu}}{n h_n p_n \beta_n} = O(1)$; and (iii) $\frac{n^{\nu+1/p-1}}{\beta_n h_n p_n \log n} = o(1)$, where ν is a positive number and $\{\beta_n\}$ is a series of numbers satisfying $\lim_{n\to\infty} \beta_n = \infty$. The significance level α_n is assumed to satisfy: (i) $\frac{\chi^2_{1,\alpha_n/2}}{n h_n p_n} = o(1)$; and (ii) $\frac{(\beta_n \log n)^2 n h_n p_n}{n^{2\nu} \chi^2_{1,\alpha_n/2}} = O(1)$. Then $\lim_{n\to\infty} d_H(\hat{D}_n \bigcap \Omega_\rho, D \bigcap \Omega_\rho) = 0$, $a.s.$, with rate $O(p_n)$, where $\Omega_\rho := \{s : d(s, \partial\Omega) \geq \rho; d(s, s^*) \geq \rho; s \in \Omega;$ and s^* is some singular point of the JLCs$\}$ and $\rho > 0$ is any constant.

Remark 3.3 By checking the conditions in the above theorem, the convergence rate of $\lim_{n\to\infty} d(\hat{D}_n \bigcap \Omega_\rho, D \bigcap \Omega_\rho) = 0$, $a.s.$, could reach $o(n^{-1/6+\tau})$, for any $0 < \tau < 1/6$. The proof of Theorem 3.1 is given in Appendix B.

Theorem 3.1 says that excluding any small areas around the singular points of the JLCs and the boundary of the design space, the set of detected jumps converges almost surely to the set of true

jumps in Hausdorff distance. It should be noticed that this result requires the regression surface to be Lipschitz (1) continuous only in the design space except on the JLCs, which is reasonable for most applications. Theorem 3.1 also tells us that regions around the singular points of the JLCs and the boundary of the design space are the places where the procedure (2.6)-(2.8) may fail to detect jumps.

If the generalized procedure (2.9)-(2.10) is used for jump detection, then the definition of nonsingular points of the JLCs needs to be changed accordingly. More specifically, the condition (iii) in the original definition needs to be modified into the following condition (iii)':

(iii)' $\cdots$ Then there exists a neighborhood $N(t_0)$ of t_0 and an angle $\theta \in \{\theta_i\}_{i=1}^m$ such that in this neighborhood the JLC can be expressed by $y(t) = \phi^*(x(t))$ for some Lipschitz (1) continuous function $\phi^*(\cdot)$ after the coordinate system is rotated counterclockwise by θ.

It can be checked that the consistency property of the detected jumps stated in Theorem 3.1 is still true in this case after (2.6) and (2.8) are replaced by (2.9) and (2.10), respectively, and the condition (iii) in the original definition of nonsingular points is changed to the above condition (iii)'.

4 Performance Measurement of the Detected Jumps

We discuss performance measurement of the detected jumps in this section. Theoretically speaking, the Hausdorff distance (see its definition in Section 3) can be used for this purpose. In reality, this distance is hard to compute. Its computational complexity is $O(n^{3/2})$.

In this paper, we suggest an alternative performance measurement defined by:

$$d^*(\hat{D}_n, D) := w\frac{|\hat{D}_n \backslash D|}{|\Omega \backslash D|} + (1-w)\frac{|D \backslash \hat{D}_n|}{|D|} \tag{4.1}$$

where $0 \le w \le 1$ is a constant, $|A|$ denotes the number of design points in point set A, and D can be replaced by $D^* := \{(x_i, y_i) : d((x_i, y_i), D) \le 1/\sqrt{2n}\}$ for the purpose of calculating d^*. In (4.1), $|\hat{D}_n \backslash D|$ is the number of false jump detections and $|\Omega \backslash D|$ is the total number of non-jump points in the design space. Therefore $\frac{|\hat{D}_n \backslash D|}{|\Omega \backslash D|}$ is the proportion of the detected non-jump points to all non-jump points. Similarly, $\frac{|D \backslash \hat{D}_n|}{|D|}$ is the percentage of the true jump points missed by the jump

detection procedure. d^* is their weighted average. The weight w represents the relative importance of the first percentage and it needs to be specified by users. In the case that the two proportions are equally important, w can be simply chosen 0.5. In the two extreme cases that the JLCs are completely detected while there is no false jump detection and that all the non-jump points are detected while all the true jumps are missed, d^* equals 0 and 1, respectively. In a general case, d^* is between 0 and 1. The larger its value, the better the jump detection procedure performs and vise versa.

Suppose that there is a unique JLC in the design space $[0,1] \times [0,1]$. This JLC is a line parallel to the x-axis at $y = .5$. It is further assumed that there are two sets of detected jump points. The first set consists of the design points on line $y = .5$ and a point $(.8, .1)$ (Figure 4.1(a)). The second set consists of the design points on line $y = .2$ (Figure 4.1(b)). Then the Hausdorff distance between the true JLC and the detected jump points is .4 in the first case and .3 in the second case. Thus it can be concluded that the second set of detected jump points is better than the first set, which might be the opposite to what we would expect. The main reason behind this situation is that the Hausdorff distance is sensitive to individual points (the point $(.8, .1)$ in this case). In the definition of d^*, we use proportions instead of supremum/infimum values to make d^* more robust to individual points. It is not hard to check that d^* is close to zero in the case of Figure 4.1(a) and larger than .5 in the case of Figure 4.1(b). Another reason why we prefer d^* is its simple computation. It can be checked that its computational complexity is $O(n)$.

