Jump Detection in Regression Surfaces

Peihua Qiu; Brian Yandell

Journal of Computational and Graphical Statistics, Vol. 6, No. 3. (Sep., 1997), pp. 332-354.

Stable URL:
http://links.jstor.org/sici?sici=1061-8600%28199709%296%3A3%3C332%3AJDIRS%3E2.0.CO%3B2-P

Journal of Computational and Graphical Statistics is currently published by American Statistical Association.

Jump Detection in Regression Surfaces

Peihua QIU and Brian YANDELL

We consider the problem of locating jumps in regression surfaces. A jump detection algorithm is suggested based on local least squares estimation. This method requires $O(Nk)$ computations, where N is the sample size and k is the window width of the neighborhood. This property makes it possible to handle large data sets. The conditions imposed on the jump location curves, the jump surfaces, and the noise are mild. We demonstrate this method in detail with some numerical examples.

Key Words: Image processing; Jump detection criterion; Jump location curves; Jump surfaces; Least squares plane; Modification procedure; Slopes; Threshold value.

1. INTRODUCTION

In computer image analysis, a very important problem involves detecting the edges of objects, or equivalently, detecting the discontinuities of the underlying "intensity function" (the brightness of each point in the image is expressed by this function). In meteorology and oceanography, the equi-temperature surfaces of the high sky and the deep ocean are usually discontinuous. From a statistical viewpoint, all of these problems could be regarded as applications of estimation of two dimensional (2-D) jump regression surfaces (JRS). The purpose of this article is to develop a method to detect the jump locations of the 2-D JRS.

The research on jump regression models is currently under rapid development. In one dimensional (1-D) cases, McDonald and Owen (1986) suggested a "split linear smoother" that provided a discontinuity preserving curve estimator. Hall and Titterington (1992) proposed an alternative but simpler method to detect the jumps by establishing some relations among three local linear smoothers. Chen (1988), Müller (1992), Qiu (1994), Wu and Chu (1993a,b), and Yin (1988), among many others, discussed various kernel-type methods. These methods were all based on the difference between two one-sided kernel smoothers. The jump regression models were related to the kernel density estimation techniques by Chu (1994). Shiau (1985, 1987) regarded the jump regression models as partial linear regression models and fitted them with partial splines. Eubank and Speckman (1994) and Speckman (1993) treated the 1-D jump regression model as a semi-parametric regression

---

Peihua Qiu is Senior Research Consulting Statistician, Biostatistics Program, The Ohio State University, 320 West 10th Avenue, Columbus, OH 43210; e-mail: qiu@stat.ohio-state.edu. Brian Yandell is Professor, Department of Statistics, University of Wisconsin–Madison, 1210 West Dayton Street, Madison, WI 53706.

Journal of Computational and Graphical Statistics, Volume 6, Number 3, Pages 332–354

model and proposed estimates of the jump locations and magnitudes. Qiu and Yandell (1994) developed a jump detection algorithm with local polynomial regression. Loader (1994) suggested a jump detector based on local polynomial kernel estimators. Wang (1995) proposed detecting jumps with wavelet transformations.

In 2-D cases, Russian scientists did a lot of theoretical research in this area (see Brodsky and Darkhovsky (1993), Korostelev and Tsybakov (1993), and the references cited there). Korostelev and Tsybakov (1993) investigated jump location detection and fitting jump surfaces under several kinds of design and jump boundaries. They suggested approximating jump location curves by piecewise polynomials and then estimating the coefficients by maximum likelihood estimation. This method was proven to reach the "minimax optimal rate of convergence." O'Sullivan and Qian (1994) suggested detecting object boundaries by defining a contrast statistic. The boundary curves they considered are "smooth simple closed" curves. Müller and Song (1994) proposed "maximin" estimators of the jump boundaries of the d-dimensional ($d \geq 1$) jump surfaces under the condition that the number of such jump boundary curves (surfaces) is known and the population of the possible jump boundary curves (surfaces) is also known. Qiu (1997) suggests a so-called *rotational difference kernel estimator* (RDKE) of the jump location curves of the JRS. Both of the described methods were based on two one-sided kernel smoothers along a direction and the estimators were obtained by maximizing the jump detection criteria with respect to this direction. This makes the computation quite expensive. Rudemo and Stryhn (1994) studied two types of two-region image models with a univariate boundary representation and suggested a nonparametric histogram-like contour estimator. Godtliebsen and Chu (1995) discussed the estimation of the number of true gray levels, their values and relative frequencies of a noisy image by binning the original observations. When the jump location curves are completely known, Shiau, Wahba, and Johnson (1986) fit the 2-D JRS by the partial spline method.

Jump detection in regression surfaces is directly related to edge detection in computer image processing. Gonzalez and Woods (1992), Rosenfeld and Kak (1982), and Torre and Poggio (1986) present excellent overviews of computer edge detection techniques. Traditional edge detection operators such as the gradient operator, the Laplacian operator or the Laplacian-of-Gaussian operator (Marr and Hildreth 1980) represent high-pass filtering operators. These operators are only suitable for detecting limited types of edges and are highly susceptible to noise. Some improvements to the aforementioned derivative-based methods can be found (Johnson 1990; Qiu and Bhandarkar 1996). More recent edge detection techniques are based on optimal filtering (Canny 1986; Sarkar and Boyer 1990); random field models (Besag 1986; Geman and Geman 1984); surface fitting (Haralick 1984; Sinha and Schunk 1992); anisotropic diffusion (Perona and Malik 1990; Saint-Marc, Chen, and Medioni 1991); residual analysis (Chen, Lee, and Pavlidis 1991), global cost minimization using hill-climbing search (Tan, Gelfand, and Delp 1989); simulated annealing (Tan et al., 1991); and genetic algorithm (Bhandarkar, Zhang, and Potter 1994).

In this article we make another attempt to detect the jump locations of the JRS. Our method is based on the local simple least squares (LS) fitting. At a point in question, a LS plane is fitted in a neighborhood. The LS coefficients of this plane give an approximation of the gradient direction of the JRS at this point. They carry both the

continuous and the jump information about the JRS. We then try to delete the continuous information from the LS coefficients by considering two small neighborhoods along the approximated gradient direction, on either side of the original neighborhood. In such a way, the jump information is extracted. Based on that a jump detection criterion is derived. This criterion will be shown more sensitive to the jump structure than those of the kernel-type methods. This property makes the jump detection more robust to selection of the threshold value, which is desirable in applications. This method avoids using maximization with respect to the direction at each design point. It saves a lot of computation and makes it possible to handle very large data sets. Computation of the jump detection criterion can be updated easily from one point to the next. The whole algorithm requires $O(Nk)$ calculations, with N the sample size and k the window width of the neighborhoods. The only condition needed on the jump location curves is that their curvature has an upper bound. Comparing with the existing derivative-based edge detectors in image processing literature, we explicitly characterize the jump information in the edge detection criterion and eliminate the effect of the continuous variation of the intensity function on the edge detection. The numerical examples in Section 3 will show the value of the method. We also establish the statistical consistency of the edge detection procedure and provide the rate of convergence. The conditions imposed on the edge curves are mathematically explicitly expressed. These efforts, we think, might be helpful to the further development of edge detection techniques.

The rest of the article is organized as follows. The model and the jump detection method are described in Section 2. Numerical examples are discussed in Section 3, where we also compare our method with two popular edge detectors. We give some concluding remarks in Section 4. The proof of a theorem is given in the Appendix A.

2. JUMP DETECTION ALGORITHM

Observations $\{z_{ij}\}$ come from the following model

$$z_{ij} = f(x_i, y_j) + \epsilon_{ij}, \quad i, j = 1, 2, \ldots, n, \tag{2.1}$$

where $\{(x_i, y_j) = (i/n, j/n), i, j = 1, 2, \ldots, n\}$ are equally spaced design points in $[0, 1] \times [0, 1]$, $\{\epsilon_{ij}\}$ are iid random numbers with mean 0 and variance σ^2. The sample size is $N = n^2$. The regression function $f(x, y)$ is continuous over $[0, 1] \times [0, 1]$ except on some curves, which are called the jump location curves (JLCs) hereafter.

We call (x, y) *a singular point* of the JLCs if it is on a JLC and satisfies either one of the following two conditions.

1. There exists some constant $\zeta_0 > 0$ such that for any $0 < \zeta < \zeta_0$ the neighborhood of (x, y) with diameter ζ is divided into more than two connected regions by the JLCs.
2. We cannot find a constant $\rho_0 > 0$ such that there exist two orthogonal lines crossing at (x, y) and two vertical quadrants formed by these two lines belong to two different regions separated by a JLC in a neighborhood of (x, y) with diameter ρ_0.

Remark 1. *Condition 1 actually defines cross points of the JLCs (Fig. 1(a)). Con-*

dition 2 corresponds to acute angles of the JLCs (Fig. 1(b)). If a JLC has a unique tangent line at (x, y) (Fig. 1(c)) or a JLC has an obtuse angle at (x, y) (Fig. 1(d)), then (x, y) is not a singular point.

