

Jump Detection In Blurred Regression Surfaces

Yicheng Kang and Peihua Qiu

Department of Biostatistics, University of Florida

Abstract

We consider the problem of detecting jump location curves of regression surfaces when they are spatially blurred and contaminated pointwise by random noise. This problem is common in various applications, including equi-temperature surface estimation in meteorology and oceanography and edge detection in image processing. In the literature, most existing jump detection methods are developed under the assumption that there is no blurring involved, or that the blurring mechanism described by a point spread function is completely specified. In this paper, we propose four possible jump detectors, without imposing restrictive assumptions on either the point spread function or the true regression surface. Their theoretical and numerical properties are studied and compared. We also propose a new quantitative metric for measuring the performance of a jump detector. A data-driven bandwidth selection procedure via the bootstrap is suggested as well. This paper has supplementary material online.

Key Words: Denoising; Edge detection; Gradient; Jump location curves; Nonparametric regression; Performance measure; Surface estimation.

1 Introduction

Surface estimation is an important problem in many scientific areas, including image processing, geology, meteorology, oceanography, and so forth. In many surface estimation problems, the true surfaces have jumps and discontinuities, which are often called edges of the related image intensity surface in the image processing literature (cf., Gonzalez and Woods 1992, Qiu 2005). Jumps in a surface often represent outlines of objects (e.g., the boundary locations of image objects), or places where the structure of a 2-D process changes abruptly (e.g., places where an equi-temperature surface in high sky or deep ocean changes dramatically). Therefore, it is important to detect the jump locations of regression surfaces accurately from the observed data. In practice, however, the

observed data are usually not exactly the same as the true surfaces, but are degraded versions of their true surfaces. There are many different sources of degradations. For instance, in aerial reconnaissance, astronomy, and remote sensing, image intensity surfaces are often degraded by atmospheric turbulence, aberrations of the optical system, or relative motion between a camera and an object. Among different degradations, *pointwise degradation* (or, random noise) and *spatial degradation* (or, blur) are the most common in practice. As a demonstration, Figure 1(a)-(c) show a true image with a step edge at $x = 0.5$, a blurred version, and a blurred-noisy version, respectively. The cross section of the blurred noisy version at $y = 0.5$ is shown in 1(d). From these plots, it can be seen that noise alters individual gray levels in a random pattern but the alteration by blur is spatially correlated. From Figure 1(b)-(d), it can also be seen that the step edge of the original image is still visible in its blurred or blurred-noisy version, although the edge is tapered by some smoothing process that is often unknown in practice and such a smoothing process can vary over location as well (e.g., atmospheric turbulence often causes spatially variant blur in satellite images because both its direction and intensity are usually different at different locations). Other types of degradations involve chromatic or temporal effects. See Bates and McDonnell (1986) for a detailed discussion about formation of various degradations. This paper focuses on jump detection in regression surfaces when both spatial blur and pointwise noise are present in the observed data.

In the literature, most existing jump detection methods assume that there is no blurring involved in the observed data. These methods are usually based on appropriate estimation of the first-order and/or the second-order derivatives of the regression surfaces. See, for instance, Bowman *et al.* (2006), Chu *et al.* (2012), Ferger (2004), Garlipp and Müller (2006, 2007), Hall *et al.* (2008), Qiu (2002), Sun and Qiu (2007), and Wang (1998) in the statistical literature; and Canny (1986), Clark (1989), Fleck (1992), Heath *et al.* (1998), Marr and Hildreth (1980), Qiu and Bhandarkar (1996), and Torre and Poggio (1986) in the image processing literature.

In this paper, we propose four jump detectors in cases when the observed data contain both spatial blur and pointwise noise. The idea behind our proposed methods can be explained intuitively as follows. First, although spatial blur would alter the regression surface structure, especially around the true jump location curves, the blurred regression surface would change most dramatically at the true jump location curves along their normal directions (cf., 1(b)-(d)). Therefore,

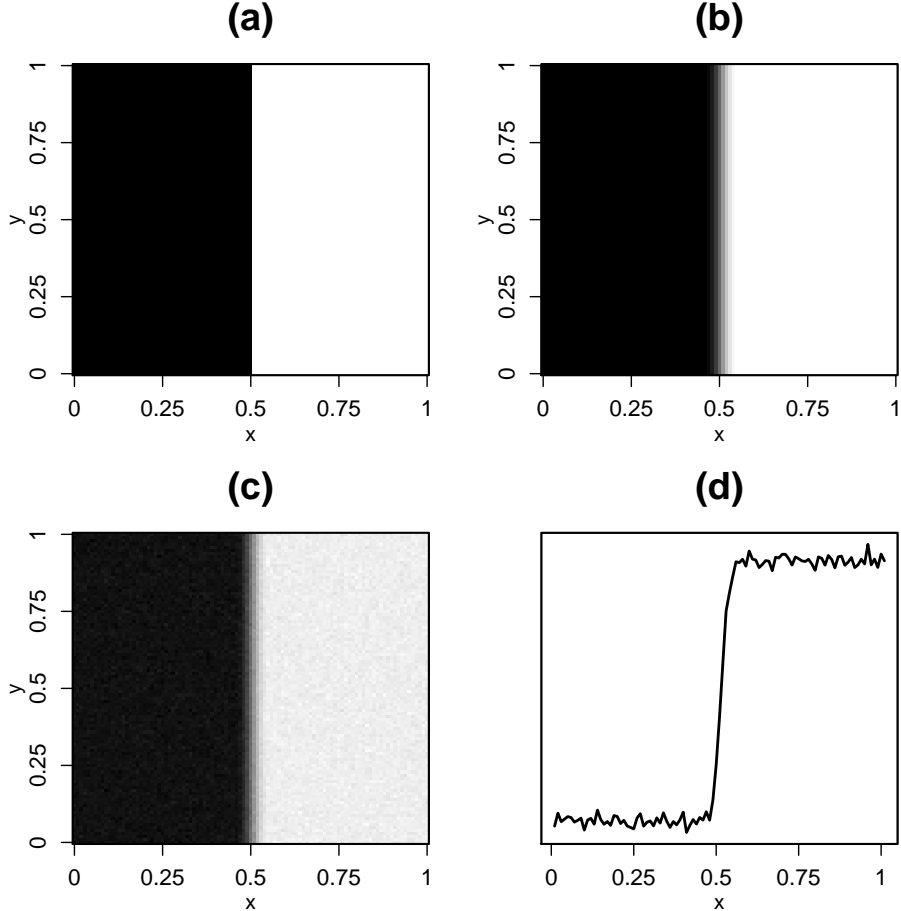


Figure 1: (a)-(c): Original image with a single step edge, a blurred version and a blurred-noisy version, respectively. (d): The cross section of the image in (c) at $y = 0.5$.

appropriate estimators of the surface gradients can be used for jump detection. Second, spatial blur is a smoothing process; it smooths the regression surface at the true jump locations. If we can recover the jump structure to a certain degree when removing noise, it may help detect jumps. Our proposed jump detectors make use of these properties of spatial blur in different ways. One major feature of these jump detectors is that they do not require restrictive assumptions on either the point spread function that describes the blurring mechanism or the true regression surface. Their theoretical and numerical properties are studied and compared. Further, in this paper, we propose a new quantitative metric for measuring the performance of a jump detector. A data-driven bandwidth selection procedure via the bootstrap is suggested as well.

The remainder of the article is organized as follows: In the next section, our proposed jump

detectors are described in detail. Some of their statistical properties are presented in Section 3. Their numerical performance is evaluated in Section 4. Several remarks conclude the article in Section 5. Some technical details are given in a supplementary file.

2 Proposed Jump Detectors

In this section, we first describe our proposed jump detectors in detail in Subsection 2.1, and then propose a new metric for measuring the jump detection performance in Subsection 2.2. Based on the proposed performance measure, a bootstrap bandwidth selection procedure is suggested in that subsection as well.

2.1 Four jump detectors based on local kernel smoothing

In the image processing literature, a commonly used model for describing the relationship between a true regression surface f and its observed degraded version Z is as follows.

$$Z(x, y) = G\{f\}(x, y) + \varepsilon(x, y), \quad \text{for } (x, y) \in \Omega, \quad (1)$$

where $G\{f\}(x, y) = \int \int_{\mathbf{R}^2} g(u, v; x, y) f(x - u, y - v) \, dudv$ denotes the convolution between a 2-D *point spread function* (psf) g (see Section 3 for a detailed mathematical description) and the true regression function f , $\varepsilon(x, y)$ is the pointwise noise at (x, y) , and Ω is the design space of the surface. In model (1), it is assumed that the true regression function f is degraded spatially by g and pointwise by ε , that the spatial blur is linear in the sense that $G\{a_1 f_1 + a_2 f_2\} = a_1 G\{f_1\} + a_2 G\{f_2\}$, where a_1 and a_2 are two arbitrary constants and f_1 and f_2 are any two images, and that the pointwise noise is additive. In most references, it is further assumed that the psf g , which describes the spatial blurring mechanism, is location invariant. That is, $g(u, v; x, y)$ does not depend on (x, y) . See Hall and Qiu (2007) for a related discussion.