We should point out that the performance measurement d^* is not perfect for applications. In the example of Figure 4.1, 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^*, the performance does not depend on n. To measure the jump detection performance properly is an important topic because it is related directly to comparisons of different jump detection procedures. We hope there will be more future research to address this issue.

5 Numerical Study

In this section, we present some numerical results in two parts. In Section 5.1, numerical performance of the jump detection procedure (2.6)-(2.8) is evaluated. Selection of m of the generalized procedure (2.9)-(2.10) is also discussed. Then the assumption (A 2.1) is discussed with numerical

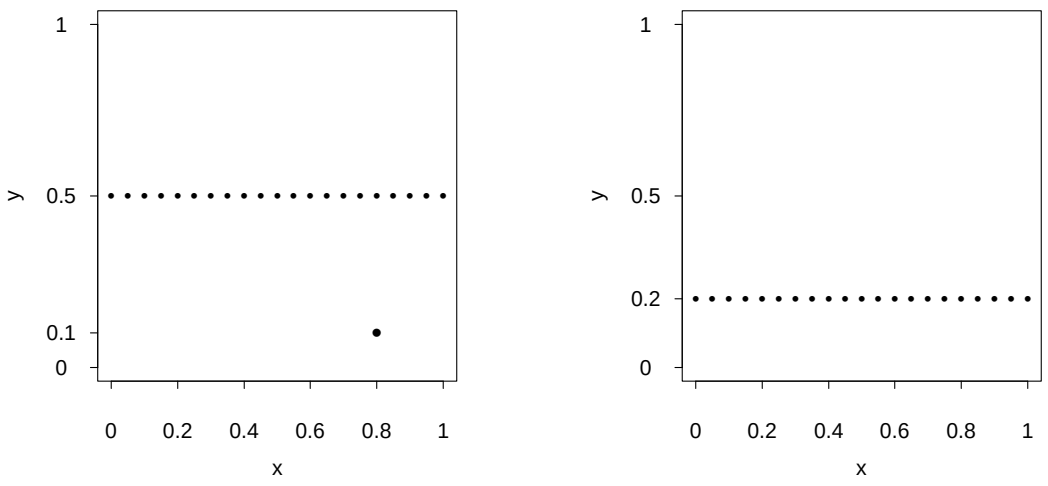


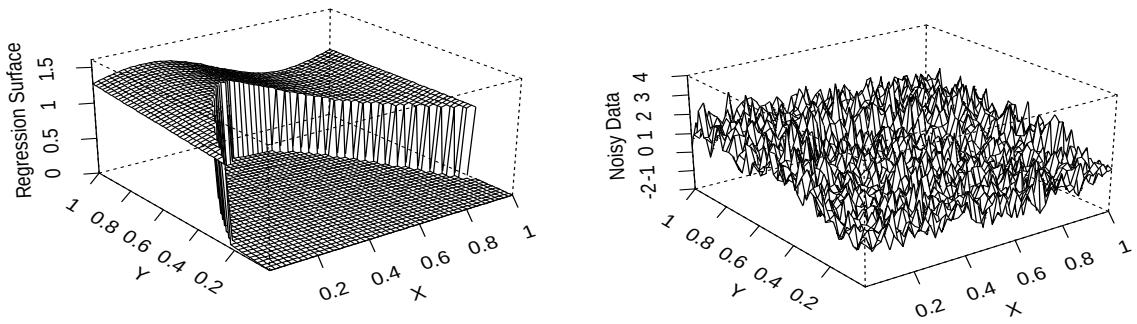
Figure 4.1: Suppose that the true JLC is the line $y = .5$. The set of detected jump points consists of the design points on line $y = .5$ and a point $(.8, .1)$ in plot (a) and of the design points on line $y = .2$ in plot (b).

examples in Section 5.2.

5.1 Numerical Performance of the Jump Detection Procedure

We first consider the regression function $f(x, y) = \frac{1}{4}(1 - x)y + [1 + .2\sin(2\pi x)]I_{y \geq .6\sin(\pi x)+.2}$, for $(x, y) \in [0, 1] \times [0, 1]$ which has a unique JLC $\phi(x) = .6\sin(\pi x) + .2$ with jump magnitude $1 + .2\sin(2\pi x)$. Observations are generated from model (2.1) with $\varepsilon_1 \sim N(0, \sigma^2)$ at design points $(x_i, y_j) = (i/n_1, j/n_1)$, for $i, j = 1, 2, \cdots, n_1$. The sample size is $n = n_1^2$. We define $K_2(x, y) = \frac{12}{11}(1 - x^2) \cdot \frac{12}{11}(1 - (y - .5)^2)I_{[-1/2,1/2]\times[0,1]}$, which is a product of two Epanechnikov kernel functions (see e.g., Härdle 1991, page 45), and $K_1(x, y) = K_2(x, -y)$. The significance level is fixed at $\alpha_n = .01$. Without loss of generality, the window widths h_n and p_n are assumed to take the values of $h_n = k_1/n_1$ and $p_n = k_2/n_1$ where k_1 and k_2 are two positive integer numbers. Because the supports of the kernel functions $K_1(x, y)$ and $K_2(x, y)$ are symmetric about 0 with respect to x (cf. Figure 2.1(a)), only odd integer numbers need to be considered for k_1 to detect jumps at the design points. For convenience, we only consider odd integer numbers for k_2 too. Without confusion, k_1 and k_2 are sometimes called window widths in this section. Figure 5.1 shows the true regression surface (plot (a)) and its noisy version with $\sigma = .5$ (plot (b)).

We first fix $n_1 = 100, \sigma = .5$ and let k_1 and k_2 change. Values of the performance measurement



(a) (b)

Figure 5.1: (a) The true regression surface; (b) a noisy version of the regression surface with $\sigma = .5$.