At any design point $(x_i, y_j), \ell + 1 \le i, j \le n - \ell$, we consider its neighborhood $N(x_i, y_j)$ with window width $k = 2\ell + 1 \ll n$, where ℓ is a non-negative integer.

$$N(x_i, y_j) := \{(x_{i+s}, y_{j+t}), s, t = -\ell, -\ell + 1, \ldots, 0, \ldots, \ell - 1, \ell\}.$$

A least squares plane is fitted in this neighborhood.

$$\hat{z}_{ij}(x, y) = \hat{\beta}_0^{(i,j)} + \hat{\beta}_1^{(i,j)}(x - x_i) + \hat{\beta}_2^{(i,j)}(y - y_j), \ (x, y) \in N(x_i, y_j).$$

After some calculations, we have

$$\begin{aligned} \hat{\beta}_0^{(i,j)} &= \frac{1}{k^2} z_{..} \\ \hat{\beta}_1^{(i,j)} &= \frac{1}{kS_x^2} \sum_{s=-\ell}^{\ell} (x_{i+s} - x_i) z_{i+s,.} \\ \hat{\beta}_2^{(i,j)} &= \frac{1}{kS_y^2} \sum_{t=-\ell}^{\ell} (y_{j+t} - y_j) z_{.,j+t}, \end{aligned} \tag{2.2}$$

where $z_{..} = \sum_{s,t=-\ell}^{\ell} z_{i+s,j+t}$, $z_{i+s,.} = \sum_{t=-\ell}^{\ell} z_{i+s,j+t}$, $z_{.,j+t} = \sum_{s=-\ell}^{\ell} z_{i+s,j+t}$, $S_x^2 = \sum_{s=-\ell}^{\ell} (x_{i+s} - x_i)^2$, $S_y^2 = \sum_{t=-\ell}^{\ell} (y_{j+t} - y_j)^2$. It is not hard to check that $\hat{\beta}_0^{(i,j)}, \hat{\beta}_1^{(i,j)}$ and $\hat{\beta}_2^{(i,j)}$ are uncorrelated. Furthermore, they have the following property:

Theorem 1. *In model (2.1), suppose that $f(x, y)$ has continuous first order partial derivatives over (0,1) × (0,1) except on the JLCs at which it has the first order right and left partial derivatives. The window width k satisfies the conditions that $\lim_{n\to\infty} k = \infty$ and $\lim_{n\to\infty} k/n = 0$. If there is no jump in $N(x_i, y_j)$, then*

$$\hat{\beta}_1^{(i,j)} = f_x'(x_i, y_j) + O\left(\frac{n\sqrt{loglogk}}{k^2}\right), \ a.s.$$

and

$$\hat{\beta}_2^{(i,j)} = f_y'(x_i, y_j) + O\left(\frac{n\sqrt{loglogk}}{k^2}\right), \ a.s.$$

If (x_i, y_j) is on a JLC and it is not a singular point, then

$$\hat{\beta}_1^{(i,j)} = f_x'(\tilde{x}_i, \tilde{y}_j) + h_1^{(i,j)} C(i, j) + \gamma_1 C_x(i, j) + O\left(\frac{n\sqrt{loglogk}}{k^2}\right), \ a.s.$$

and

$$\hat{\beta}_2^{(i,j)} = f_y'(\tilde{x}_i, \tilde{y}_j) + h_2^{(i,j)} C(i, j) + \gamma_2 C_y(i, j) + O\left(\frac{n\sqrt{loglogk}}{k^2}\right), \ a.s.$$

where $(\tilde{x}_i, \tilde{y}_j)$ is some point around (x_i, y_j) which satisfies (1) it is on the same side of the JLC as (x_i, y_j) and (2) the distance between $(\tilde{x}_i, \tilde{y}_j)$ and (x_i, y_j) tends to zero;

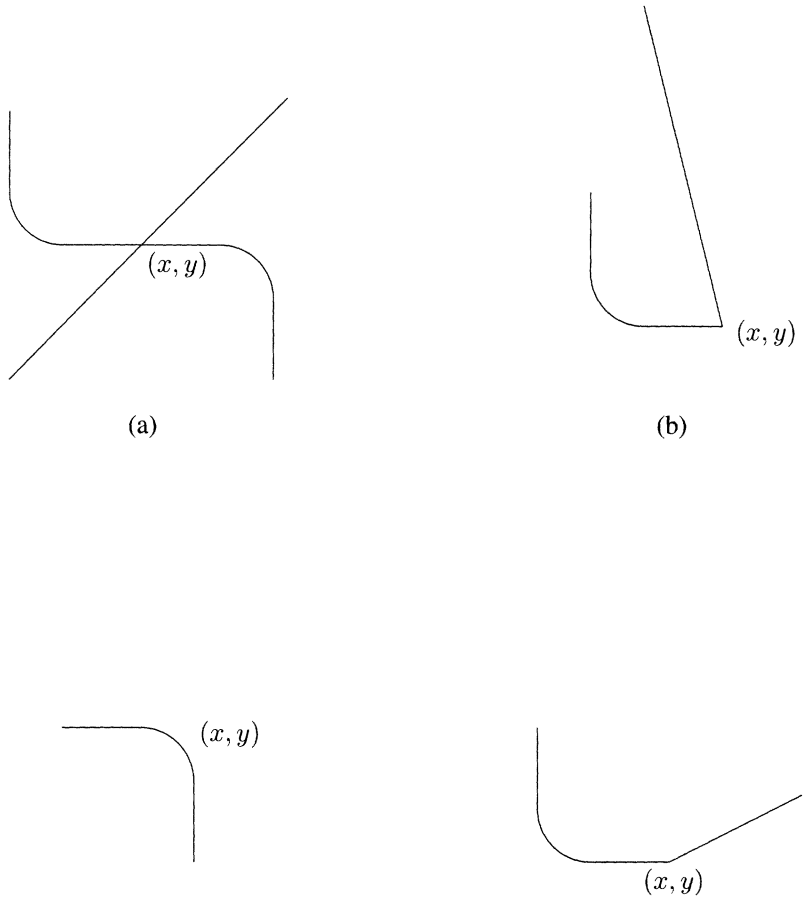

Figure 1. (a) The point (x, y) is a cross point of JLCs; (b) JLC has an acute angle at (x, y); (c) JLC has a unique tangent line at (x, y); (d) JLC has an obtuse angle at (x, y). The point (x, y) is a singular point in cases (a) and (b). It is a nonsingular point in cases (c) and (d).

$C(i, j), C_x(i, j)$, and $C_y(i, j)$ are the jump magnitudes of $f(x, y)$ and its first order x and y partial derivatives; $h_1^{(i,j)}$ and $h_2^{(i,j)}$ are two constants satisfying

$$\sqrt{\left(h_1^{(i,j)}\right)^2 + \left(h_2^{(i,j)}\right)^2} = O(n/k);$$

γ_1 and γ_2 are two constants between –1 and 1.

Proof is given in the Appendix A.

Remark 2. *Some quantities used in this article, such as $k, \hat{\beta}_1^{(i,j)}, \hat{\beta}_2^{(i,j)}, h_1^{(i,j)}$, and $h_2^{(i,j)}$ depend on n. We did not make this explicit in notation for simplicity. Their meaning should be clear from contextual introduction. The big "O" notation used in this article, $a_n = O(b_n)$, means that there exists some positive integer N_0 such that $|a_n/b_n| \leq M$ for $n > N_0$ and for some constant $M > 0$.*

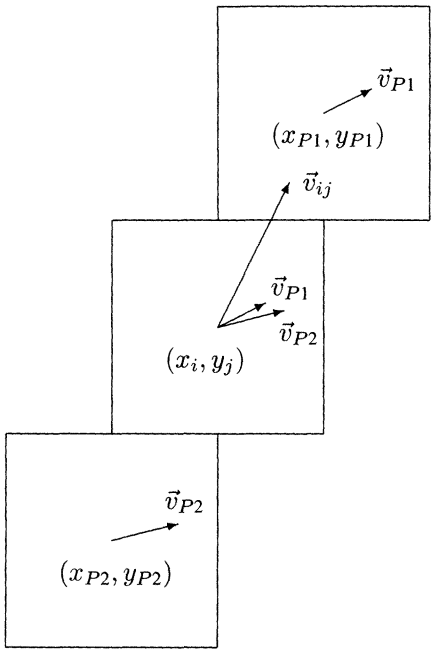

Figure 2. At design point (x_i, y_j), the jump detection criterion δ_{ij} is defined as the minimum length of the vectors $\vec{v}_{ij} - \vec{v}_{P1}$ and $\vec{v}_{ij} - \vec{v}_{P2}$, where $\vec{v}_{ij}, \vec{v}_{P1}$ and $\vec{v}_{P2}$ are the gradient vectors of the LS planes at (x_i, y_j) and two neighboring design points (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}) which are defined in (2.3)–(2.4).

In Theorem 1, the term $O\left(\frac{n\sqrt{\log\log k}}{k^2}\right)$ is due to noise. We could see that the slopes $\hat{\beta}_1^{(i,j)}$ and $\hat{\beta}_2^{(i,j)}$ carry both the continuous and the jump information of the JRS. We try to extract the jump information in the following way for a particular lattice point (x_i, y_j). The angle formed by $\vec{v}_{ij} := \left(\hat{\beta}_1^{(i,j)}, \hat{\beta}_2^{(i,j)}\right)$ and the positive direction of x-axis is denoted as $\theta \in [-\pi/4, 7\pi/4]$. Two neighboring design points (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}) are determined by the following formulas.