For simplicity, let us assume that the design space is $\Omega = [0, 1] \times [0, 1]$, $\{(x_i, y_j) = (i/n, j/n), i, j = 1, 2, \dots, n\}$ are equally spaced design points in Ω , and $\{(x_i, y_j, Z_{ij}), i, j = 1, 2, \dots, n\}$ follow the model

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

where $\{\varepsilon_{ij}\}$ are i.i.d. random errors with mean 0 and unknown variance σ^2 . For a given point $(x, y) \in H_n = [h_n, 1 - h_n] \times [h_n, 1 - h_n]$, where $h_n \in (0, 1/2)$ is a bandwidth parameter, we consider its circular neighborhood:

$$O_n(x, y) = \left\{ (u, v) : (u, v) \in \Omega \text{ and } \sqrt{(u-x)^2 + (v-y)^2} \leq h_n \right\}.$$

In this neighborhood, let us consider the following local linear kernel smoothing procedure:

$$\min_{a, b, c} \sum_{(x_i, y_j) \in O_n(x, y)} \{Z_{ij} - [a + b(x_i - x) + c(y_j - y)]\}^2 K_{ij}, \quad (3)$$

where $K_{ij} := K((x_i - x)/h_n, (y_j - y)/h_n)$ and K is a circularly symmetric bivariate density kernel function defined on the unit disk centered at the origin. Let the solution to $\{a, b, c\}$ of the problem (3) be denoted as $\{\hat{a}(x, y), \hat{b}(x, y), \hat{c}(x, y)\}$. Then, $(\hat{b}(x, y), \hat{c}(x, y))$ is an estimator of the gradient of the true regression surface f at (x, y) . Now, we divide $O_n(x, y)$ into two halves, denoted as $U_n(x, y)$ and $V_n(x, y)$, along the direction perpendicular to $(\hat{b}(x, y), \hat{c}(x, y))$. More specifically, we define

$$\begin{aligned} U_n(x, y) &= \{(x_i, y_j) \in O_n(x, y) : (x_i - x)\hat{b}(x, y) + (y_j - y)\hat{c}(x, y) \geq 0\}, \\ V_n(x, y) &= \{(x_i, y_j) \in O_n(x, y) : (x_i - x)\hat{b}(x, y) + (y_j - y)\hat{c}(x, y) < 0\}. \end{aligned}$$

The change in the values of f from $V_n(x, y)$ to $U_n(x, y)$ should be relatively large if (x, y) is on a jump location curve. Let

$$\begin{aligned} \hat{f}_{LCK,+}(x, y) &= \frac{\sum_{(x_i, y_j) \in U_n(x, y)} Z_{ij} K_{ij}}{\sum_{(x_i, y_j) \in U_n(x, y)} K_{ij}}, \\ \hat{f}_{LCK,-}(x, y) &= \frac{\sum_{(x_i, y_j) \in V_n(x, y)} Z_{ij} K_{ij}}{\sum_{(x_i, y_j) \in V_n(x, y)} K_{ij}} \end{aligned} \quad (4)$$

be the Nadaraya-Watson local constant kernel (LCK) estimators of $f(x, y)$, constructed from the observations in the two one-sided neighborhoods $U_n(x, y)$ and $V_n(x, y)$, respectively. Then, $\hat{f}_{LCK,+}(x, y) - \hat{f}_{LCK,-}(x, y)$ would be a good measure of the change in values of f along the direction from $V_n(x, y)$ to $U_n(x, y)$. Our first jump detection criterion is defined by its standardized version

$$\text{LCK}_n(x, y) = \frac{\hat{f}_{LCK,+}(x, y) - \hat{f}_{LCK,-}(x, y)}{\sqrt{\frac{\sum_{(x_i, y_j) \in U_n(x, y)} K_{ij}^2}{[\sum_{(x_i, y_j) \in U_n(x, y)} K_{ij}]^2} + \frac{\sum_{(x_i, y_j) \in V_n(x, y)} K_{ij}^2}{[\sum_{(x_i, y_j) \in V_n(x, y)} K_{ij}]^2}}},$$

and the corresponding jump detector is denoted as LCK. Theoretically, it can be checked that the jump detection criterion $\text{LCK}_n(x, y)$ has the following two properties (proofs are given in the supplementary material):

(i) if f is continuous around (x, y) , then we have

$$\text{LCK}_n(x, y) \xrightarrow{d} N(0, \sigma^2), \text{ as } n \rightarrow \infty \quad (5)$$

(ii) if f has a jump at (x, y) , then we have

$$\text{LCK}_n(x, y) \xrightarrow{P} \infty, \text{ as } n \rightarrow \infty. \quad (6)$$

Therefore, the criterion $\text{LCK}_n(x, y)$ contains useful information for jump detection. Our proposed jump detector LCK can be summarized as follows.

- At a given design point (x_i, y_j) , for $i, j = 1, 2, \dots, n$, solve the minimization problem (3) and obtain the solution vector $\{\widehat{b}(x_i, y_j), \widehat{c}(x_i, y_j)\}$.
- Divide $O_n(x_i, y_j)$ into two halves $U_n(x_i, y_j)$ and $V_n(x_i, y_j)$ along the direction perpendicular to the estimated gradient vector $\{\widehat{b}(x_i, y_j), \widehat{c}(x_i, y_j)\}$. Compute the Nadaraya-Watson LCK estimators of $f(x, y)$ from the observations in $U_n(x_i, y_j)$ and $V_n(x_i, y_j)$, respectively, as described in (4).
- The design point (x_i, y_j) is flagged as a detected jump point if

$$|\text{LCK}_n(x_i, y_j)| > Z_{1-\alpha_n} \sigma,$$

where $Z_{1-\alpha_n}$ is the $(1 - \alpha_n)$ -th quantile of the standard normal distribution and α_n is a significance level.

In practice, σ is often unknown, and it needs to be estimated from the observed data. To this end, it can be estimated by the conventional kernel smoothing approach in a chosen region in which the regression surface is relatively smooth. Or, σ^2 can be estimated by the residual mean squares of the jump-preserving surface estimation procedure suggested by Qiu (2004).

Note that the jump detector LCK does not take into account the possible blurring in the observed data. As illustrated in Figure 1(b), blurring alters the image most significantly around a jump location curve (JLC, see its formal definition in Qiu (2005)) and least significantly at places where the intensity surface is straight. So, in cases when blurring is present and (x, y) is on a JLC, if a design point in $O_n(x, y)$ is closer to the line that divides $O_n(x, y)$ into $U_n(x, y)$ and $V_n(x, y)$,

then it is more likely that the corresponding observation Z_{ij} has blurring involved. Thus, it should receive smaller weight in the related weighted average. To address this issue, we suggest using a univariate kernel function L for assigning such weights. By this idea, our second jump detection criterion is defined by

$$\text{LC2K}_n(x, y) = \frac{\widehat{f}_{\text{LC2K},+}(x, y) - \widehat{f}_{\text{LC2K},-}(x, y)}{\sqrt{\frac{\sum_{(x_i, y_j) \in U_n(x, y)} K_{ij}^2 L_{ij}^2}{\left[\sum_{(x_i, y_j) \in U_n(x, y)} K_{ij} L_{ij}\right]^2} + \frac{\sum_{(x_i, y_j) \in V_n(x, y)} K_{ij}^2 L_{ij}^2}{\left[\sum_{(x_i, y_j) \in V_n(x, y)} K_{ij} L_{ij}\right]^2}},$$

where

$$\begin{aligned} \widehat{f}_{\text{LC2K},+}(x, y) &= \frac{\sum_{(x_i, y_j) \in U_n(x, y)} Z_{ij} K_{ij} L_{ij}}{\sum_{(x_i, y_j) \in U_n(x, y)} K_{ij} L_{ij}}, \\ \widehat{f}_{\text{LC2K},-}(x, y) &= \frac{\sum_{(x_i, y_j) \in V_n(x, y)} Z_{ij} K_{ij} L_{ij}}{\sum_{(x_i, y_j) \in V_n(x, y)} K_{ij} L_{ij}}, \end{aligned} \quad (7)$$

$L_{ij} := L(d_{ij}/h_n)$, L is a univariate increasing density kernel function with support $[0, 1]$, and d_{ij} is the Euclidean distance from the point (x_i, y_j) to the line that divides $O_n(x, y)$ into $U_n(x, y)$ and $V_n(x, y)$. The corresponding jump detector is labeled as LC2K, where LC2K denotes local constant smoothing with 2 kernel functions. The jump detection criterion $\text{LC2K}_n(x, y)$ has similar asymptotic results to (5) and (6). Also, a numerical algorithm of the jump detector LC2K can be developed in a similar way to that of the jump detector LCK.

In cases when the true regression surface f is quite steep but continuous around a given point (x, y) , by the jump detectors LCK and $LC2K$, (x, y) can be falsely detected as a jump point because both their jump detection criteria $LCK_n(x, y)$ and $\text{LC2K}_n(x, y)$ are good estimators of the directional derivative of f along the gradient direction at (x, y) . To overcome this limitation, in the one-sided neighborhoods $U_n(x, y)$ and $V_n(x, y)$, instead of computing the Nadaraya-Watson LCK estimators (cf., (4)), we consider fitting local planes by solving the following local linear kernel (LLK) smoothing problems:

$$\min_{a_0, a_1, a_2} \sum_{(x_i, y_j) \in U_n(x, y)} \{Z_{ij} - [a_0 + a_1(x_i - x) + a_2(y_j - y)]\}^2 K_{ij}, \quad (8)$$

and

$$\min_{a_0, a_1, a_2} \sum_{(x_i, y_j) \in V_n(x, y)} \{Z_{ij} - [a_0 + a_1(x_i - x) + a_2(y_j - y)]\}^2 K_{ij}. \quad (9)$$