Table 5.1: Values of the performance measurement d^* with several pairs of window widths.

	k_1				
k_2	3	5	7	9	11
3	0.40227	0.28684	0.15510	0.18795	0.16237
5	0.01147	0.01142	0.01723	0.02177	0.02564
7	0.01497	0.02819	0.03683	0.04259	0.04749
9	0.03052	0.04641	0.05718	0.06504	0.07057
11	0.04629	0.06792	0.07792	0.08720	0.08976

d^* with several pairs of window widths are presented in Table 5.1 (the value of w used in the definition of d^* is chosen to be 0.5). To eliminate some randomness, all results presented in this section are averages of 100 replications. As Table 5.1 indicated, the window widths should not be too small or too large. When they are too small, the jump detection criterion $M_n(x, y)$ is still quite noisy, making the threshold value u_n relatively large (cf. (2.8)). Consequently some real jump points would be missed. On the other hand, when they get larger, more design points have their jump detection criterion values affected by the jump structure of the regression surface. Consequently, there would be more false jump detections, making the detected JLC thick (see Figure 5.2(b) and the related discussion given below). The best pair of window widths in Table 5.1 is $(k_1, k_2) = (5, 5)$.

Figure 5.2(b) shows the detected jump points of the procedure (2.6)-(2.8) with window widths $(k_1, k_2) = (5, 5)$. We notice that there are two kinds of deceptive jump candidates. The first kind is those scattered in the design space due to the nature of hypothesis testing on which the threshold value is based. The second kind of deceptive jump candidates are those around the true

JLC due to the nature of local smoothing. Qiu and Yandell (1997) suggested two modification procedures to delete these two kinds of candidates. Figures 5.2(c) and 5.2(d) show the results after the modification procedures are sequentially applied to the results in plot (b). As a comparison, the true JLC is plotted in Figure 5.2(a).

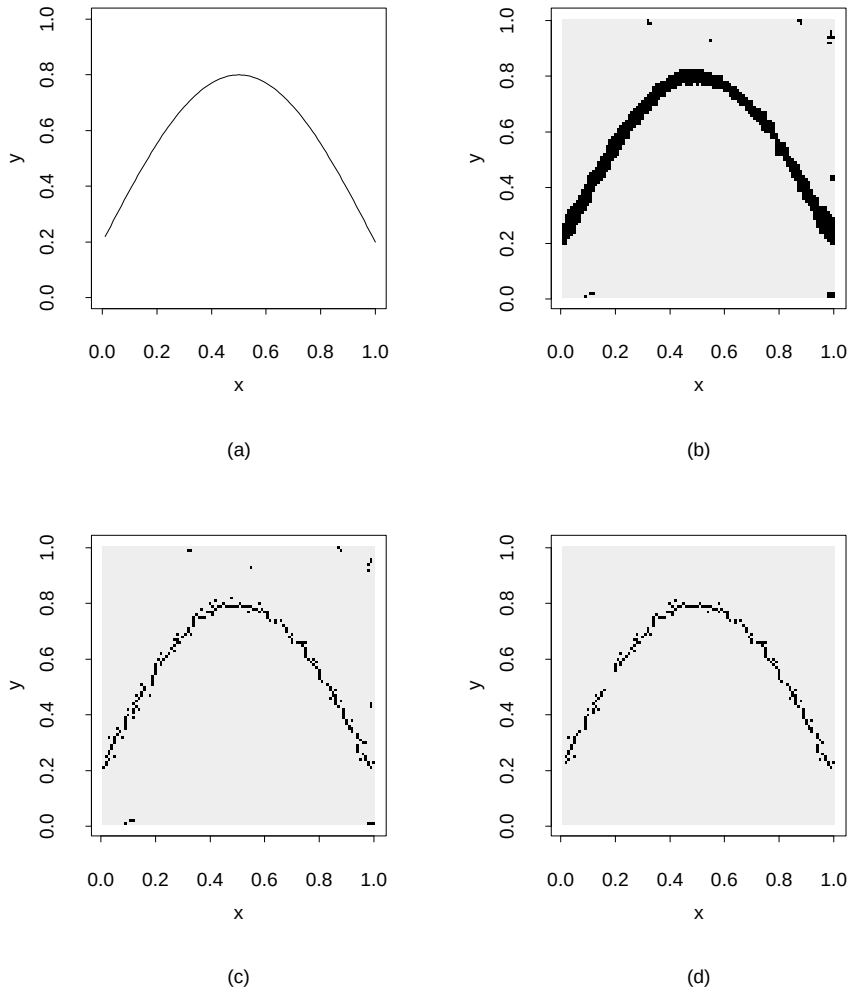


Figure 5.2: (a) The true jump location curve; (b) the detected jump points by the procedure (2.2)-(2.4); (c) the modified version of plot (b) by the first modification procedure in Qiu and Yandell (1997) (to make the detected JLC thinner); (d) the modified version of plot (c) by the second modification procedure in Qiu and Yandell (1997) (to delete some scattered jump candidates). In this example, $n = 100, \sigma = .5, (k_1, k_2) = (5, 5)$ and $\alpha_n = .01$.