If $\pi/4 \le \theta < 3\pi/4$ or $5\pi/4 \le \theta < 7\pi/4$, then

$$\begin{aligned} x_{P1} &= x_i + \frac{k}{n \cdot \tan\theta}, \quad y_{P1} = y_j + \frac{k}{n} \\ x_{P2} &= x_i - \frac{k}{n \cdot \tan\theta}, \quad y_{P2} = y_j - \frac{k}{n}. \end{aligned} \tag{2.3}$$

If $-\pi/4 \le \theta < \pi/4$ or $3\pi/4 \le \theta < 5\pi/4$, then

$$\begin{aligned} x_{P1} &= x_i + \frac{k}{n}, \quad y_{P1} = y_j + \frac{k}{n} \cdot \tan\theta \\ x_{P2} &= x_i - \frac{k}{n}, \quad y_{P2} = y_j - \frac{k}{n} \cdot \tan\theta. \end{aligned} \tag{2.4}$$

If the two points determined by (2.3)–(2.4) are not exactly the grid points, we just choose two grid points which are closest to them instead.

Figure 2 gives an intuitive explanation of (2.3)–(2.4). (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}) have the following properties: (1) they are two design points on the line through (x_i, y_j)

and with slope $\hat{\beta}_2^{(i,j)}/\hat{\beta}_1^{(i,j)}$; (2) they are closest to (x_i, y_j) among the points on that line which neighborhoods have no overlap with $N(x_i, y_j)$. Notice that $\vec{v}_{ij}$ is the gradient vector of the fitted LS plane. The underlying JRS increases (or decreases) most rapidly along a nearby direction. If (x_i, y_j) is on a JLC and it is not a singular point at the same time, then as a direct conclusion of Theorem 1, the JLC could not be in $N(x_{P1}, y_{P1})$ and $N(x_{P2}, y_{P2})$ when n is large enough. In other words, (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}) are on two different sides of the JLC. We then define the following jump detection criterion δ_{ij}.

$$\delta_{ij} := \min\{\|\vec{v}_{ij} - \vec{v}_{P1}\|, \|\vec{v}_{ij} - \vec{v}_{P2}\|\}, \tag{2.5}$$

where $\vec{v}_{P1} := \left(\hat{\beta}_1^{(P1)}, \hat{\beta}_2^{(P1)}\right)$ and $\vec{v}_{P2} := \left(\hat{\beta}_1^{(P2)}, \hat{\beta}_2^{(P2)}\right)$ are gradient vectors of the fitted LS planes at (x_{P1}, y_{P1}) and (x_{P2}, y_{P2}), respectively, and $\|\cdot\|$ is the Euclidean norm.

If there is no jump in these three neighborhoods, then $\vec{v}_{ij}, \vec{v}_{P1}$, and $\vec{v}_{P2}$ should be close to each other. Hence δ_{ij} is small. If (x_i, y_j) is on a JLC and it is not a singular point, then by Theorem 1, $\delta_{ij} \approx \sqrt{\left(h_1^{(i,j)}\right)^2 + \left(h_2^{(i,j)}\right)^2}\, C(i,j) = O(n/k)C(i,j)$, which tends to infinity when n increases. Hence, δ_{ij} could be used to detect the jumps.

An equivalent and maybe more intuitive explanation of δ_{ij} is as follows. Let $\vec{u}_{ij} = \left(\hat{\beta}_1^{(i,j)}, \hat{\beta}_2^{(i,j)}, -1\right)$, $\vec{u}_{P1} = \left(\hat{\beta}_1^{(P1)}, \hat{\beta}_2^{(P1)}, -1\right)$, and $\vec{u}_{P2} = \left(\hat{\beta}_1^{(P2)}, \hat{\beta}_2^{(P2)}, -1\right)$ be the normal directions of the LS planes at (x_i, y_j), (x_{P1}, y_{P1}), and (x_{P2}, y_{P2}), respectively. δ_{ij} is the minimum length of vectors $\vec{u}_{ij} - \vec{u}_{P1}$ and $\vec{u}_{ij} - \vec{u}_{P2}$. If (x_i, y_j) is on a JLC, then $\vec{u}_{ij}$ has an abrupt change from both $\vec{u}_{P1}$ and $\vec{u}_{P2}$. Hence, δ_{ij} is relatively large.

A large value of δ_{ij} indicates a possible jump at (x_i, y_j). For any constant $b > 0$,

$$\begin{aligned} P(\delta_{ij} > b) &\leq P\left(\|\vec{v}_{ij} - \vec{v}_{P1}\| > b\right) \\ &= P\left(\left(\hat{\beta}_1^{(i,j)} - \hat{\beta}_1^{(P1)}\right)^2 + \left(\hat{\beta}_2^{(i,j)} - \hat{\beta}_2^{(P1)}\right)^2 > b^2\right) \\ &= E\left\{P\left(\left(\hat{\beta}_1^{(i,j)} - \hat{\beta}_1^{(P1)}\right)^2 + \left(\hat{\beta}_2^{(i,j)} - \hat{\beta}_2^{(P1)}\right)^2 > b^2 \,\middle|\, \hat{\beta}_1^{(i,j)}, \hat{\beta}_2^{(i,j)}\right)\right\}. \end{aligned}$$

For fixed $\hat{\beta}_1^{(i,j)}$ and $\hat{\beta}_2^{(i,j)}$, $\left(\left(\hat{\beta}_1^{(i,j)} - \hat{\beta}_1^{(P1)}\right)^2 + \left(\hat{\beta}_2^{(i,j)} - \hat{\beta}_2^{(P1)}\right)^2\right)/\sigma_{P1}^2$ is approximately χ_2^2 distributed under the assumption that there is no jump in

$$N(x_i, y_j) \bigcup N(x_{P1}, y_{P1}).$$

Here $\sigma_{P1}^2 = \text{var}\left(\hat{\beta}_1^{(P1)}\right) = \frac{\sigma^2}{kS_x^2}$. Therefore, a natural threshold value of δ_{ij} is

$$b = \sqrt{\chi_{2,\alpha_n}^2 \cdot \frac{\hat{\sigma}^2}{kS_x^2}} = \hat{\sigma}\sqrt{\frac{\chi_{2,\alpha_n}^2}{kS_x^2}}, \tag{2.6}$$

where χ_{2,α_n}^2 is a $1 - \alpha_n$ quantile of the χ_2^2 distribution and $\hat{\sigma}$ is a consistent estimator of σ.

Suppose that (x_i, y_j) is on a JLC with jump magnitude $C(i,j)$. Then the values of most kernel-type jump detection criteria (e.g., Müller and Song 1994) are about $C(i,j)$ at this point and our criterion is of order $O(n/k)$, which tends to infinity with the sample size. Hence δ_{ij} is more sensitive to the jumps. This property has two benefits. One is that δ_{ij} visually reveals the jumps better. The other is that our jump detector is more robust to the selection of the threshold. The threshold could be chosen a little bit larger than usual without missing the jumps when the sample size is larger because δ_{ij} is quite large in this case.

The design points $\{(x_i, y_j) : \delta_{ij} > b, i, j = (3k+1)/2, \ldots, n-(3k-1)/2\}$ could be flagged as jump candidates. But two kinds of deceptive jump candidates may also be flagged. One kind is those close to the real jumps. If (x_i, y_j) is flagged, then its neighboring design points will be flagged with high probability. This kind of deceptive candidate could make the detected jump boundaries thick. Another kind of deceptive candidate arises from the randomness of the jump detection criterion. (x_i, y_j) could be flagged with probability α_n even if it is not the jump point. This kind of deceptive candidate is scattered over the whole design space. Therefore we need some modification procedures (MPs) to delete these two kinds of candidates. Here are two possible strategies.

MP1 (to make the detected jump boundaries thin). For each $(3k+1)/2 \le j \le n-(3k-1)/2$, consider the design points $\{(x_i, y_j) : (3k+1)/2 \le i \le n-(3k-1)/2\}$ on line $y = y_j$. Among them, the flagged candidates that satisfy $-\pi/4 \le \theta < \pi/4$ or $3\pi/4 \le \theta < 5\pi/4$ are denoted as $\{(x_{i_s}, y_j) : 1 \le s \le n_1\}$, where θ is defined as in (2.3)–(2.4). If there are $r_1 < r_2$ such that the increments of the sequence $i_{r_1} < i_{r_1+1} < \cdots < i_{r_2}$ are all less than the window width k, but $i_{r_1} - i_{r_1-1} > k$ and $i_{r_2+1} - i_{r_2} > k$, then we say that $\{(x_{i_s}, y_j) : r_1 \le s \le r_2\}$ forms a *tie* and we select the middle point $((x_{i_{r_1}} + x_{i_{r_2}})/2, y_j)$ as a new candidate to replace the tie set. That is, reduce the candidate set to one representative, the middle point, from each tie set. Along y direction, for each $(3k+1)/2 \le i \le n-(3k-1)/2$, consider the design points $\{(x_i, y_j) : (3k+1)/2 \le j \le n-(3k-1)/2\}$ on line $x = x_i$. We do the same modification as that along x direction except that only those candidates satisfying $\pi/4 \le \theta < 3\pi/4$ or $5\pi/4 \le \theta < 7\pi/4$ are considered this time.