The solutions to a_0 of (8) and (9) are denoted as $\widehat{f}_{LLK,+}(x, y)$ and $\widehat{f}_{LLK,-}(x, y)$, respectively, and they have the expressions

$$\begin{aligned}\widehat{f}_{LLK,+}(x, y) &= \frac{\sum_{(x_i, y_j) \in U_n(x, y)} w_{ij}(x, y) Z_{ij}}{\sum_{(x_i, y_j) \in U_n(x, y)} w_{ij}(x, y)} \\ \widehat{f}_{LLK,-}(x, y) &= \frac{\sum_{(x_i, y_j) \in V_n(x, y)} w'_{ij}(x, y) Z_{ij}}{\sum_{(x_i, y_j) \in V_n(x, y)} w'_{ij}(x, y)},\end{aligned}$$

where

$$\begin{aligned}w_{ij}(x, y) &= [A_1(x, y) + A_2(x, y)(x_i - x) + A_3(x, y)(y_j - y)]K_{ij}, \\ A_1(x, y) &= r_{20}(x, y)r_{02}(x, y) - r_{11}(x, y)r_{11}(x, y), \\ A_2(x, y) &= r_{01}(x, y)r_{11}(x, y) - r_{10}(x, y)r_{02}(x, y), \\ A_3(x, y) &= r_{10}(x, y)r_{11}(x, y) - r_{01}(x, y)r_{20}(x, y), \\ r_{s_1, s_2}(x, y) &= \sum_{(x_i, y_j) \in U_n(x, y)} (x_i - x)^{s_1} (y_j - y)^{s_2} K_{ij}, \quad \text{for } s_1, s_2 = 0, 1, 2,\end{aligned}$$

and $w'_{ij}(x, y)$ is defined to be the same as $w_{ij}(x, y)$ except that $U_n(x, y)$ needs to be replaced by $V_n(x, y)$ throughout. Then, we define our third jump detection criterion by

$$\text{LLK}_n(x, y) = \frac{\widehat{f}_{LLK,+}(x, y) - \widehat{f}_{LLK,-}(x, y)}{\sqrt{\frac{\sum_{(x_i, y_j) \in U_n(x, y)} w_{ij}(x, y)^2}{[\sum_{(x_i, y_j) \in U_n(x, y)} w_{ij}(x, y)]^2} + \frac{\sum_{(x_i, y_j) \in V_n(x, y)} w'_{ij}(x, y)^2}{[\sum_{(x_i, y_j) \in V_n(x, y)} w'_{ij}(x, y)]^2}}}.$$

It can be checked that, in cases when f is continuous at (x, y) , the value of $|\text{LLK}_n(x, y)|$ would be small, even when f is steep around (x, y) , because the slope effect has been accommodated in fitting the local planes in (8) and (9). See, for instance, Qiu (2004) for a related discussion.

Obviously, the jump detection criterion $\text{LLK}_n(x, y)$ has not taken into account the blurring effect, as discussed above about the jump detector LCK. By combining the ideas of the jump detection criteria $\text{LC2K}_n(x, y)$ and $\text{LLK}_n(x, y)$, we define the fourth jump detection criterion by

$$\text{LL2K}_n(x, y) = \frac{\widehat{f}_{LL2K,+}(x, y) - \widehat{f}_{LL2K,-}(x, y)}{\sqrt{\frac{\sum_{(x_i, y_j) \in U_n(x, y)} \tilde{w}_{ij}(x, y)^2}{[\sum_{(x_i, y_j) \in U_n(x, y)} \tilde{w}_{ij}(x, y)]^2} + \frac{\sum_{(x_i, y_j) \in V_n(x, y)} \tilde{w}'_{ij}(x, y)^2}{[\sum_{(x_i, y_j) \in V_n(x, y)} \tilde{w}'_{ij}(x, y)]^2}}},$$

where $\widehat{f}_{LL2K,+}(x, y)$ and $\widehat{f}_{LL2K,-}(x, y)$ are respectively the solutions to a_0 of the following local weighted least square problems:

$$\min_{a_0, a_1, a_2} \sum_{(x_i, y_j) \in U_n(x, y)} \{Z_{ij} - [a_0 + a_1(x_i - x) + a_2(y_j - y)]\}^2 K_{ij} L_{ij}, \quad (10)$$

$$\min_{a_0, a_1, a_2} \sum_{(x_i, y_j) \in V_n(x, y)} \{Z_{ij} - [a_0 + a_1(x_i - x) + a_2(y_j - y)]\}^2 K_{ij} L_{ij}, \quad (11)$$

$$\begin{aligned} \tilde{w}_{ij}(x, y) &= \left[\tilde{A}_1(x, y) + \tilde{A}_2(x, y)(x_i - x) + \tilde{A}_3(x, y)(y_j - y) \right] K_{ij} L_{ij}, \\ \tilde{A}_1(x, y) &= \tilde{r}_{20}(x, y)\tilde{r}_{02}(x, y) - \tilde{r}_{11}(x, y)\tilde{r}_{11}(x, y), \\ \tilde{A}_2(x, y) &= \tilde{r}_{01}(x, y)\tilde{r}_{11}(x, y) - \tilde{r}_{10}(x, y)\tilde{r}_{02}(x, y), \\ \tilde{A}_3(x, y) &= \tilde{r}_{10}(x, y)\tilde{r}_{11}(x, y) - \tilde{r}_{01}(x, y)\tilde{r}_{20}(x, y), \\ \tilde{r}_{s_1, s_2}(x, y) &= \sum_{(x_i, y_j) \in U_n(x, y)} (x_i - x)^{s_1} (y_j - y)^{s_2} K_{ij} L_{ij}, \text{ for } s_1, s_2 = 0, 1, 2, \end{aligned}$$

and $\tilde{w}'_{ij}(x, y)$ is defined to be the same as $\tilde{w}_{ij}(x, y)$, except that $U_n(x, y)$ needs to be replaced by $V_n(x, y)$ throughout.

For the jump detection criteria $LLK_n(x, y)$ and $LL2K_n(x, y)$, we can derive their asymptotic results similar to those in (5) and (6). Their corresponding jump detection procedures are denoted as LLK and LL2K, respectively, and are briefly summarized as follows.

- At a given design point (x_i, y_j) , for $i, j = 1, 2, \dots, n$, solve the minimization problem (3) and obtain the solution vector $\{\hat{b}(x_i, y_j), \hat{c}(x_i, y_j)\}$.
- Divide $O_n(x_i, y_j)$ into two halves $U_n(x_i, y_j)$ and $V_n(x_i, y_j)$ along the direction that is perpendicular to the estimated gradient vector $\{\hat{b}(x_i, y_j), \hat{c}(x_i, y_j)\}$. Compute $\hat{f}_{LLK,+}(x, y)$ and $\hat{f}_{LLK,-}(x, y)$ (or, $\hat{f}_{LL2K,+}(x, y)$ and $\hat{f}_{LL2K,-}(x, y)$) from (8) and (9) (or, (10) and (11)).
- The design point (x_i, y_j) is flagged as a detected jump point if the jump detector LLK is used and

$$|LLK_n(x_i, y_j)| > Z_{1-\alpha_n} \hat{\sigma}$$

or, the jump detector LL2K is used and

$$|LL2K_n(x_i, y_j)| > Z_{1-\alpha_n} \hat{\sigma}$$

where $\hat{\sigma}$ is an appropriate estimator of σ .

2.2 Selection of the bandwidth

The proposed jump detectors described in Subsection 2.1 have the bandwidth parameter h_n involved. In 1-D cases, Gijbels and Goderniaux (2004) demonstrated that this parameter plays an important role in their 1-D jump detection procedure. Based on our numerical experience, this is also true for the current 2-D procedure. In this subsection, we propose a procedure for choosing h_n in 2-D cases.

The 2-D bandwidth selection problem is more complicated than its 1-D counterpart. In order to choose the bandwidth properly in 2-D cases, we first need to choose a metric for measuring the distance between the pointset S of the true jump points and its estimator \widehat{S}_n by a jump detector with a given bandwidth h_n . Qiu (2002) investigated this performance measure problem carefully, and suggested the following performance measure:

$$d_Q(\widehat{S}_n, S; h_n) = \omega \frac{|\widehat{S}_n \setminus S|}{|\Omega \setminus S|} + (1 - \omega) \frac{|S \setminus \widehat{S}_n|}{|S|},$$

where $0 \leq \omega \leq 1$ is a weight, Ω denotes the entire design space, $A \setminus B$ denotes the set of points in A but not in B , and $|A|$ denotes the number of design points in the point set of A . Clearly, d_Q is a weighted average of the false positive rate and the false negative rate of the related jump detector, and the weight ω reflects the relative importance of the two rates. In an application, if the two rates are equally important, then ω can be simply chosen as 0.5. It should be pointed out that d_Q can be computed only when the pointset of the true jump points S is known.

In mathematics, a popular metric for measuring the distance between two pointsets A and B is the following Hausdorff distance:

$$d_H(A, B) = \max \left\{ \sup_{s_1 \in A} \inf_{s_2 \in B} d_E(s_1, s_2), \sup_{s_1 \in B} \inf_{s_2 \in A} d_E(s_1, s_2) \right\},$$

where $d_E(s_1, s_2)$ denotes the Euclidean distance between two points s_1 and s_2 . Qiu (2002) demonstrated that the above Hausdorff distance had two major limitations: (i) it was sensitive to individual points in the related pointsets, and (ii) it required extensive computations. As a comparison, the measure d_Q is easier to compute and more robust to individual points. However, we find that d_Q has its own limitations. For instance, assume that the true jump location curve is the line $y = 0.5$ and the detected jumps are those design points on the line $y = 0.5 + 1/n_1$, where n_1 is a

positive integer. Then, we would expect that the related jump detector performs better when n_1 gets larger. However, it can be checked that the value of d_Q does not depend on n_1 in such a case, which is intuitively unreasonable. To avoid this drawback, in this paper, we propose the following performance measure:

$$d_{KQ}(\widehat{S}_n, S; h_n) = \frac{\omega}{|\widehat{S}_n|} \sum_{(x', y') \in \widehat{S}_n} d_E((x', y'), S) + \frac{1-\omega}{|S|} \sum_{(x, y) \in S} d_E((x, y), \widehat{S}_n), \quad (12)$$

where $0 \leq \omega \leq 1$ is a weight, and $d_E((x', y'), S)$ denotes the Euclidean distance from the point (x', y') to the pointset S , i.e., the minimum of all pairwise distances between (x', y') and (w, z) for (w, z) in S . Obviously, d_{KQ} is a weighted average of two averages, where the first average in (12) is the average Euclidean distance from the individual detected jump points to the set of true jump points and the second average is the average Euclidean distance from the individual true jump points to the set of detected jump points. By using d_{KQ} , the drawback of the measure d_Q mentioned above is obviously overcome.