The above simulation is then repeated with several different values of n_1 and σ. For each combination of n_1 and σ, the best pair of window widths and the corresponding d^* value are presented in Table 5.2. From the table, it can be seen that: (1) d^* gets smaller when n_1 is chosen larger, which may reflect the consistency of the detected jump points; (2) the window widths should be chosen larger if the data is noisier (namely, σ is bigger); (3) the window widths should

Table 5.2: For each combination of n_1 and σ, the best pair of window widths (k_1, k_2) and the corresponding d^* value (in parenthesis) are presented.

	n_1			
σ	100	200	300	400
0.25	3,3 (0.00961)	3,3 (0.00469)	3,3 (0.00301)	3,3 (0.00233)
0.50	5,5 (0.01142)	5,5 (0.00564)	5,5 (0.00370)	5,5 (0.00280)
0.75	7,7 (0.01422)	7,7 (0.00701)	7,7 (0.00450)	7,7 (0.00321)
1.0	9,9 (0.01880)	9,9 (0.00784)	9,9 (0.00540)	9,9 (0.00430)

be quite stable when the sample size changes. Theorem 3.1 tells us that the convergence rate of the detected jumps is $O(p_n)$. Therefore the window widths k_1 and k_2 should be stable to achieve the fastest convergence rate when the sample size increases. Our simulation results confirm this conclusion. From Table 5.2, it seems that k_1 and k_2 should equal each other, which conflicts with the assumption (A 2.1). This issue will be explained in Section 5.2.

Next, the generalized jump detection procedure (2.9)-(2.10) is used to detect jumps. Some parameters are chosen to be: $n_1 = 100$, $\sigma = .5$ and $(k_1, k_2) = (5, 5)$. Simulations are performed with 10 different m values: 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20. The amount of computation each simulation requires is roughly proportional to the value of m. The performance measurement values are plotted in Figure 5.3. This plot indicates that jump detection improves as more directions are searched at each design point. But it spends more computing time at the same time.

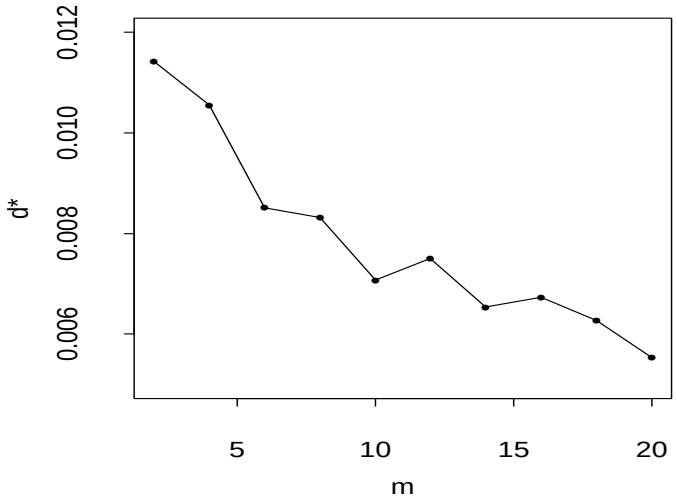


Figure 5.3: Simulation results of the generalized jump detection procedure (2.9)-(2.10).

At the end of this part, we apply our procedure to a log-transformed C-band, HH-polarization, synthetic aperture radar image of an area near Thetford forest, England. This image was introduced in some detail by Glasbey and Horgan (1995) and the data can be obtained from ftp://peipa.essex.ac.uk/ipa/pix/books/glasbey-horgan/.

The original image is presented in Figure 5.4(a). It has 250×250 pixels. By a preliminary local linear kernel fitting, an estimate of σ is 16.953. In the procedure (2.9)-(2.10), $(k_1, k_2) = (5, 5)$ and $\alpha_n = .01$ as before. The value of m is eithor 2 or 4 (when $m = 2$, the procedure (2.9)-(2.10) is the same as the procedure (2.6)-(2.8)). The detected jumps are presented in Figures 5.4(b) and 5.4(c), respectively, for $m = 2$ and 4. The results of Figure 5.4(c) are then modified by the two modification procedures of Qiu and Yandell (1997) and the modified results are plotted in Figure 5.4(d). We can see that most edges are detected well by our procedure. If we compare Figures 5.4(b) and 5.4(c) carefully, we can see that the procedure with $m = 4$ detect edges slightly better than the procedure with $m = 2$ (see e.g., the middle-left parts of the plots).

5.2 Discussion of Assumption (A 2.1)

From Table 5.2, it seems that the assumption (A 2.1) is not necessary for the procedure to detect jumps efficiently. In this part, we discuss this issue in some detail.

Let us first recall where this assumption is used in Theorem 3.1. From its proof given in Appendix B, this assumption is used in equations (B.6) and (B.7). Without loss of generality, we assume that a JLC has an expression $y = \psi(x)$ in a neighborhood of a given point (x, y) as shown in Figure 5.5 by the dotted curve. The rectangle in the plot represents the support of $K_2(\frac{u-x}{h_n}, \frac{v-y}{p_n})$ as a function of (u, v). Then the ratio of the area of region I (which is the part of the rectangle below the JLC) to the area of the entire rectangle is of order $O(h_n/p_n)$ because the area of region I is of order $O(h_n^2)$ by the Lipschitz (1) condition of the JLC and the area of the entire rectangle is $h_n p_n$. The assumption (A 2.1) basically says that region I is negligible comparing to the entire rectangle. Consequently, $M_n^{(1)}(x, y) \approx C(x, y)$, where $C(x, y)$ is the jump magnitude of the regression surface at (x, y). If $h_n = O(p_n)$ instead, then $M_n^{(1)}(x, y) \approx \gamma C(x, y)$, where γ is a constant between 0 and 1. The value of γ depends on the curvature of the JLC at (x, y) and on the ratio h_n/p_n as well. In the case of Figure 5.1, it can be checked that γ is larger than .5 when n is large enough. Therefore the assumption (A 2.1) is not critical in that case. Actually when we know that the JLC is very smooth