MP2 (to delete scattered candidates). For any candidate (x_i, y_j), if the number of other jump candidates in $N(x_i, y_j)$ is less than $(k-1)/2$, then we delete (x_i, y_j) from the candidates set.

Remark 3. *In MP1, along x direction, the condition $-\pi/4 \le \theta < \pi/4$ or $3\pi/4 \le \theta < 5\pi/4$ implies that the possible JLC forms an acute angle with the y-axis at the candidate point. This avoids cancelling the real jump boundaries that are parallel to the x-axis.*

Remark 4. *We want to point out that the modification procedures (MP1 and MP2) are only two possible ones. The detected JLCs may also be fragmentary here or there. (This*

can be seen from the numerical examples in the next section.) Some MPs are necessary to connect the detected JLCs. For more discussion of the MPs, see Gonzalez and Woods (1992, chap. 6).

We summarize the jump detection method in the following algorithm.

2.1 THE JUMP DETECTION ALGORITHM

1. At any (x_i, y_j) with $\ell + 1 \le i, j \le n - \ell$, fit a LS plane in $N(x_i, y_j)$ by formula (2.2).
2. Use (2.3)–(2.4) to determine two neighboring design points of $(x_i, y_j), (3k + 1)/2 \le i, j \le n - (3k - 1)/2$.
3. Use formula (2.5) to calculate δ_{ij}.
4. Use formula (2.6) to determine the threshold value b.
5. Flag the design point (x_i, y_j) as a jump candidate if it satisfies $\delta_{ij} > b$.
6. Use modification procedures MP1 and MP2 to determine the final candidates.

Remark 5. *We use square-shaped neighborhoods of the design points in the method only for convenience of the expressions and calculations. Other kinds of neighborhoods, such as ellipses, could also be used.*

Remark 6. *The LS planes fitted in the jump detection algorithm can be updated easily from one point to the next because only a few points change. Therefore, the algorithm needs $O(Nk)$ calculations. It will simplify the computer program further if we notice the symmetric structures between $\hat{\beta}_1^{(i,j)}$ and $\hat{\beta}_2^{(i,j)}$ (see (2.2)).*

Theorem 2. *Besides the conditions stated in Theorem 1, if there are only finite number of singular points on JLCs and α_n in (2.6) is chosen such that (1) $\lim_{n\to\infty}\alpha_n = 0$; (2) $\lim_{n\to\infty}\log(\alpha_n)/\log(\log(k)) = -\infty$; and (3) $\lim_{n\to\infty}\log(\alpha_n)/k^2 = 0$, then the detected jumps are a.s. consistent in the Hausdorff distance and the convergence rate is $O(n^{-1}\log(n))$.*

The proof is based on the following facts. First, from Theorem 1, we know that the jump information in the jump detection criterion δ_{ij} is of order $O(n/k)$. Second, the threshold value b in (2.6) is of order $O\left(n\sqrt{-\log(\alpha_n)}/k^2\right)$. (We use the fact that $\chi^2_{2,\alpha_n} = -2\log(\alpha_n)$ here.) Third, the order of the standard deviation of δ_{ij} is $O\left(n\sqrt{\log\log k}/k^2\right)$. So the jump information dominates the randomness in the jump detection criterion as long as k tends to infinity with n. An outline of the proof is: (1) no jumps can be detected outside the bands centered at the real jump boundaries and with width $\sqrt{2}k/n$ when n is large enough; (2) if the design point (x_i, y_j) is on a JLC and it is not a singular point, then it will be detected because $\delta_{ij} = O(n/k) > b$ with large enough n. The convergence rate of the detected jumps is $O(k/n)$. If $k = O(\log(n))$, then this rate is $O(n^{-1}\log(n))$. The Hausdorff distance between two sets $\mathcal{A}$ and $\mathcal{B}$ is defined by $d(\mathcal{A}, \mathcal{B}) := \max\left(\sup_{x\in\mathcal{A}}\inf_{y\in\mathcal{B}}\|x - y\|, \sup_{x\in\mathcal{B}}\inf_{y\in\mathcal{A}}\|x - y\|\right)$.

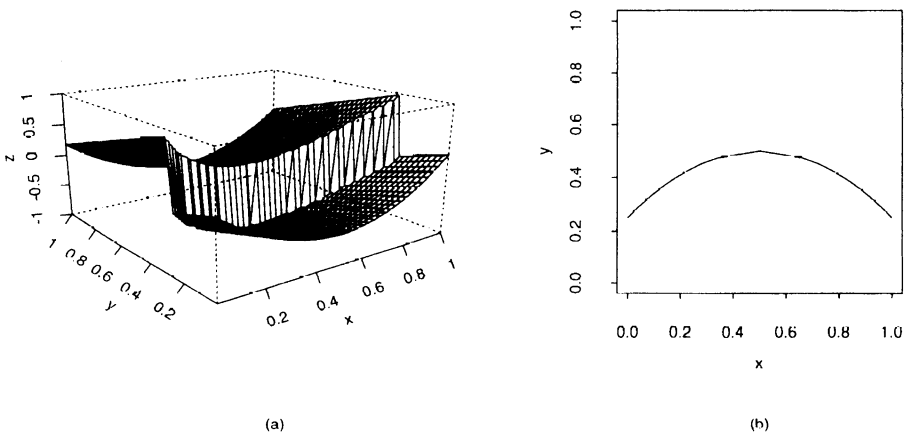

Figure 3. (a) The jump regression surface used in the example; (b) the jump location curve.

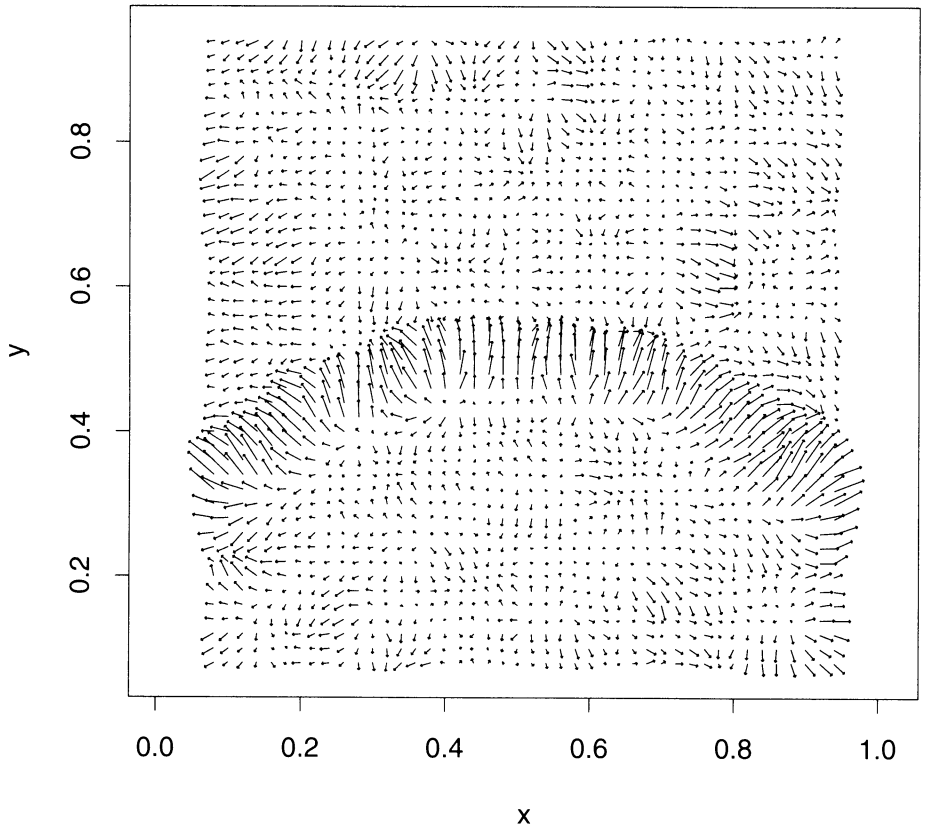

Figure 4. The gradient vector $\vec{v}_{ij}$ of the fitted LS plane at each design point.

3. NUMERICAL ANALYSIS

3.1 A SIMULATION EXAMPLE

In this section we do some simulations with an artificial example. The regression function $f(x, y)$ has the expression

$$f(x, y) = -.5 - y + 3(x - .5)^2 + I_{\{y > -(x-.5)^2 + .5\}}.$$

There is one JLC $y = -(x - .5)^2 + .5$ with constant jump magnitude 1. The regression function and the JLC are plotted in Figure 3. Ten thousand observations $\{z_{ij}, i, j = 1, 2, \ldots, 100\}$ are generated from $z_{ij} = f(i/n, j/n) + \epsilon_{ij}$ with $n = 100$ and iid random numbers from $N(0, .5^2)$.

We then use formulas (2.2)–(2.5) to calculate the jump detection criterion $\{\delta_{ij}\}$, initially with $k = 7$. The gradient vector $\vec{v}_{ij}$ of the fitted LS plane at each design point is shown by Figure 4. $\{\delta_{ij}\}$ is plotted in Figure 5(a) (3-D plot) and Figure 5(b) (a corresponding image plot). In the image plot, the brightness at each point represents the response value—the darker, the bigger. We use the default legend of the "`image()`" function in S-Plus to generate the image plots used in this section.