To demonstrate the strengths and limitations of the three performance measures discussed above, let us consider the following two examples. Without loss of generality, in both examples, ω is fixed at 0.5 and the sample size is fixed at $n = 100$. In the first example, it is assumed that there is a single JLC in the design space $[0, 1] \times [0, 1]$, and it is the line parallel to the x -axis at $y = 0.5$. We further assume that the detected jump points include all design points with $y = 0.5$ and a point at $(0.5, 0.5 + r)$, where $0 \leq r \leq 0.5$ is a constant. These detected jump points are shown in Figure 2(a) by the dark dots. The single outlying point affects the value of d_H dramatically. But, it does not affect either d_Q or d_{KQ} much. As a matter of fact, it can be checked that

$$d_H(\widehat{S}_n, S) = r, \quad d_Q(\widehat{S}_n, S) = \frac{0.5}{n^2 - n}, \quad d_{KQ}(\widehat{S}_n, S) = \frac{0.5r}{n + 1}.$$

When r changes from 0 to 0.5, $d_H(\widehat{S}_n, S)$, $d_Q(\widehat{S}_n, S)$, and $d_{KQ}(\widehat{S}_n, S)$ are shown in Figure 2(b) by the solid, dashed and dotted lines, respectively. In the plot, $d_{KQ}(\widehat{S}_n, S)$ is not visible because it almost overlaps with $d_Q(\widehat{S}_n, S)$. From the plot, it can be seen that $d_H(\widehat{S}_n, S)$ is indeed sensitive to the single detected jump point at $(0.5, 0.5 + r)$, $d_Q(\widehat{S}_n, S)$ does not depend on r , and $d_{KQ}(\widehat{S}_n, S)$ depends on r but it is almost unchanged with r because there is only one detected jump point moved when r changes. In this example, it is obvious that $d_H(\widehat{S}_n, S)$ does not reflect the jump detection performance well when r is large, and the other two performance measures are more reasonable to use.

Next, in the second example, let us assume that the detected jump points are those design points on the line at $y = 0.5 + t$ (shown in Figure 2(c)), where $0 \leq t \leq 0.5$ is a constant, and the remaining setup is the same as those in the first example. In this example, it can be checked that the three performance measures are

$$d_H(\widehat{S}_n, S) = t, \quad d_Q(\widehat{S}_n, S) = \frac{0.5n}{n^2 - n} + 0.5, \quad d_{KQ}(\widehat{S}_n, S) = t.$$

When t changes from 0 to 0.5, these performance measures are shown in Figure 2(d). Because $d_H(\widehat{S}_n, S)$ and $d_{KQ}(\widehat{S}_n, S)$ are exactly the same in this example, they are overlapped in the plot. From the plot, it can be seen that $d_Q(\widehat{S}_n, S)$ remains unchanged when t increases, which is unreasonable because the jump detection performance should become worse when t increases. As a comparison, both $d_H(\widehat{S}_n, S)$ and $d_{KQ}(\widehat{S}_n, S)$ reflect this fact well in their values. From this and the first examples, it seems that the major limitations of d_H and d_Q have been overcome by the proposed new jump detection performance measure d_{KQ} .

However, we should point out that d_{KQ} is not a mathematical metric and it is proposed for selecting bandwidth due to the reason that the mathematically well-defined Hausdorff distance is highly sensitive to individual points as demonstrated in Figure 2. Thus, the Hausdorff distance still needs to be adopted for characterizing asymptotic properties of jump detectors in Section 3, but d_{KQ} is used in Section 4 to evaluate numerical performance in finite-sample scenarios. In practice, S can be replaced by its discrete version $S^* = \{(x_i, y_j) : d((x_i, y_j), S) \leq 1/(2n)\}$ for the purpose of calculating d_{KQ} . In simulations, the point set S is usually known; thus, h_n can be chosen by minimizing $d_{KQ}(\widehat{S}_n, S; h_n)$. In practice, however, S is often unknown. In such cases, we propose a bootstrap procedure described below for choosing h_n .

Bootstrap Procedure For Choosing h_n

1. Apply a jump detector to the original dataset $\{(x_i, y_j, Z_{ij}), i, j = 1, 2, \dots, n\}$. The set of detected jump points is denoted as \widehat{S}_n .
2. Construct an estimator $\widehat{G\{f\}}(x, y)$ of the blurred regression function $G\{f\}(x, y)$ using a jump-preserving surface estimation method, and define the residuals by $\widehat{r}_{ij} = Z_{ij} - \widehat{G\{f\}}(x_i, y_j)$.

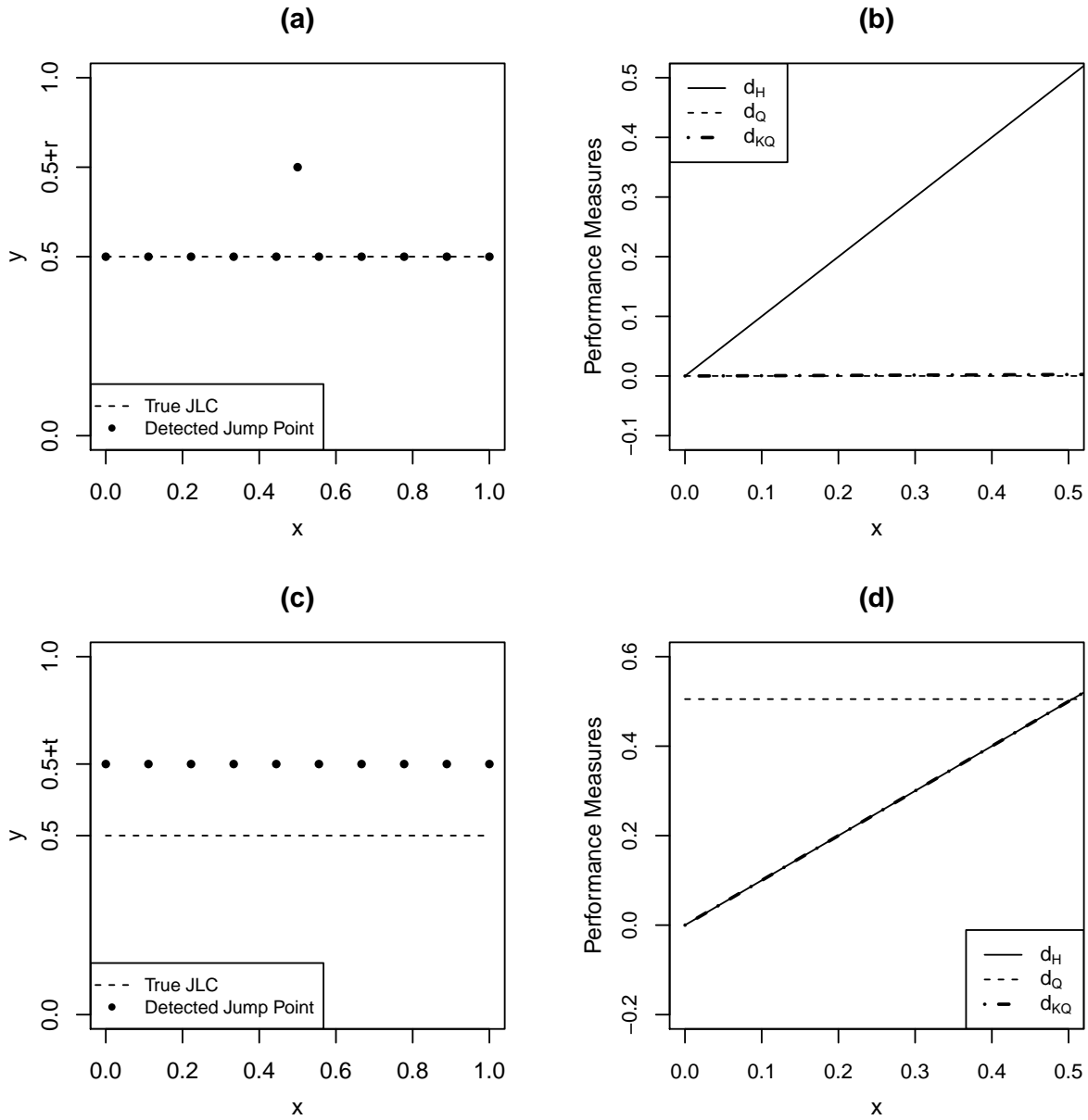


Figure 2: (a) Detected jump points are shown by the dark dots in the first example. (b) Three metrics for measuring the performance of the detected jump points in plot (a) when r changes from 0 to 0.5. (c) Detected jump points are shown by the dark dots in the second example. (d) Three metrics for measuring the performance of the detected jump points in plot (c) when t changes from 0 to 0.5. In both examples, the true jump location curves are the line $y = 0.5$ shown by the dashed lines in plots (a) and (c).