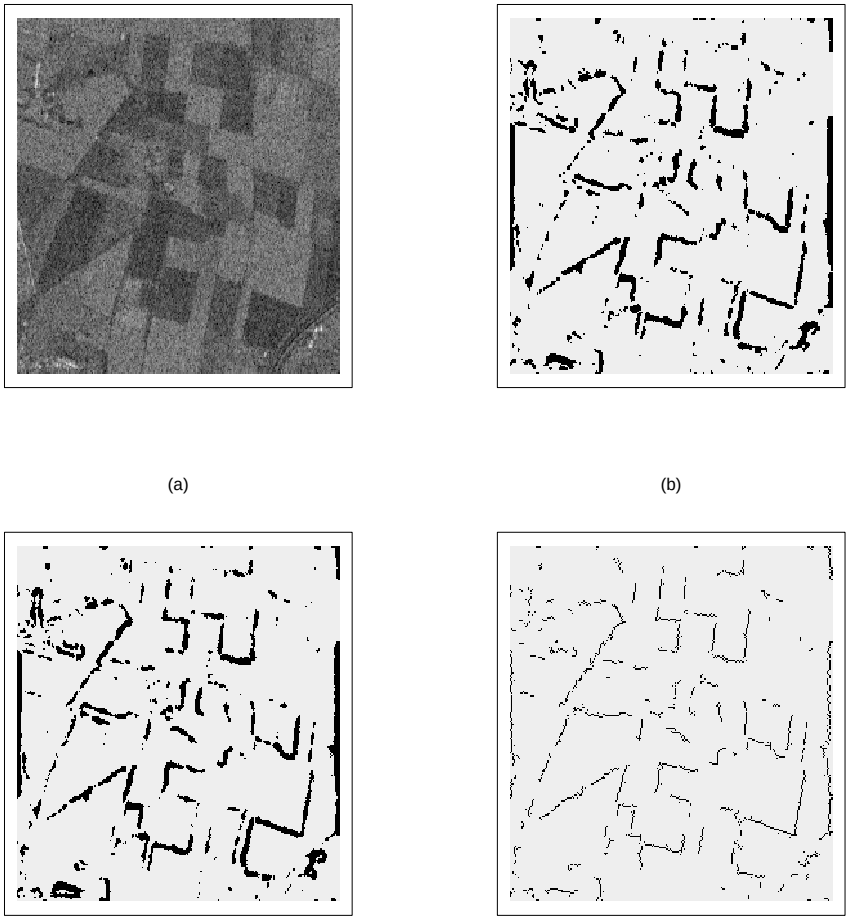


Figure 5.4: (a) The original image; (b) the detected edges by the procedure (2.6)-(2.8); (c) the detected edges by the procedure (2.9)-(2.10) with $m = 4$; (d) the results in plot (c) modified by the modification procedures of Qiu and Yandell (1997).

("smooth" means small curvature here), then h_n can be chosen even larger than p_n to make the detected JLC thin. However, this kind of window widths are not appropriate for detecting jumps at places where the JLCs are not smooth enough. This is demonstrated by the following example.

Figure 5.6(a) shows a true JLC which has a sharp angle at point $(.5, .5)$. The JLC has an expression $y = (1 - c)/2 + cx$ when $(c - 1)/(2c) \le x \le .5$ and $y = (1 + c)/2 - cx$ when $.5 < x \le (c + 1)/(2c)$, where $c = 5$. Observations are obtained at regularly spaced design points as we did before and $\sigma = .5$. When $n_1 = 100$ and $(k_1, k_2) = (5, 5)$, a set of detected jump points by the procedure (2.6)-(2.8) is presented in Figure 5.6(b). It can be seen that jump detection around the point $(.5, .5)$ is not good. To further investigate the jump detection at this point,

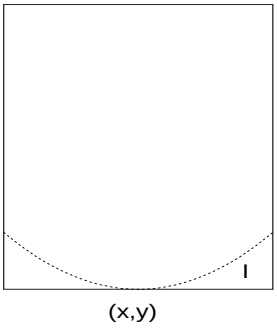


Figure 5.5: The rectangle represents the support of the kernel function $K_2(\frac{u-x}{h_n}, \frac{v-y}{p_n})$ as a function of (u, v). The dotted curve is a JLC on which the point (x, y) is located. "I" denotes the part of the rectangle below the JLC.

we next concentrate on the cross section of $x = .5$. The performance measurement value is still calculated by formula (4.1), but all quantities in the formula are computed from the design subspace $\{(x, y) : x = .5, 0 \leq y \leq 1\}$. Table 5.3 presents the best pairs of window widths along with the corresponding performance measurement values for several combinations of n_1 and c.