Then a threshold is calculated by formula (2.6) with $\alpha_n = .001$, which is the smallest number in most χ^2 tables. The flagged jump candidates are plotted in Figure 6(b) by black points. As we mentioned in Section 2, the detected jump boundary is quite thick and there are also some scattered candidates. We then use the modification procedure MP1 and then MP2 to modify the set of candidates. The results are plotted in Figure 6(c) and Figure 6(d). As a comparison, we plot the real JLC in Figure 6(a). We notice that there are some breaks here or there in the detected jump boundary in Figure 6(d). The detected boundary is not thin enough at some places. These imply that there is some room for our MPs to be improved, as we mentioned in Remark 4.

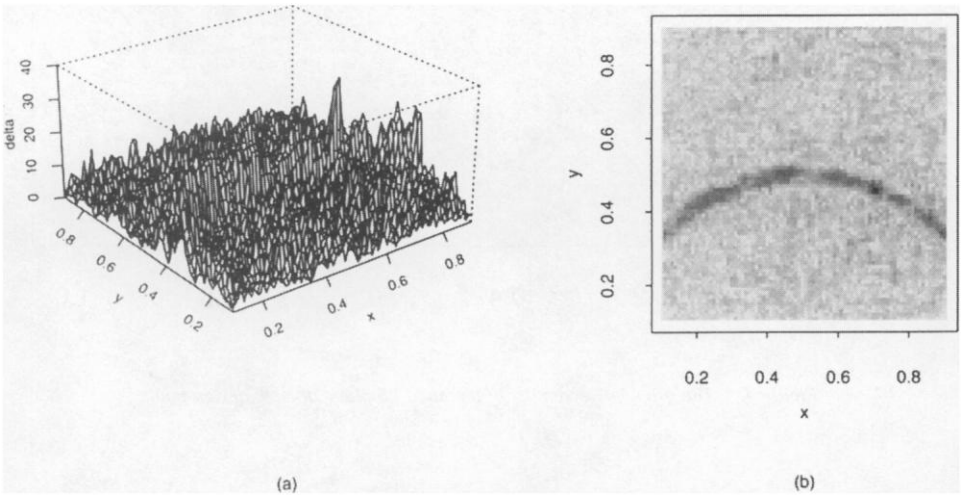

Figure 5. (a) The jump detection criterion $\{\delta_{ij}\}$ in 3-D plot; (b) the corresponding image plot of (a). The darker the color in the image, the bigger the criterion value.

The experiment is then repeated 1,000 times. The number of times of each design point to be in the final candidates set is plotted in Figure 7. We can see that the results are quite impressive.

Theoretically, we can use the Hausdorff distance to measure the performance of our algorithm. In reality, this distance could be very hard to compute. In the following, we use the average orthogonal distance of the points in the final set of candidates to the real JLCs as a performance measurement. This measurement is averaged again for 1,000 replications. The results for several n, k, and σ^2 values are presented in Figure 8. From the plots, we can see that the averaged performance measurement (APM) decreases when n increases for each σ^2 value. This may reflect the consistency of the algorithm. For fixed σ^2 value, the best k (k with smallest APM) does not appear to change much with n. That verifies the conclusion in Theorem 2 that k should be quite stable when n increases, to achieve the biggest accuracy of the detected jumps. The best k increases with σ^2 value. That implies that for noisier data more observations are needed in each window to reduce the randomness in the jump detection criterion. We also notice that APM is much smaller for smaller σ^2 value. This is intuitively reasonable.

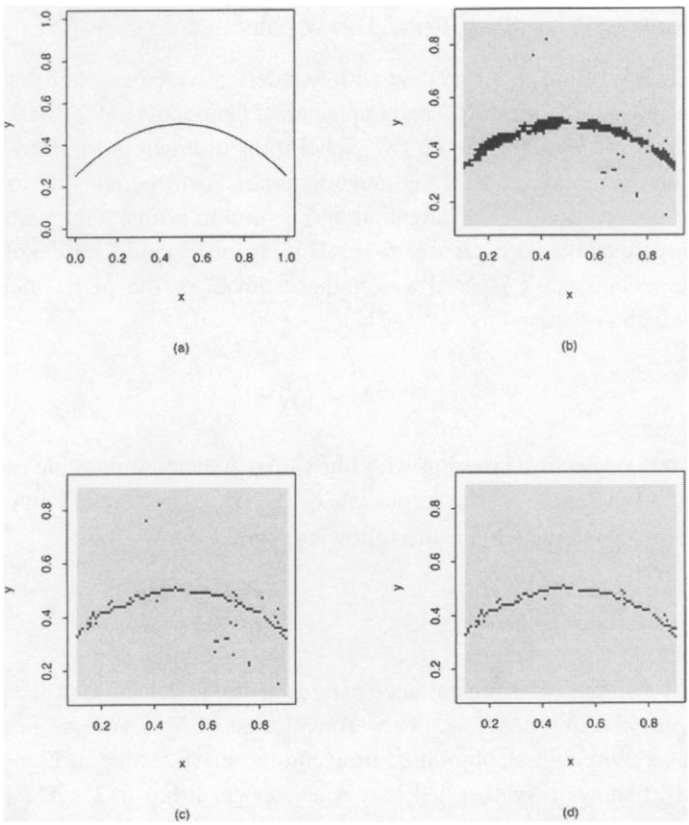

Figure 6. (a) The real jump location curve; (b) detected jump candidates from criterion (2.5), which are denoted by black points; (c) the modified jump candidates from those in (b) by the modification procedure MP1; (d) the modified jump candidates from those in (c) by the modification procedure MP2.

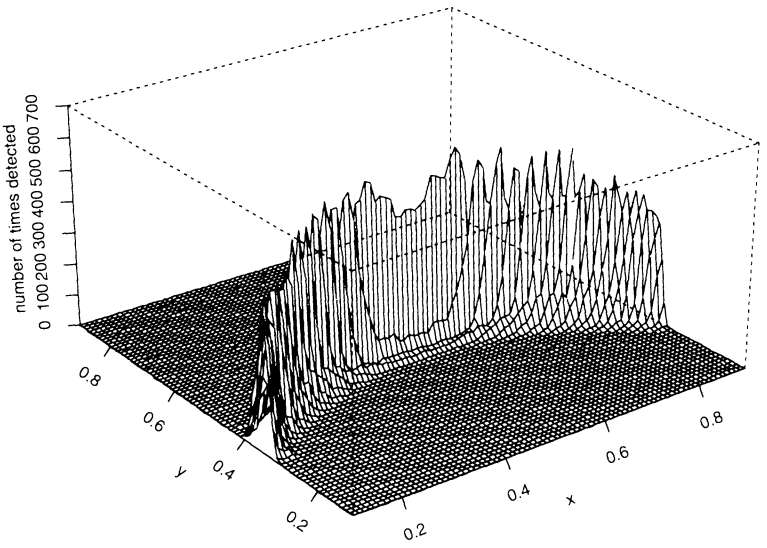

Figure 7. The number of times each design point is detected in 1,000 replications.

3.2 Comparison With Two Edge Detectors

In this part, we compare our method with two derivative-based edge detectors: Sobel (Rosenfeld and Kak 1982, sec. 10.2) and Laplacian of Gaussian (LoG) (Marr and Hildreth 1980). The Sobel method is based on two Sobel masks, shown in Figure 9. For a given design point (x_i, y_j), we use its 3×3 neighborhood. Then the convolution of the first mask with the observations in the neighborhood is used to estimate the partial derivative of the intensity function with respect to y. This estimate could be denoted as $\hat{f}_y^{(i,j)}$. Similarly, the second mask is used to get the estimate of the partial derivative with respect to x which is denoted as $\hat{f}_x^{(i,j)}$. Then

$$S := \left[\left(\hat{f}_x^{(i,j)} \right)^2 + \left(\hat{f}_y^{(i,j)} \right)^2 \right]^{1/2}$$

is used as an edge detection criterion with big values indicating possible edges.

LoG method is based on the Laplacian of a 2-D Gaussian function of the form $h(x, y) = \exp(-\frac{x^2+y^2}{2\sigma_*^2})$, which has the following form.

$$\nabla^2 h = \frac{r^2 - \sigma_*^2}{\sigma_*^4} \exp\left(-\frac{r^2}{2\sigma_*^2} \right),$$

where $r^2 = x^2 + y^2$ and σ_*^2 is the variance parameter. Edge detection is accomplished by locating the zero-crossing positions of the convolution of $\nabla^2 h$ and the image function. In applications, $\nabla^2 h$ could be simplified by using the mask shown in Figure 10.

Figure 11(a) shows a synthesized gray scale image. It has 512×512 pixels with a gray scale resolution of 8 bits per pixel (the range of gray levels is from 0 to 255). Pixels on rows 1–86 have a gray level value of 155. The gray levels on rows 87–172 are 165. From row 173 to row 212, the gray levels linearly decrease, which could be described

by the formula $255 - (i - 172) * 6$, where i is the row number. From row 213 to row 512, there are two portions. The left portion consists of 8 sections where the gray levels in each section are identical. The gray levels of adjacent sections alternate between 155 and 165. In the right portion, white noise from uniform distribution $U(-10, 10)$ is added to each pixel. Before adding the noise, gray levels are 100 on rows 213–312; 75 on rows 313–412, and 100 again on rows 413–512.