3. Draw with replacement a bootstrap sample of size n^2 , denoted as $\{\widehat{r}_{ij}^*, i, j = 1, 2, \dots, n\}$, from the residual set $\{\widehat{r}_{ij}, i, j = 1, 2, \dots, n\}$, and then obtain a bootstrap dataset $\{(x_i, y_j, Z_{ij}^*), i, j = 1, 2, \dots, n\}$, where $Z_{ij}^* = \widehat{G\{f\}}(x_i, y_j) + \widehat{r}_{ij}^*$.
4. The jump detector with the same bandwidth as that in step 1 is then applied to the bootstrap dataset, and a set of detected jump points from this bootstrap dataset can be obtained.
5. Steps 3 and 4 are repeated N times. The sets of the detected jump points from the N bootstrap datasets are denoted as $\widehat{S}_{1n}^*, \widehat{S}_{2n}^*, \dots, \widehat{S}_{Nn}^*$, respectively. Then, the bootstrap estimator of $d_{KQ}(\widehat{S}_n, S; h_n)$ is defined by

$$\widehat{d}_{KQ}^{BT}(\widehat{S}_n, S; h_n) = \frac{1}{N} \sum_{i=1}^N d_{KQ}(\widehat{S}_{in}^*, \widehat{S}_n; h_n).$$

The optimal bandwidth is then approximated by the minimizer of $\widehat{d}_{KQ}^{BT}(\widehat{S}_n, S; h_n)$ with respect to h_n .

In step 2 of the above procedure, $\widehat{G\{f\}}(x, y)$ can be constructed by any jump-preserving surface estimator proposed in the literature (cf., Hillebrand and Müller 2007, Qiu 2007). In all numerical examples presented in this paper, the local piecewise constant kernel estimator suggested in Qiu (2009) is used.

3 Statistical Properties

We discuss some statistical properties of the proposed jump detectors. In the literature, the psf $g(u, v; x, y)$ in model (2) is often assumed to be a density function (i.e., a non-negative function with a unit integration on its support), since it is believed that the blurring process does not change the mass (Bates and McDonnell 1986) of the regression surface. This conventional assumption is also adopted here. Further, we assume g is symmetric about both x-axis and y-axis, which in practice is often satisfied by common types of blur such as out-of-focus blur, motion blur or Gaussian blur. More specifically, we assume that, for any $(x, y) \in \Omega$, (i) $g(u, v; x, y) \geq 0$, for all $(u, v) \in \mathbf{R}^2$, (ii) $\int \int_{\mathbf{R}^2} g(u, v; x, y) dudv = 1$, (iii) $\int \int_{\mathbf{R}^2} ug(u, v; x, y) dudv = \int \int_{\mathbf{R}^2} vg(u, v; x, y) dudv = 0$. and (iv) $g(u, v; x, y) = 0$ if $\sqrt{u^2 + v^2} > \rho_n$, where ρ_n is a positive parameter denoting the blurring extent. Then, we have the following result about the proposed jump detectors.

Theorem 3.1. *Assume that the true regression function f has piecewise continuous third-order derivatives in each closed subset of $[0, 1] \times [0, 1]$; on each boundary curve of the pieces in which the first-order derivatives of f are continuous, the first-order derivatives of f have uniformly bounded one-sided limits defined in individual pieces; $E(\varepsilon_{11}^2) < \infty$; K is a Lipschitz-1 continuous circularly symmetric 2-D density kernel function with support $\{(x, y) : x^2 + y^2 \leq 1\}$; L is a univariate Lipschitz-1 continuous increasing density kernel function with support $[0, 1]$; the psf $g(u, v; x, y)$ is bounded; the blurring extent $\rho_n = o(1)$; the bandwidth h_n satisfies the conditions that $h_n = o(1)$, $\log(n)/(nh_n^4) = o(1)$ and $\rho_n/h_n = o(1)$; the significance level α_n satisfies the conditions that $\alpha_n = o(1)$ and $Z_{1-\alpha_n}/(nh_n) = o(1)$, $(nh_n^3)/Z_{1-\alpha_n} = o(1)$. Then, we have*

$$d_H(\widehat{S}_n \cap \Omega_{h_n} \cap J_{S, h_n}^C, S \cap \Omega_{h_n} \cap J_{S, h_n}^C) = O(h_n), a.s.,$$

where \widehat{S}_n is the set of the detected jump points by one of the four proposed jump detectors, $\Omega_{h_n} = \{(x, y) : (x, y) \in [h_n, 1 - h_n] \times [h_n, 1 - h_n]\}$, J_S includes all singular points in S , defined to be crossing points of several jump location curves or points on a single jump location curve at which at least one of the two one-sided tangent lines of the curve does not exist, or points on a single jump location curve at which the jump sizes in f are zero, $J_{S, h_n} = \{(x, y) : d_E((x, y), J_S) \leq h_n\}$, and $J_{S, h_n}^C = \Omega \setminus J_{S, h_n}$.

From Theorem 3.1, each of the four proposed jump detectors provides a consistent estimator of S in the sense that it converges almost surely to S in the Hausdorff distance, after certain small regions around the singular points in S and around the border of the design space are excluded. It should be pointed out that the assumption used in the theorem that the blurring extent ρ_n converges to 0 when n increases seems restrictive; but, it is actually quite flexible for the reason given below. Let $r_n = n\rho_n$. Then, r_n denotes the blurring extent in terms of the number of rows/columns of design points. The assumptions in Theorem 3.1 allows r_n to increase to infinity when n increases. Namely, when the resolution of the observed regression surface increases, our proposed methods allow the number of rows/columns of design points involved in the blurring at a given point to increase at a certain rate, which should be flexible enough for most applications. The proof of Theorem 3.1 is given in the supplementary material.

4 Numerical Studies

In this section, we present some numerical examples concerning the performance of the four proposed jump detectors. Throughout this section, if there is no further specification, the kernel function K used in (3) is chosen to be the truncated 2-D Gaussian density function $1/(2\pi - 3\pi \exp(-0.5))[\exp(-(x^2 + y^2)/2) - \exp(-0.5)]I_{x^2+y^2 \leq 1}$, the 2-D kernel function used in constructing each jump detection criterion (cf., e.g., (4)) is chosen to be the Epanechnikov kernel function $(2/\pi)(1 - x^2 - y^2)I_{x^2+y^2 \leq 1}$, and the 1-D kernel function L used in (7) and (10) is chosen to be $(1/1.194958) \exp(x^2/2)I_{0 \leq x \leq 1}$, which is proportional to the reciprocal of the truncated 1-D Gaussian kernel function. We choose these kernel functions because the Epanechnikov kernel function is a standard choice in the statistical literature and the truncated Gaussian kernel functions are commonly used in the image processing literature.

In the first example, the following true regression function is considered:

$$f_1(x, y) = \begin{cases} 3, & \text{if } (x - 0.5)^2 + (y - 0.5)^2 \leq 0.04 \\ 2, & \text{if } 0.04 < (x - 0.5)^2 + (y - 0.5)^2 \leq 0.09 \\ -20(\sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.4), & \text{if } 0.09 < (x - 0.5)^2 + (y - 0.5)^2 \leq 0.16 \\ 0, & \text{otherwise.} \end{cases}$$

As shown in Figure 3(a), f_1 has one circular jump location curve, and its surface is steep in the region $\{(x, y) : 0.09 < (x - 0.5)^2 + (y - 0.5)^2 < 0.16\}$.

In model (2), the psf is chosen to be

$$g(u, v; x, y) = \frac{3}{\pi \rho_n^2(x, y)} \left(1 - \frac{\sqrt{u^2 + v^2}}{\rho_n(x, y)} \right) I_{u^2+v^2 \leq \rho_n^2(x, y)}, \quad (13)$$

and the additive random errors ε_{ij} are generated from the distribution $N(0, \sigma^2)$. The above psf is circularly symmetric at each (x, y) with the blurring extent controlled by $\rho_n(x, y)$. In this example, we simply choose $\rho_n(x, y) \equiv 0.02$. In such cases, $g(u, v; x, y)$ is location invariant. Next, we try to measure the performance of the related jump detectors quantitatively using the proposed performance measure $d_{KQ}(\hat{S}_n, S; h_n)$, in which ω is fixed at 0.5 in this section. Without loss of generality, the bandwidth h_n is assumed to take the value of k/n , where $k < n$ is a positive integer. We consider two sample sizes 100×100 (i.e., $n = 100$) and 200×200 (i.e., $n = 200$), and three σ

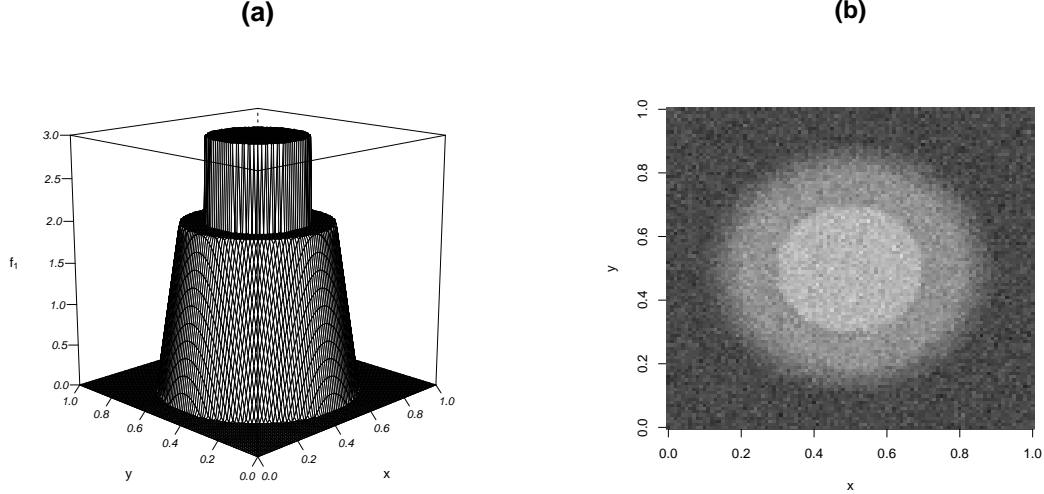


Figure 3: (a): The true regression function $f_1(x, y)$. (b): The observed image.

values 0.2, 0.3 and 0.4. The value of α_n is fixed at 0.001 when $n = 100$ and 0.0005 when $n = 200$. Figure 3(b) shows an observed image when $\sigma = 0.3$ and $n = 100$. Simulation results based on 100 replications are presented in Table 1. In the simulation, for each combination of n and σ and for each jump detector, the bandwidth h_n is chosen by minimizing the averaged value of the 100 values of $d_{KQ}(\widehat{S}_n, S; h_n)$ that are obtained from 100 replicated simulations. Such a value of h_n is called the optimal bandwidth because it provides a reasonable approximation to the minimizer of $\min_{h_n} E(d_{KQ}(\widehat{S}_n, S; h_n))$. Average bandwidths computed over 100 replications chosen by our proposed bootstrap procedure are also presented in the table.