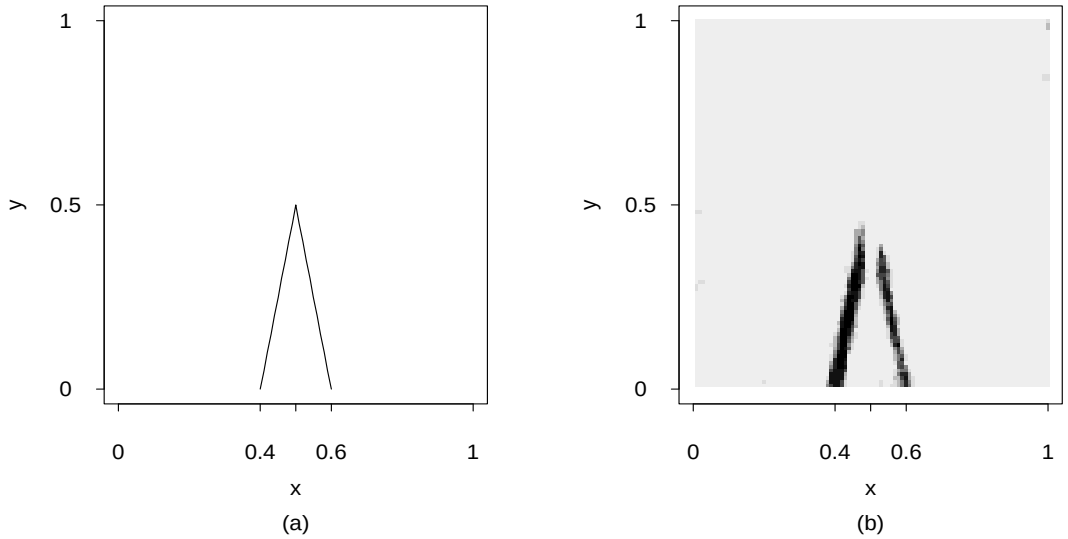


Figure 5.6: (a) The true jump location curve; (b) the detected jump points.

Table 5.3 confirms two points. First, the window widths (k_1, k_2) should be quite stable when n_1 increases as we found in Table 5.2. Second, the ratio k_1/k_2 should be smaller at places where the JLC has larger curvatures (namely, c is larger in this example), which explains the purpose of the assumption (A 2.1). That is, the assumption (A 2.1) is necessary for the procedure (2.6)-(2.8) to detect successfully all nonsingular points of the JLCs at which the JLCs may have different

Table 5.3: The best pairs of window widths (k_1, k_2) and the corresponding d^* values (in parentheses) for several combinations of n_1 and c.

	n_1			
c	100	200	300	400
1	3,11 (0.03776)	3,11 (0.01818)	3,11 (0.01141)	3,11 (0.00804)
3	3,11 (0.03827)	3,11 (0.01843)	3,13 (0.01560)	3,13 (0.00992)
5	3,13 (0.05204)	3,13 (0.02601)	3,15 (0.01980)	3,15 (0.01407)
7	3,15 (0.07245)	3,15 (0.03485)	3,15 (0.02013)	3,15 (0.01570)

curvatures.

6 Concluding Remarks

We have presented a jump detection procedure based on local smoothing techniques. This procedure simplifies the computation of some existing kernel-type methods in the statistical literature and makes their model assumptions more flexible as well. It can be regarded as a generalization of the Sobel edge detector in the image processing literature in the sense that the current procedure can use windows of size bigger than 3×3 in a way that the detected edges are statistically consistent.

We are currently concerned about the following issues. First, variable window widths might be more reasonable than the fixed ones. In some planar regions of the surface, we could use large-size windows whereas in highly textured regions, the window sizes could be small. Second, as pointed out in Section 4, the performance measurement d^* can still be improved. Third, some parameters in the procedure (such as k_1, k_2 and α_n) have not been well defined yet although some large sample conditions and numerical results are provided. Finally, we have investigated the Sobel edge detector in some detail. The relationship between the jump detection procedure (2.6)-(2.8) and other edge detectors in the image processing literature needs to be further studied.

Acknowledgements I would like to thank the associate editor and a referee for many helpful suggestions. Dr. Jörg Polzehl introduced me the synthetic aperture radar (SAR) image used in Section 5. This research was supported in part by a Grant-in-Aid of Research, Artistry and Scholarship of University of Minnesota.

Appendix

A Derivation of the Threshold Value u_n in (2.8)

First of all,

$$\begin{aligned} & P(M_n(x,y) > u_n) \\ = \; & P(|M_n^{(1)}(x,y)| > u_n \text{ or } |M_n^{(2)}(x,y)| > u_n) \\ \leq \; & P((M_n^{(1)}(x,y))^2 > u_n^2) + P((M_n^{(2)}(x,y))^2 > u_n^2). \end{aligned} \tag{A.1}$$

It is not hard to check that both $M_n^{(1)}(x,y)$ and $M_n^{(2)}(x,y)$ are approximately normally distributed with mean 0 and variance σ_n^2, where

$$\begin{aligned} \sigma_n^2 &= Var(M_n^{(1)}(x,y)) \\ &= \frac{\sigma^2}{(nh_np_n)^2}\sum_{i=1}^{n}[K_2(\frac{x_i-x}{h_n},\frac{y_i-y}{p_n}) - K_1(\frac{x_i-x}{h_n},\frac{y_i-y}{p_n})]^2 \\ &\approx \frac{\sigma^2}{nh_np_n}\int\int_{R^2}[K_2(x,y) - K_1(x,y)]^2\,dxdy \\ &= \frac{\sigma^2}{nh_np_n}[\int_{-1/2}^{1/2}\int_0^1 [K_2(x,y)]^2\,dxdy + \int_{-1/2}^{1/2}\int_{-1}^0 [K_1(x,y)]^2\,dxdy], \end{aligned} \tag{A.2}$$

and " $\approx$ " means that a high order term has been neglected. Therefore if we choose u_n by (2.8), then by (A.1) and (A.2),

$$P(M_n(x,y) > u_n) \leq P((M_n^{(1)}(x,y))^2/\sigma_n^2 > \chi^2_{1,\alpha_n/2}) + P((M_n^{(2)}(x,y))^2/\sigma_n^2 > \chi^2_{1,\alpha_n/2}) \approx \alpha_n.$$

B Proof of Theorem 3.1

The proof is divided into several parts. In the first part, we prove that if the regression function $f(x,y)$ is Lipschitz (1) continuous in the design space Ω, then

$$\frac{n^\nu}{\beta_n \log n}\|\hat{f}_n(x,y) - f(x,y)\|_\Omega = o(1), \quad a.s., \tag{B.1}$$

where $\hat{f}_n(x,y)$ is one of the four kernel estimators involved in the construction of $M_n(x,y)$ and $\|f(x,y)\|_\Omega$ denotes $\max\{|f(x,y)| : (x,y) \in \Omega\}$. Without loss of generality, we assume that $\hat{f}_n(x,y) = \frac{1}{nh_np_n}\sum_{i=1}^n Z_i K_1(\frac{x_i-x}{h_n},\frac{y_i-y}{p_n})$.