Figure 11(b) shows the result of Sobel detector. In the top part of the image, it misses the edge between rows 86–87 and falsely detects the region between rows 173–212. In that region, gray levels have a quite steep linear trend. But no edges exist. Because the Sobel detector does not exclude the linear trend of the underlying intensity function from edge detection, it cannot overcome this false-edge detection and detect the true edge between rows 86–87 at the same time. In the left portion of the bottom half of the image, there are two diagonal edge lines, one vertical edge line and one horizontal

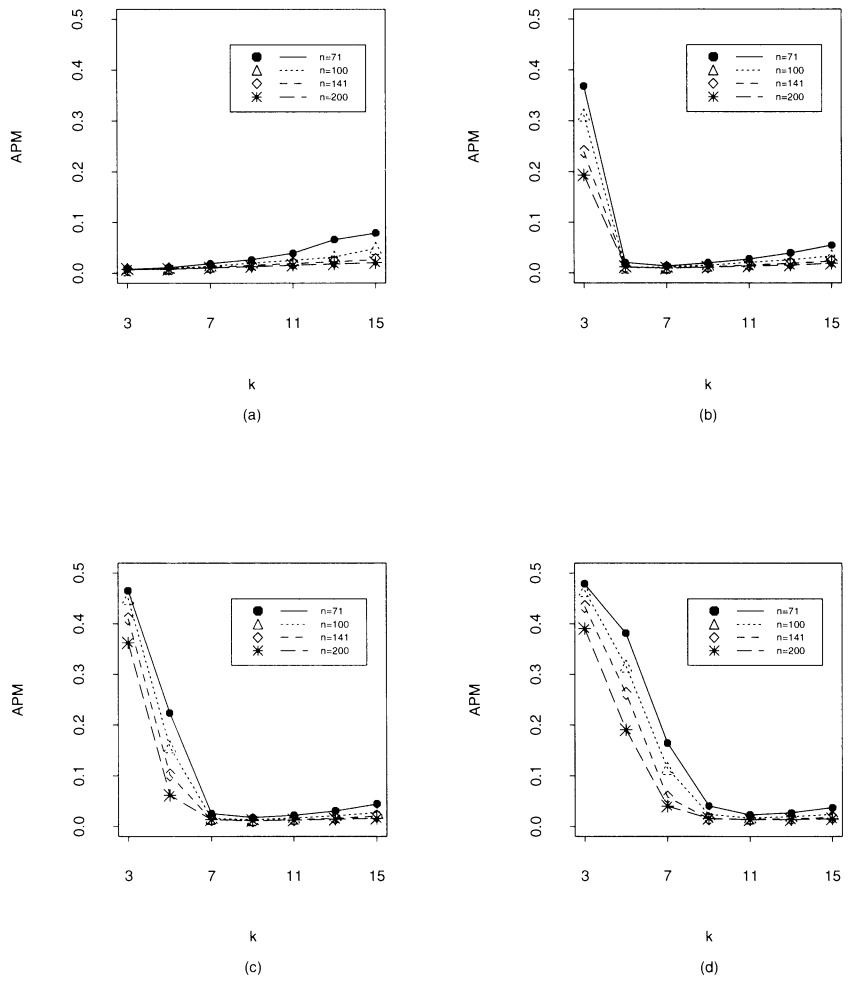

Figure 8. Averaged performance measurements (APMs) from 1,000 replications with several n, k, and σ^2 values. (a) $\sigma^2 = .25$; (b) $\sigma^2 = .5$; (c) $\sigma^2 = .75$; (d) $\sigma^2 = 1.0$.

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

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

Figure 9. Sobel masks.

edge line, all of which have the same intensity step magnitudes. As we can see from the plot, however, the Sobel detector could detect two diagonal edge lines but it misses the other two edge lines. The reason behind this is that Sobel operator only considers two orthogonal directions (x and y directions) in edge detection. Consequently, the values of edge detection criterion depend on the directions of the edge lines. The Sobel operator detects the true edge lines in the noisy portion (i.e., right lower portion) of the image, but also several noisy edge pixels as well.

Figure 11(c) shows the edges detected by LoG. The LoG operator suffers some of the same drawbacks as the Sobel operator. Note that although the false-edge detection problem in the region between rows 173–212 has been somewhat alleviated, the true edge between rows 86–87 has been missed. The LoG operator misses most of the edge boundaries in the noisy portion of the image although it suppresses the noisy edge pixels as well. It also does not detect all the eight edge lines in the left lower portion of the image.

The result of our method is shown in Figure 11(d). To make the methods comparable, we use the same window size as Sobel and LoG. We see that the false-edge detection, because of the linear trend, and the missing-edge, because of the different directions of the edge lines, have been greatly improved since we eliminate the effect of the continuous variation of the intensity function on edge detection and we construct our edge detection criterion in the approximated gradient direction. The results around the intersections of several edge lines are not quite satisfactory. These intersections correspond to the singular points of the JLCs.

0	-1	0
-1	4	-1
0	-1	0

Figure 10. Mask for LoG method.

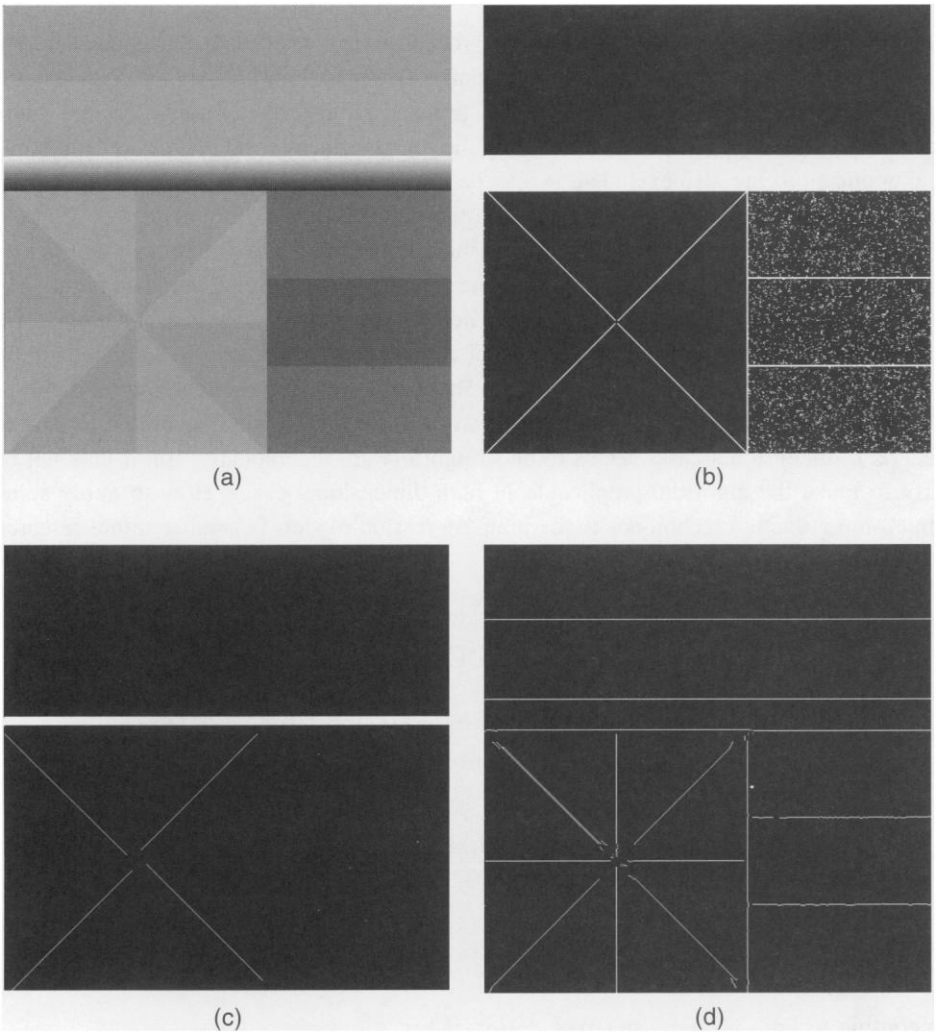

Figure 11. (a) The original image; (b) detected edge image by Sobel; (c) detected edge image by Laplacian of Gaussian; (d) detected edge image by the procedure suggested in this article.

4. SOME CONCLUDING REMARKS

We have presented a jump boundary detection algorithm with local LS plane fitting that is intuitively appealing and simple to use. It can handle relatively large data sets. Simulations show that it works well in practice.

We leave some parameters, such as the window width used in the algorithm, to be adjustable to the users. Much future research is needed to provide some guidelines on the selection of these parameters. The window sizes used in this article are fixed throughout. In some applications, windows with variable or adaptable sizes may be more appropriate. In some planar regions of the regression surface, we could use large-size windows, whereas in highly textured regions the window sizes could be small. As we mentioned, the modification procedures presented in the article are only two of the

possible ones. More careful modification procedures are needed to make the detected jump candidates match the real jump boundaries better. Another very important issue is the relationship between jump location detection and jump surface fitting. If we put more structure on the jump locations, then fitting the jump surfaces would be easier. But some real applications are also excluded. It may be important to work out some methods to fit the jump surfaces under mild conditions on the jump locations.