From Table 1, it can be seen that: (i) the jump detectors based on the local linear kernel smoothing (i.e., LLK and LL2K) perform better than their counterparts based on the local constant kernel smoothing (i.e., LCK and LC2K), (ii) generally speaking, bandwidth for each jump detector should be chosen larger for noisier data, (iii) performance of the jump detectors based on the local constant kernel smoothing does not change much when noise level or sample size changes, (iv) the jump detectors based on the local linear kernel smoothing perform better when the sample size gets larger, (v) the bootstrap procedure tends to select the bandwidth slightly larger than the optimal value, and (vi) the bootstrap procedure gives close-to-optimal bandwidths when the sample size increases. The result (i) is intuitively reasonable because the regression surface is steep in certain regions in this example (cf., Figure 3(a)), the jump detectors based on the local linear kernel

Table 1: Numerical results for example 1 based on 100 replicated simulations. Each entry is (optimal bandwidth, bandwidth chosen by the proposed bootstrap procedure) and the averaged value of $d_{KQ}(\widehat{S}_n, S; h_n)$ when h_n is chosen to be the optimal bandwidth.

n	Method	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
100	LCK	(0.02, 0.03) 0.103	(0.02, 0.04) 0.109	(0.03, 0.04) 0.111
	LC2K	(0.02, 0.03) 0.109	(0.02, 0.03) 0.105	(0.03, 0.04) 0.113
	LLK	(0.07, 0.06) 0.009	(0.09, 0.13) 0.013	(0.10, 0.14) 0.030
	LL2K	(0.06, 0.10) 0.011	(0.07, 0.11) 0.013	(0.08, 0.12) 0.022
200	LCK	(0.01, 0.02) 0.106	(0.02, 0.04) 0.075	(0.02, 0.03) 0.092
	LC2K	(0.01, 0.02) 0.094	(0.02, 0.04) 0.085	(0.02, 0.04) 0.088
	LLK	(0.05, 0.05) 0.004	(0.06, 0.06) 0.005	(0.07, 0.07) 0.006
	LL2K	(0.04, 0.04) 0.006	(0.05, 0.05) 0.006	(0.06, 0.06) 0.008

smoothing can accommodate such a case well, but the jump detectors based on the local constant kernel smoothing would have many false jump detections due to the steep regression surface. The result (iii) is also caused by the false jump detections of the jump detectors based on the local constant kernel smoothing.

In cases when $n = 100$ and $\sigma = 0.3$, the detected jumps by the four jump detectors with their optimal bandwidths presented in Table 1 are shown in Figure 4, where the results are from the simulation whose $d_{KQ}(\widehat{S}_n, S; h_n)$ value is the median of the 100 $d_{KQ}(\widehat{S}_n, S; h_n)$ values computed from the 100 replications. From the figure, it can be seen that (i) the detected jumps in both plots (a) and (b) contain some false detections in the region where the true regression surface is steep, (ii) both the jump detectors LCK and LC2K fail to detect some true jump points, and (iii) the detected jumps in plot (d) seem to be a little more variable than those in plot (c). All these results are consistent with those in Table 1 and with our theoretical justifications discussed before. This example demonstrates that the jump detectors LLK and LL2K can accommodate the slope effect of the true regression surface well, while the jump detectors LCK and LC2K can not.

Next, we consider another example in which the true regression function is

$$f_2(x, y) = \begin{cases} (x - 0.2)^2 + 2, & \text{if } 0 \leq x \leq 0.2 \\ (x - 0.2)^2, & \text{if } 0.2 < x \leq 0.7 \\ (x - 0.7)^2 + 1.25, & \text{if } 0.7 < x \leq 1, \end{cases}$$

and the psf is defined by (13) with

$$\rho_n(x, y) = 0.1(1 - y).$$

Therefore, the true regression function f_2 has two JLCs, both are parallel to the y -axis at $x = 0.2$ and $x = 0.7$, and the jump size is larger (i.e., 2.0) for $x \leq 0.2$ and smaller (i.e., 1.0) for $x > 0.7$. The psf is location-dependent in this example, with blurring extent increasing when y gets smaller. The blurred version of f_2 is shown in Figure 5(a). A noisy and blurred version of f_2 with $n = 100$ and $\sigma = 0.3$ is shown in Figure 5(b).

In the same format as Table 1, the optimal bandwidths, the corresponding $d_{KQ}(\widehat{S}_n, S; h_n)$ values, and the bandwidths chosen by the bootstrap procedure, based on 100 replicated simulations, are presented in Table 2. From the table, it can be seen that (i) the four jump detectors all perform

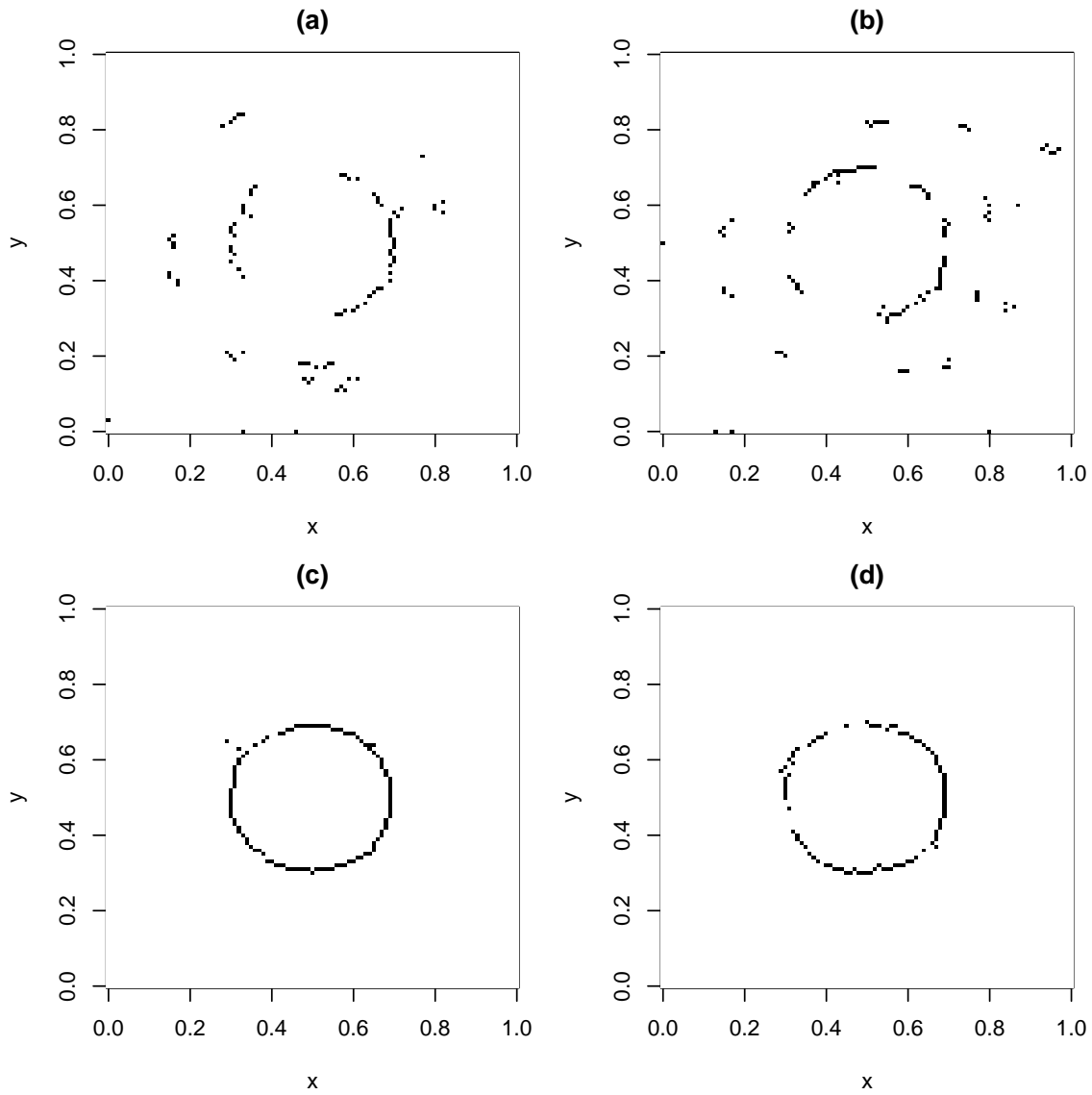


Figure 4: Detected jumps for one realization of the simulation in example 1, for each of the four jump detectors. (a): LCK; (b): LC2K; (c): LLK; (d): LL2K. See text for description of how the realizations were chosen.