To prove (B.1), some arguments from the proof of Theorem 3 in Cheng and Lin (1981) can be borrowed, which proved a one-dimensional version of (B.1). Next, we give an outline of the proof of (B.1) in three steps.

Step 1 By the Lipschitz (1) condition on $f(x,y)$ and $K_1(x,y)$, it is not hard to check that

$$\|E\hat{f}_n(x,y) - f(x,y)\|_\Omega = O(p_n) + O(h_n) + O(\frac{1}{n^\lambda h_n p_n}) + O(\frac{\sqrt{h_n^2+p_n^2}}{\sqrt{n}h_n^2 p_n^2})$$

Step 2 Let

$$\begin{aligned}
\bar{\varepsilon}_i &= \varepsilon_i I_{|\varepsilon_i| \le i^{1/p}}, i = 1, 2, \cdots, n \\
\bar{g}_n(x,y) &= \frac{1}{nh_n p_n}\sum_{i=1}^n \bar{\varepsilon}_i K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) \\
g_n^*(x,y) &= \frac{1}{nh_n p_n}\sum_{i=1}^n \varepsilon_i K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}).
\end{aligned}$$

Without loss of generality, we assume that $\Omega \subset [0,1]\times[0,1]$. Let $A_n = \{(i/[n^\eta], j/[n^\zeta]) : i = 1, 2, \cdots, [n^\eta], j = 1, 2, \cdots, [n^\zeta]\} \bigcap \Omega$, where η and ζ are two positive constants and $[x]$ denotes the integer part of x. Then for any $(x,y) \in \Omega$, there exists $(v(x), w(y)) \in A_n$ such that $|x - v(x)| \le 1/[n^\eta]$ and $|y - w(y)| \le 1/[n^\zeta]$. It is obvious that

$$\frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(x,y) - E\bar{g}_n(x,y)\|_\Omega \le S_{1n} + S_{2n} + S_{3n},$$

where

$$\begin{aligned}
S_{1n} &= \frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(x,y) - \bar{g}_n(v(x), w(y))\|_\Omega \\
S_{2n} &= \frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(v(x), w(y)) - E\bar{g}_n(v(x), w(y))\|_\Omega \\
S_{3n} &= \frac{n^\nu}{\beta_n \log n}\|E\bar{g}_n(v(x), w(y)) - E\bar{g}_n(x,y)\|_\Omega.
\end{aligned}$$

By using some similar arguments to those in Cheng and Lin (1981), we can prove that all S_{1n}, S_{2n} and S_{3n} converge to 0 almost surely. Therefore

$$\frac{n^\nu}{\beta_n \log n}\|\bar{g}_n(x,y) - E\bar{g}_n(x,y)\|_\Omega = o(1), \quad a.s.$$

Step 3 Obviously,

$$\begin{aligned}
&\|\hat{f}_n(x,y) - E\hat{f}_n(x,y)\|_\Omega = \|g_n^*(x,y)\|_\Omega \\
\le\ &\|g_n^*(x,y) - \bar{g}_n(x,y)\|_\Omega + \|\bar{g}_n(x,y) - E\bar{g}_n(x,y)\|_\Omega + \|E\bar{g}_n(x,y)\|_\Omega.
\end{aligned}$$

By the fact that $K_1(x,y)$ is a bounded function, it is not difficult to prove that

$$\frac{n^\nu}{\beta_n \log n}\|g_n^*(x,y) - \bar{g}_n(x,y)\|_\Omega = o(1), \quad a.s.,$$

$$\frac{n^\nu}{\beta_n \log n}\|E\bar{g}_n(x,y)\|_\Omega = o(1).$$

The equation (B.1) is proved after combining the above three steps.

Next, we define

$$O_n(x,y) = \left\{(x',y') : \sqrt{(x'-x)^2 + (y'-y)^2} \le b_n \text{ for } (x',y') \in \Omega \backslash O(\partial\Omega, b_n)\right\},$$

$$D_n = \bigcup_{(x,y)\in D} O_n(x,y).$$

By (B.1), it is obvious that

$$\frac{n^\nu}{\beta_n \log n}\|M_n(x,y)\|_{\Omega\backslash(O(\partial\Omega,b_n)\bigcup D_n)} = o(1), \quad a.s.$$

So when n is large enough, the following expression is true for any $\rho > 0$:

$$\hat{D}_n \bigcap \Omega_\rho \subset D_n \bigcap \Omega_\rho, \quad a.s. \tag{B.2}$$