We discussed jumps in the regression functions in this article. In some situations jumps in derivatives are also interesting. (The so-called "roof-edges" in image processing correspond to the jumps in the first order derivatives of the regression functions.) We think that the coefficients of the fitted local polynomials of order $k+1$ contain useful information about the jumps in the kth derivatives of the underlying regression functions. This kind of relationship needs to be investigated further. Generalization from 2-D to general d-dimensional cases seems to be straightforward theoretically. But it may not be easy to make the algorithm applicable in high-dimensional cases. How to apply some dimension-reduction techniques to the jump regression models is another future research topic.

A. APPENDIX

A.1 PROOF OF THEOREM 1

If there is no jump in $N(x_i, y_j)$, then from (2.2),

$$\begin{aligned}\hat{\beta}_1^{(i,j)} &= \frac{1}{kS_x^2}\sum_{s=-\ell}^{\ell}(x_{i+s}-x_i)z_{i+s,.} \\ &= \frac{1}{kS_x^2}\sum_{s=-\ell}^{\ell}(x_{i+s}-x_i)\sum_{t=-\ell}^{\ell}[f(x_{i+s},y_{j+t})+\epsilon_{i+s,j+t}]. \qquad \text{(A.1)}\end{aligned}$$

According to Lai, Robbins, and Wei (1979), if $\lim_{n\to\infty} k = \infty$, then

$$\frac{\sum_{s=-\ell}^{\ell}\sum_{t=-\ell}^{\ell} s\epsilon_{i+s,j+t}}{\sqrt{k^4 \log\log k^4}} = O(1), \text{ a.s.}$$

Hence

$$\begin{aligned}&\frac{1}{kS_x^2}\sum_{s=-\ell}^{\ell}(x_{i+s}-x_i)\sum_{t=-\ell}^{\ell}\epsilon_{i+s,j+t} \\ &= \frac{1}{kS_x^2}\sum_{s=-\ell}^{\ell}\frac{s}{n}\sum_{t=-\ell}^{\ell}\epsilon_{i+s,j+t} \\ &= \frac{1}{nkS_x^2}\sqrt{k^4\log\log k^4}\,\frac{\sum_{s=-\ell}^{\ell}\sum_{t=-\ell}^{\ell} s\epsilon_{i+s,j+t}}{\sqrt{k^4\log\log k^4}} \\ &= O\left(\frac{n}{k^2}\sqrt{\log\log k}\right), \text{ a.s.} \qquad \text{(A.2)}\end{aligned}$$

By Taylor's expansion,

$$\begin{aligned} f(x_{i+s}, y_{j+t}) &= f(x_i, y_j) + f'_x(x_i + \eta(x_{i+s} - x_i), y_j + \eta(y_{j+t} - y_j))(x_{i+s} - x_i) \\ &\quad + f'_y(x_i + \eta(x_{i+s} - x_i), y_j + \eta(y_{j+t} - y_j))(y_{j+t} - y_j), \end{aligned}$$

where $0 < \eta < 1$ is a constant which depends on (x_{i+s}, y_{j+t}). So when $\lim_{n\to\infty} k/n = 0$,

$$\begin{aligned} &\frac{1}{kS_x^2} \sum_{s=-\ell}^{\ell} (x_{i+s} - x_i) \sum_{t=-\ell}^{\ell} f(x_{i+s}, y_{j+t}) \\ &\quad = f'_x(x_i, y_j) + \frac{1}{k} \sum_{t=-\ell}^{\ell} [f'_x(x_i + \eta(x_{i+s} - x_i), y_j + \eta(y_{j+t} - y_j)) - f'_x(x_i, y_j)] \\ &\qquad + \frac{1}{kS_x^2} \sum_{s=-\ell}^{\ell} (x_{i+s} - x_i) \sum_{t=-\ell}^{\ell} [f'_y(x_i + \eta(x_{i+s} - x_i), y_j + \eta(y_{j+t} - y_j)) \\ &\qquad - f'_y(x_i, y_j)](y_{j+t} - y_j) \\ &\quad = f'_x(x_i, y_j) + o(1). \end{aligned} \tag{A.3}$$

In the second equation of (A.3), we used the facts that

$$\max_{0<\eta<1, -\ell\le t\le\ell} |f'_x(x_i + \eta(x_{i+s} - x_i), y_j + \eta(y_{j+t} - y_j)) - f'_x(x_i, y_j)| = o(1),$$

and

$$\max_{0<\eta<1, -\ell\le t\le\ell} |f'_y(x_i + \eta(x_{i+s} - x_i), y_j + \eta(y_{j+t} - y_j)) - f'_y(x_i, y_j)| = o(1).$$

Combining (A.1)–(A.3), we have

$$\hat{\beta}_1^{(i,j)} = f'_x(x_i, y_j) + O\left(\frac{n}{k^2}\sqrt{\log\log k}\right), \text{ a.s.}$$

Similarly,

$$\hat{\beta}_2^{(i,j)} = f'_y(x_i, y_j) + O\left(\frac{n}{k^2}\sqrt{\log\log k}\right), \text{ a.s.}$$

Now, if (x_i, y_j) is on a JLC and it is not a singular point, and $N(x_i, y_j)$ is divided into two parts I_1 and I_2 by this JLC (without loss of generality, let us assume that there is a positive jump from I_1 to I_2 at (x_i, y_j)), then from (A.1)–(A.2)

$$\begin{aligned} \hat{\beta}_1^{(i,j)} &= \frac{1}{kS_x^2} \sum_{s=-\ell}^{\ell} (x_{i+s} - x_i) \sum_{t=-\ell}^{\ell} f(x_{i+s}, y_{j+t}) + O\left(\frac{n}{k^2}\sqrt{\log\log k}\right), \text{ a.s.} \\ &= \frac{1}{kS_x^2} \sum_{s=-\ell}^{\ell} (x_{i+s} - x_i) \sum_{t=-\ell}^{\ell} f^*(x_{i+s}, y_{j+t}) + h_1^{(i,j)} C(i,j) \\ &\quad + \gamma_1 C_x(i,j) + O\left(\frac{n}{k^2}\sqrt{\log\log k}\right) \\ &= f'_x(\tilde{x}_i, \tilde{y}_j) + h_1^{(i,j)} C(i,j) + \gamma_1 C_x(i,j) + O\left(\frac{n}{k^2}\sqrt{\log\log k}\right), \end{aligned}$$

where $(\tilde{x}_i, \tilde{y}_j)$ is some design point around (x_i, y_j) which satisfies the conditions stated in the theorem,

$$f^*(x_{i+s}, y_{j+t}) = \begin{cases} f(x_{i+s}, y_{j+t}) & \text{if } (x_{i+s}, y_{j+t}) \in I_1 \\ f(x_{i+s}, y_{j+t}) & \\ \quad -C(i,j) - (x_{i+s} - x_i)C_x(i,j) & \text{if } (x_{i+s}, y_{j+t}) \in I_2, \end{cases}$$

$$h_1^{(i,j)} = \frac{1}{kS_x^2} \sum_{(x_{i+s}, y_{j+t}) \in I_2} (x_{i+s} - x_i),$$

and

$$\gamma_1 = \frac{1}{kS_x^2} \sum_{(x_{i+s}, y_{j+t}) \in I_2} (x_{i+s} - x_i)^2.$$

Clearly, γ_1 is a constant between 0 and 1. To make $C_x(i,j)$ positive, we let γ_1 take its value between -1 and 1.
Similarly,

$$\hat{\beta}_2^{(i,j)} = f_y'(\tilde{x}_i, \tilde{y}_j) + h_2^{(i,j)} C(i,j) + \gamma_2 C_y(i,j) + O\left(\frac{n}{k^2}\sqrt{\log\log k}\right), \text{ a.s.}$$

with

$$h_2^{(i,j)} = \frac{1}{kS_y^2} \sum_{(x_{i+s}, y_{j+t}) \in I_2} (y_{j+t} - y_j)$$

and

$$\gamma_2 = \frac{1}{kS_y^2} \sum_{(x_{i+s}, y_{j+t}) \in I_2} (y_{j+t} - y_j)^2.$$

Next, we notice that condition 2 in the definition of a singular point guarantees the existence of the left and the right derivatives of the JLC at (x_i, y_j). To make this more precise, let us suppose that a JLC has a parametric form $(x(t), y(t)), t \in [T_1, T_2]$. Then it has right and left derivatives at $t_0 \in (T_1, T_2)$, if $\lim_{t \to t_0 + 0} x(t)$, $\lim_{t \to t_0 - 0} x(t)$, $\lim_{t \to t_0 + 0} y(t)$, and $\lim_{t \to t_0 - 0} y(t)$ all exist. Based on this fact, we will prove that

$$\sqrt{\left(h_1^{(i,j)}\right)^2 + \left(h_2^{(i,j)}\right)^2} = O(n/k) \tag{A.4}$$

by considering the following two cases.