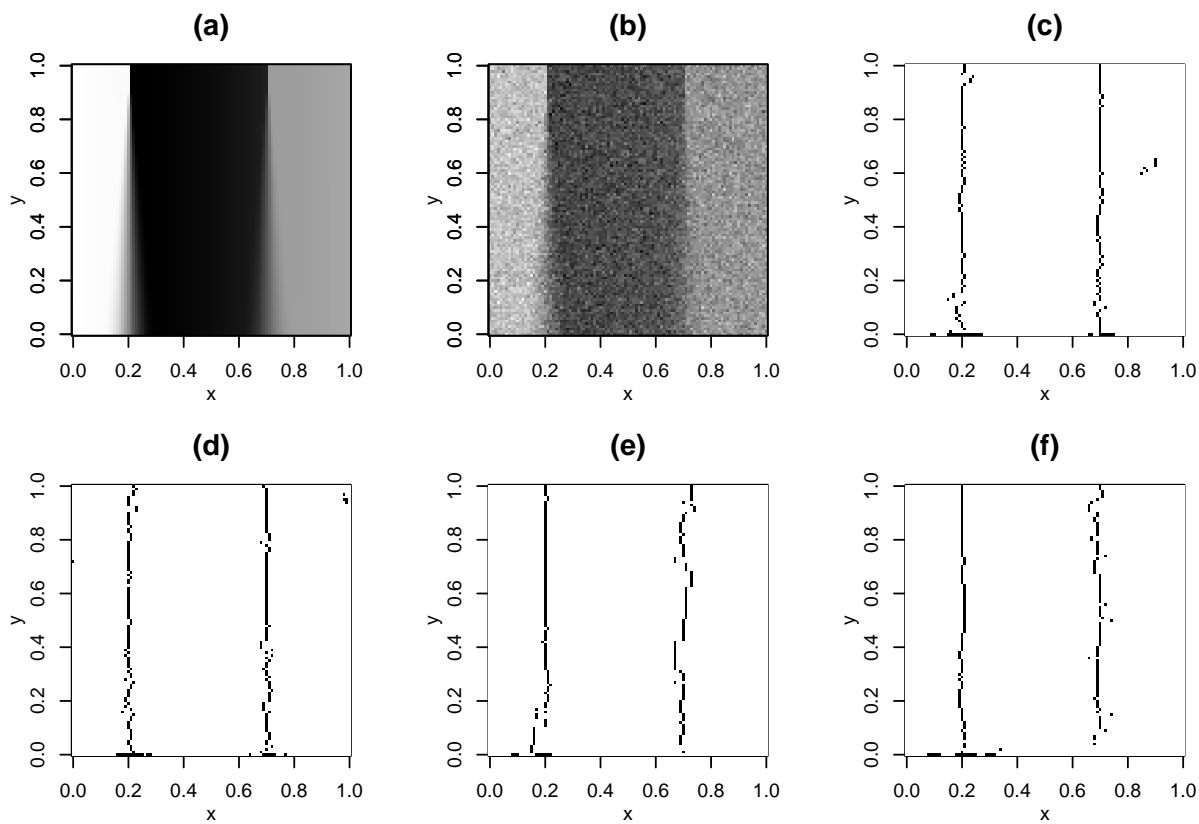


Figure 5: (a): The blurred version of f_2 ; (b): A noisy and blurred version of f_2 with $n = 100$ and $\sigma = 0.3$; (c)-(f): Detected jumps by the jump detectors LCK, LC2K, LLK and LL2K, respectively.

better when the sample size increases, (ii) the jump detectors LLK and LL2K are more sensitive to the noise level in the sense that their $d_{KQ}(\widehat{S}_n, S; h_n)$ values increase more rapidly as the noise level increases, compared to the jump detectors LCK and LC2K, (iii) for all four jump detectors, a larger bandwidth is chosen when the noise level is larger, and (iv) the bandwidths chosen by the bootstrap procedure are all quite close to the optimal bandwidths. The result (ii) is consistent with the theoretical result that the local linear kernel estimators would have a larger variability than the local constant kernel estimators, if their bandwidths are similar (cf., Qiu 2005, Chapter 2). The results (i) and (iii) are intuitively reasonable.

Table 2: Numerical results for example 2 based on 100 replicated simulations. Each entry is (optimal bandwidth, bandwidth chosen by the proposed bootstrap procedure) and the averaged value of $d_{KQ}(\widehat{S}_n, S; h_n)$ when h_n is chosen to be the optimal bandwidth.

n	Method	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
100	LCK	(0.03, 0.05) 0.016	(0.05, 0.06) 0.018	(0.05, 0.06) 0.018
	LC2K	(0.03, 0.05) 0.017	(0.05, 0.06) 0.019	(0.05, 0.07) 0.019
	LLK	(0.14, 0.13) 0.015	(0.15, 0.15) 0.020	(0.16, 0.16) 0.023
	LL2K	(0.12, 0.14) 0.016	(0.14, 0.14) 0.019	(0.15, 0.15) 0.024
200	LCK	(0.03, 0.04) 0.013	(0.03, 0.05) 0.014	(0.04, 0.05) 0.015
	LC2K	(0.03, 0.04) 0.014	(0.03, 0.05) 0.015	(0.04, 0.05) 0.016
	LLK	(0.11, 0.13) 0.010	(0.13, 0.14) 0.011	(0.14, 0.15) 0.014
	LL2K	(0.10, 0.11) 0.011	(0.11, 0.11) 0.012	(0.12, 0.12) 0.014

Simulation results shown in Figure 5(c)-(f) have the same setup as that in Figure 4. From the

plots, it can be seen that (i) all four jump detectors detect the larger jumps at $x = 0.2$ reasonably well, (ii) the jump detectors LCK and LC2K that are based on the local constant kernel smoothing detectors perform better than the jump detectors LLK and LL2K that are based on the local linear kernel smoothing when detecting the smaller jumps at $x = 0.7$, (iii) by comparing plots (c) and (d), it seems that the jump detector LC2K is slightly more robust to the blurring than the jump detector LCK, and (iv) by comparing plots (e) and (f), the jump detector LL2K seems more robust to the blurring than the jump detector LLK. This example demonstrates that (i) the jump detectors LC2K and LL2K using 2 kernel functions in their construction seem more robust to the blurring, compared to the other two jump detectors, and (ii) the results by the jump detectors LCK and LC2K seem less variable, compared to the jump detectors LLK and LL2K.

Next, we consider a real test image shown in Figure 6(a), which is synthetic aperture radar (SAR) image of an area near the Thetford forest in England. This image has 250×250 pixels with gray levels in the range $[0, 255]$. Its blurred and noisy version by the psf in (13) with $\rho_n(x, y) \equiv 0.02$ and by the i.i.d. noise from $N(0, 5^2)$ is shown in Figure 6(b). The detected jumps by the four jump detectors LCK, LC2K, LLK, and LL2K are presented in Figure 6(c)-(f), respectively. In all four jump detectors, α_n is fixed at 0.001, and their bandwidths are chosen by the bootstrap procedure to be $3/250$, $3/250, 11/250$ and $9/250$. From the plots, it can be seen that (i) all four jump detectors detect the major jump location curves reasonably well, (ii) the jump location curves detected by LCK and LC2K are thinner and more straight than those detected by LLK and LL2K; but, some false jumps are detected by them here and there, (iii) the jump detector LLK fails to detect certain jumps (e.g., the jumps located in the upper-right portion of the image), and (iv) the jump detector LL2K detects more true jumps than the jump detector LLK, which is consistent with our analysis in Section 2. Given that the noise level in the SAR image is relatively high, we suggest using jump detectors based on the local constant kernel smoothing (i.e., LCK or LC2K) in this example.

We conclude this section by considering an image shown in Figure 7. From the figure, it can be seen that objects within the focus (e.g., the pen) are quite sharp but objects in distance (e.g., the child's face) are blurred. So, this image provides us an example with spatially varying blur. By the jump-preserving image restoration procedure suggested by Qiu (2004), an estimate of σ is computed to be 4.558. The detected jump points by LCK, LC2K, LLK and LL2K are shown in Figure 8(a)-(d), respectively. From the plots, it can be seen that both LCK and LLK fail to detect

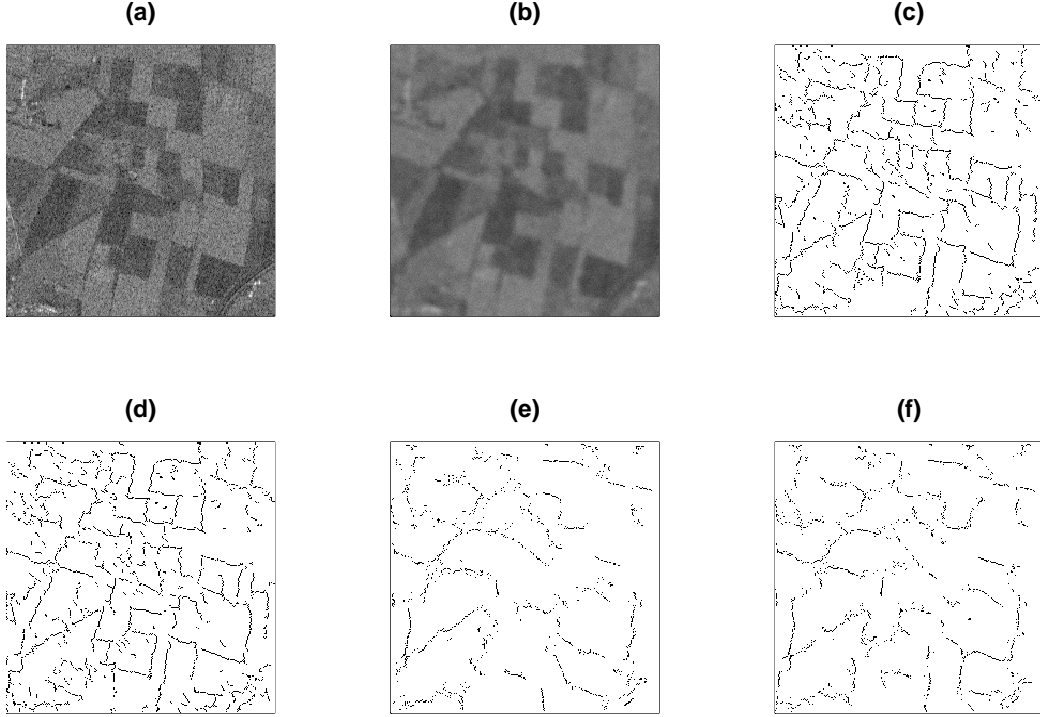


Figure 6: (a): Original SAR image; (b): A noisy and blurred version of the SAR image; (c)-(f): Detected jumps by detectors LCK, LC2K, LLK and LL2K, respectively.