We now assume that the point (x,y) is on a JLC and it is a nonsingular point. Without loss of generality, the JLC is assumed to have a function expression $y = \psi(x)$ and $\psi(\cdot)$ satisfies the Lipschitz (1) condition in a neighborhood of (x,y). Clearly,

$$\begin{aligned}
& \frac{1}{nh_np_n}\sum_{i=1}^n Z_iK_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= \; & \frac{1}{nh_np_n}(\sum{}' + \sum{}'')Z_iK_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= \; & \frac{1}{nh_np_n}[\sum{}' Z_i + \sum{}''(Z_i + C(x,y))]K_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) - \\
& \frac{1}{nh_np_n}\sum{}'' C(x,y)K_2(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n})
\end{aligned} \tag{B.3}$$

where $\sum'$ denotes summation of the terms of which the design points are on one side (denoted as side 1) of the JLC and $\sum''$ denotes summation of the remaining terms of which the design points are on the other side of the JLC (denoted as side 2). In (B.3), without loss of generality, we also assume that the jump from side 2 to side 1 is positive with jump magnitude $C(x,y)$. Similarly,

$$\begin{aligned}
& \frac{1}{nh_np_n}\sum_{i=1}^n Z_iK_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) \\
= \; & \frac{1}{nh_np_n}[\sum{}'(Z_i - C(x,y)) + \sum{}'' Z_i]K_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n}) + \\
& \frac{1}{nh_np_n}\sum{}' C(x,y)K_1(\frac{x_i-x}{h_n}, \frac{y_i-y}{p_n})
\end{aligned} \tag{B.4}$$

By (B.1),

$$\begin{aligned} & \frac{1}{nh_np_n}[\sum{}' Z_i + \sum{}'' (Z_i + C(x,y))]K_2(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) - \\ & \frac{1}{nh_np_n}[\sum{}' (Z_i - C(x,y)) + \sum{}'' Z_i]K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) \\ = \quad & C(x,y) + o(\frac{\beta_n \log n}{n^\nu}), \quad a.s. \end{aligned} \tag{B.5}$$

On the other hand, we can check that the area of the intersection of the support of $K_2(\frac{u-x}{h_n}, \frac{v-y}{p_n})$ with the side 2 of the JLC is at most $M^*(x,y)h_n^2/2$, where $M^*(x,y) > 0$ is a constant, because the JLC is Lipschitz (1) continuous at (x,y). Therefore

$$\begin{aligned} & \frac{1}{nh_np_n} \sum{}'' C(x,y)K_2(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) \\ = \quad & \frac{M^*(x,y)h_n^2/2}{h_np_n} \cdot \frac{1}{nM^*(x,y)h_n^2/2} \sum{}'' C(x,y)K_2(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) \\ = \quad & O(h_n/p_n) \end{aligned} \tag{B.6}$$

Similarly,

$$\frac{1}{nh_np_n} \sum{}' C(x,y)K_1(\frac{x_i - x}{h_n}, \frac{y_i - y}{p_n}) = O(h_n/p_n) \tag{B.7}$$

By (B.3)-(B.7), we have

$$M_n^{(1)}(x,y) = C(x,y) + o(\frac{\beta_n \log n}{n^\nu}) + O(h_n/p_n), \quad a.s.$$

By the fact that $\min_{(x,y)\in D\bigcap\Omega_\rho} M^*(x,y) > 0$ and $\min_{(x,y)\in D\bigcap\Omega_\rho} C(x,y) > 0$, it can be checked that the above equation is uniformly true for $(x,y) \in D\bigcap\Omega_\rho$. So when n is large enough,

$$\hat{D}_n\bigcap\Omega_\rho \supset D\bigcap\Omega_\rho, \quad a.s. \tag{B.8}$$

By (B.2) and (B.8),

$$d_H(\hat{D}_n\bigcap\Omega_\rho, D\bigcap\Omega_\rho) \le d_H(D_n\bigcap\Omega_\rho, D\bigcap\Omega_\rho) \le b_n = \sqrt{h_n^2/4 + p_n^2}, \quad a.s.,$$

which is the conclusion of the theorem.

References

Bhandarkar, S.M., Zhang, Y., and Potter, W.D. (1994), "An edge detection technique using genetic algorithm-based optimization," *Pattern Recognition 27,* 1159-1180

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

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.

Cheng, K.F., and Lin, P.E. (1981), "Nonparametric estimation of a regression function," *Z. Wahrscheinlichkeitstheorie view. Gebiete 57,* 223-233.

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.

Glasbey, C., and Horgan, G. (1995), *Image Analysis for the Biological Sciences*, John Wiley & Sons: New York.

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.

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

Hall, P., and Raimondo, M. (1997), "Approximating a line thrown at random onto a grid," *The Annals of Applied Probability 7,* 648-665.

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.

Härdle, W. (1991), *Smoothing Techniques: with implementation in S*, New York: Springer-Verlag.

Korostelev, A.P., and Tsybakov, A.B. (1993), *Minimax Theory of Image Reconstruction*, Lecture Notes in Statistics, Vol. 82, Springer, New York.

Marr, D., and Hildreth, E. (1980), "Theory of edge detection," *Proceedings of the Royal Society of London 207*, 187-217.

Müller, H.G. (1988), *Nonparametric Regression Analysis of Longitudinal Data,* Lecture Notes in Statistics, Springer-Verlag: New York.

Müller, H.G., and Song, K.S. (1994), "Maximin estimation of multidimensional boundaries," *Journal of the Multivariate Analysis 50*, 265-281.

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.

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.

Qiu, P. (1997), "Nonparametric estimation of jump surface," *Sankhyā (Series A) 59*, 268-294.

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

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

Qiu, P., and Yandell, B. (1997), "Jump detection in regression surfaces," *Journal of Computational and Graphical Statistics 6,* 332-354.

Rosenfeld, A., and Kak, A.C. (1982), *Digital Picture Processing,* Vols. 1 and 2, Academic Press, New York.

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.

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.

Wang, Y. (1998), "Change curve estimation via wavelets," *Journal of the American Statistical Association 93,* 163-172.