A.2 CASE 1

The left and right tangent half lines of the JLC at (x_i, y_j) are both under the line $y = y_j$ in $N(x_i, y_j)$ (see Fig. A.1). In this case, by condition 2 of the definition of a singular point we can find two orthogonal lines crossing each other at (x_i, y_j) and two

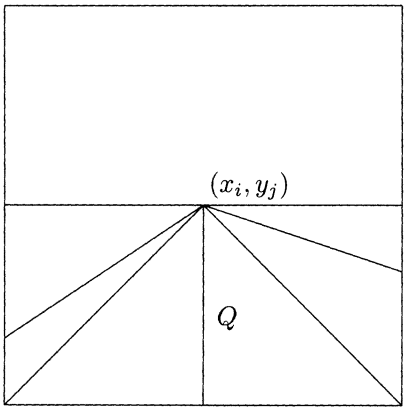

Figure A.1

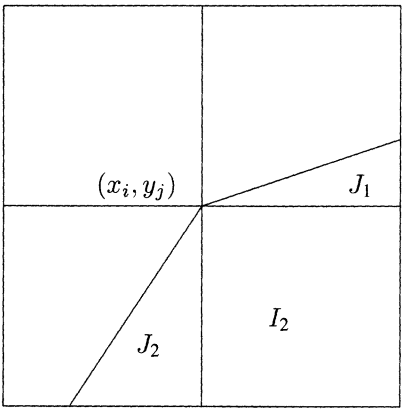

Figure A.2

Figure A.1–A.2. Two cases that (x_i, y_j) is on a JLC and it is not a singular point.

vertical quadrants formed by them belong to different parts I_1 and I_2. For convenience, let us assume that the lower quadrant Q belongs to I_2. Clearly, when n is large enough (the difference between the tangent half lines and the real JLC could be neglected in the small neighborhood),

$$\begin{aligned} |h_2^{(i,j)}| &> |\frac{1}{kS_y^2} \sum_{(x_{i+s}, y_{j+t}) \in Q} (y_{j+t} - y_j)| \\ &> |\frac{1}{kS_y^2} \sum_{s=0}^{\ell} \sum_{t=-\ell}^{0} (y_{j+t} - y_j)| \\ &= O(n/k). \end{aligned} \tag{A.5}$$

On the other hand,

$$\max\{|h_1^{(i,j)}|, |h_2^{(i,j)}|\} \leq 2 \cdot |\frac{1}{kS_y^2} \sum_{s=0}^{\ell} \sum_{t=-\ell}^{0} (y_{j+t} - y_j)|.$$

Hence (A.4) is true.

A.3 CASE 2

The left and right tangent half lines of the JLC at (x_i, y_j) are in two vertical quadrants in $N(x_i, y_j)$ (see Fig. A.2). Without loss of generality, we assume that one is in the upper-right quadrant and the other is in the lower-left quadrant. The former forms a region J_1 with line $y = y_j$ and the latter forms a region J_2 with line $x = x_i$. If

$$\sum_{(x_{i+s}, y_{j+t}) \in J_2} |y_{j+t} - y_j| \geq \sum_{(x_{i+s}, y_{j+t}) \in J_1} |y_{j+t} - y_j|,$$

then (A.5) and consequently (A.4) is true. If

$$\sum_{(x_{i+s}, y_{j+t}) \in J_2} |y_{j+t} - y_j| < \sum_{(x_{i+s}, y_{j+t}) \in J_1} |y_{j+t} - y_j|,$$

then by the facts that

$$\sum_{(x_{i+s},y_{j+t})\in J_1} |y_{j+t} - y_j| < \sum_{(x_{i+s},y_{j+t})\in J_1} |x_{i+s} - x_i|$$

and

$$\sum_{(x_{i+s},y_{j+t})\in J_2} |y_{j+t} - y_j| > \sum_{(x_{i+s},y_{j+t})\in J_2} |x_{i+s} - x_i|,$$

we have

$$\begin{aligned} |h_1^{(i,j)}| &> |\frac{1}{kS_x^2} \sum_{s=0}^{\ell} \sum_{t=-\ell}^{0} (x_{i+s} - x_i)| \\ &= O(n/k). \end{aligned}$$

Hence (A.4) is also true.

By the geometric symmetry, the other cases can be discussed in the same way. We finish the proof.

ACKNOWLEDGMENTS

We thank the editor, the associate editor, and three anonymous referees for their careful reading and many helpful comments that greatly improved the article.

[Received June 1996. Revised October 1996.]

REFERENCES

Besag, J. (1986), "On the Statistical Analysis of Dirty Pictures" (with discussion), *Journal of the Royal Statistical Society,* Ser. B, 48, 259–302.

Bhandarkar, S.M., Zhang, Y., and Potter, W.D. (1994), "An Edge Detection Technique Using Genetic Algorithm-Based Optimization," *Pattern Recognition,* 27, 1159–1180.

Brodsky, B.E., and Dharkhovsky, B.S. (1993), *Nonparametric Methods in Change-Point Problems*, The Netherlands: Kluwer Academic Publishers.

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

Chen, X.R. (1988), "Hypothesis Testing and Interval Estimation About the Model Which has Only One Change-Point," *Scientia Sinica,* 8A, 1–13.

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.

Chu, C.K. (1994), "Estimation of Change-Points in a Nonparametric Regression Function Through Kernel Density Estimation," *Communications in Statistics—Theory and Methods,* 23, 3037–3062.

Eubank, R.L., and Speckman, P.L. (1994), "Nonparametric Estimation of Functions With Jump Discontinuities," in *IMS Lecture Notes Change-Point Problems,* vol. 23, eds. E. Carlstein, H.G. Müller, and D. Siegmund, Hayward, CA: IMS, 130–144.

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.

Godtliebsen, F., and Chu, C.K. (1995), "Estimation of the Number of True Gray Levels, Their Values, and Relative Frequencies in a Noisy Image," *Journal of the American Statistical Association,* 90, 890–899.

Gonzalez, R.C., and Woods, R.E. (1992), *Digital Image Processing*, Reading, MA: Addison–Wesley.

Hall, P., and Titterington, M. (1992), "Edge-Preserving and Peak-Preserving Smoothing," *Technometrics,* 34, 429–440.

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.

Johnson, R.P. (1990), "Contrast Based Edge Detection," *Pattern Recognition,* 23, 311–318.

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

Lai, T.L., Robbins, H., and Wei, C.Z. (1979), "Strong Consistency of Least Squares Estimates in Multiple Regression II," *Journal of Multivariate Analysis,* 9, 343–362.

Loader, C.R. (1994), "Change Point Estimation Using Nonparametric Regression," AT&T Bell Laboratories.

Marr, D., and Hildreth, E. (1980), "Theory of Edge Detection," *Proceedings of the Royal Statistical Society of London,* 207, 187–217.

McDonald, J.A., and Owen, A.B. (1986), "Smoothing With Split Linear Fits," *Technometrics,* 28, 195–208.

Müller, H.G. (1992), "Change-Points in Nonparametric Regression Analysis," *The Annals of Statistics,* 20, 737–761.

Müller, H.G., and Song, K.S. (1994), "Maximin Estimation of Multidimensional Boundaries," *Journal of 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. (1994), "Estimation of the Number of Jumps of the Jump Regression Functions," *Communications in Statistics—Theory and Methods,* 23, 2141–2155.

——— (1997), "Nonparametric Estimation of Jump Surface," *Sankhya,* Series A, June issue.

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. (1994), "A Local Polynomial Jump Detection Algorithm in Nonparametric Regression," University of Wisconsin-Madison, Dept. of Statistics (tentatively accepted by *Technometrics*).

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

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.

Saint-Marc, P., Chen, J., and Medioni, G. (1991), "Adaptive Smoothing: A General Tool for Early Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 13, 514–529.

Sarkar, S.S., and Boyer, K.L. (1990), "On Optimal Infinite Impulse Response Edge Detection Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 12, 1154–1171.

Shiau, J.H.(1985), "Smoothing Spline Estimation of Functions With Discontinuities," technical report 768, University of Wisconsin-Madison, Dept. of Statistics.

——— (1987), "A Note on MSE Coverage Intervals in a Partial Spline Model," *Communications in Statistics—Theory and Methods,* 16, 1851–1866.

Shiau, J.H., Wahba, G., and Johnson, D.R. (1986), "Partial Spline Models for the Inclusion of Tropopause and Frontal Boundary Information in Otherwise Smooth Two- and Three-Dimensional Objective Analysis," *Journal of Atmospheric and Oceanic Technology,* 3, 714–725.

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.

Speckman, P.L. (1993), "Detection of Change-Points in Nonparametric Regression," University of Missouri, Dept. of Statistics.

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.

——— (1991), "A Cost Minimization Approach to Edge Detection Using Simulated Annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14, 3–18.

Torre, V., and Poggio, T.A. (1986), "On Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 8, 147–163.

Wang, Y. (1995), "Jump and Sharp Cusp Detection by Wavelets," *Biometrika,* 82, 385–397.

Wu, J.S., and Chu, C.K. (1993a), "Kernel Type Estimators of Jump Points and Values of a Regression Function," *The Annals of Statistics,* 21, 1545–1566.

——— (1993b), "Nonparametric Function Estimation and Bandwidth Selection for Discontinuous Regression Functions," *Statistica Sinica,* 3, 557–576.

Yin, Y.Q. (1988), "Detecting of the Number, Locations and Magnitudes of Jumps," *Communications in Statistics—Stochastic Models,* 4, 445–455.