some edges in the blurred region, which is especially true for LLK. The detected jump points by LC2K and LL2K look reasonably good. Because the blur is quite severe in this example, we suggest using the jump detector LC2K or the jump detector LL2K.

5 Conclusions

We have presented four jump detectors for detecting jumps in blurred regression surfaces. All four jump detectors are based on local constant or local linear kernel smoothing, two of which (i.e., LC2K and LL2K) take the possible blurring in the observed data into consideration while the other two do not. Also, a new quantitative metric for measuring the performance of a jump detector is proposed, which overcomes the major limitations of the Hausdorff distance and the distance metric proposed by Qiu (2002). A data-driven bandwidth selection procedure via bootstrap is suggested as well in the paper.



Figure 7: A test image with spatially varying blur involved.

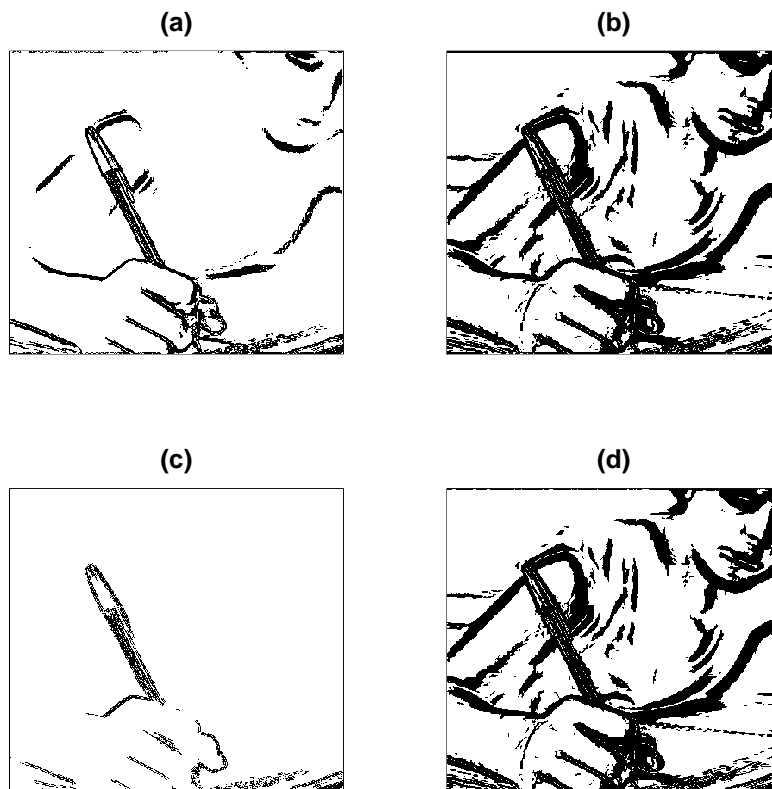


Figure 8: (a)-(d): Detected jumps from the image shown in Figure 7 by the jump detectors LCK, LC2K, LLK and LL2K, respectively.

Numerical examples presented in the previous section show that these jump detectors have their own strengths and limitations when handling various different situations. These results can be summarized as follows. (i) The jump detectors based on local constant kernel smoothing (i.e., LCK and LC2K) are more robust to noise and their detected jump location curves tend to be thinner. However, they cannot accommodate the slope effect of the true regression surface well and would detect many false jumps at places where the true regression surface is steep. (ii) The jump detectors based on local linear kernel smoothing (i.e., LLK and LL2K) can accommodate the slope effect of the true regression surface well; but they are more sensitive to noise. (iii) The jump detectors using 2 kernel functions (i.e., LC2K and LL2K) are more robust to blurring; however, their detected jumps tend to have a larger variability. Based on these results, we provide the following guidelines for the use of these jump detectors in practice. (i) The two jump detectors based on the local constant kernel smoothing (i.e., LCK and LC2K) are recommended in cases when the noise level in the observed image is relatively high. (ii) The two jump detectors based on the local linear kernel smoothing (i.e., LLK and LL2K) should be used in cases when the observed image contains some parts at which the image intensity surface is steep. (iii) The jump detectors using two kernel functions in their construction (i.e., LC2K and LL2K) are recommended in cases when the observed image has a quite severe blur involved. Much future research is needed to find an appropriate way to combine the major strengths of the four jump detectors and avoid their major limitations at the same time.

Supplementary Materials

supplemental.pdf: This pdf file contains proofs for Theorem 3.1.

ComputerCodesAndData.zip: This zip file contains Fortran source code of our proposed jump detection methods and the two test images used in the paper.

Acknowledgments: We thank the editor, an AE, and two referees for their constructive comments and suggestions which greatly improved the quality of the paper. This paper was supported in part by an NSF grant.

References

- Bates, R.H.T., and McDonnell, M.J. (1986), *Image Restoration and Reconstruction*, Clarendon Press: Oxford.
- Bowman, A.W., Pope, A., and Ismail, B. (2006), “Detecting discontinuities in nonparametric regression curves and surfaces,” *Statistics and Computing*, **16**, 377–390.
- Canny, J. (1986), “A Computational Approach to Edge Detection,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **8**, 679–698.
- Chu C.K., Siao, J.S., Wang, L.C., and Deng, W.S. (2012), “Estimation of 2D jump location curve and 3D jump location surface in nonparametric regression,” *Statistics and Computing*, **22**, 17–31.
- Clark, J.J. (1989), “Authenticating edges produced by zero-crossing algorithms,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **11**, 43–57.
- Ferger, D. (2004), “Boundary estimation based on set-indexed empirical processes,” *Journal of Nonparametric Statistics*, **16**, 245–260.
- Fleck, M.M. (1992), “Some defects in finite-difference edge finders,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **14**, 337–345.
- Garlipp, T., and Müller, C.H. (2006), “Detection of linear and circular shapes in image analysis,” *Computational Statistics & Data Analysis*, **51**, 1479–1490.
- Garlipp, T., and Müller, C.H. (2007), “Robust jump detection in regression surface,” *Sankhya (Series A)*, **69**, 55–86.
- Gijbels, I. and Goderniaux, A.C. (2004), “Bandwidth selection for changepoint estimation in nonparametric regression” *Technometrics*, **46**, 76–86
- Gonzalez, R.C., and Woods, R.E. (2002), *Digital Image Processing (2nd ed.)*, Prentice Hall: New York.

- Hall, P., and Qiu, P. (2007), “Nonparametric estimation of a point spread function in multivariate problems,” *Annals of Statistics*, **35**, 1512-1534.
- Hall, P., Qiu, P., and Rau, C. (2008), “Edge, corners and vertex estimation for images and regression surfaces,” *Scandinavian Journal of Statistics*, **35**, 1–17.
- Heath, M.D., Sarkar, S., Sanocki, T., and Bowyer, K.W. (1998), “Comparison of edge detectors - a methodology and initial study,” *Computer Vision and Image Understanding*, **69**, 38–54.
- Hillebrand, M., and Müller, C.H. (2007), “Outlier robust corner-preserving methods for reconstructing noisy images,” *The Annals of Statistics*, **35**, 132–165.
- Marr, D., and Hildreth, E. (1980), “Theory of edge detection,” *Proceedings of the Royal Society of London. Series B*, **207**, 187–217.
- Qiu, P. (1997), “Nonparametric estimation of the jump regression surface,” *Sankhya (Series A)*, **59**, 268-294.
- Qiu, P. (2002), “A nonparametric procedure to detect jumps in regression surfaces,” *Journal of Computational and Graphical Statistics*, **11**, 799–822.
- Qiu, P. (2004), “The local piecewisely linear kernel smoothing procedure for fitting jump regression surfaces,” *Technometrics*, **46**, 87–98.
- Qiu, P. (2005), *Image Processing and Jump Regression Analysis*, John Wiley & Sons: New York.
- Qiu, P. (2007), “Jump surface estimation, edge detection, and image restoration,” *Journal of the American Statistical Association*, **102**, 745–756
- Qiu, P. (2009), “Jump-preserving surface reconstruction from noisy data,” *Annals of the Institute of Statistical Mathematics*, **61**, 715–751.
- Qiu, P., and Bhandarkar, S.M. (1996), “An edge detection technique using local smoothing and statistical hypothesis testing,” *Pattern Recognition Letters*, **17**, 849–872.
- Qiu, P., and Yandell, B. (1997), “Jump detection in regression surfaces,” *Journal of Computational and Graphical Statistics*, **6**, 332-354.

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

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