Jump Surface Estimation, Edge Detection, And Image Restoration

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

Abstract

Surface estimation is important in many applications. When conventional smoothing procedures, such as the running averages, local polynomial kernel smoothing procedures, and smoothing spline procedures, etc., are used for estimating jump surfaces from noisy data, jumps would be blurred at the same time when noise is removed. In recent years, new smoothing methodologies have been proposed in the statistical literature for detecting jumps in surfaces and for estimating jump surfaces with jumps preserved. We provide a review of these methodologies in this paper. Because a monochrome image can be regarded as a jump surface of the image intensity function, with jumps at the outlines of objects, edge detection and image restoration problems in image processing are closely related to the jump surface estimation problem in statistics. In this paper, we also review major methodologies on edge detection and image restoration, and discuss about connections and differences between these methods and the related methods in the statistical literature.

Key Words: Adaptive smoothing; Bilateral filtering; Deblurring; Denoising; Edge detection; Image reconstruction; Image restoration; Jump detection; Jump location curves; Jump-preserving surface estimation; Nonparametric regression; Smoothing.

1 Introduction

Regression analysis provides a major statistical tool for building functional relationship between a response variable and an explanatory variable (or, an explanatory variable vector in multivariate cases). When this tool is first suggested by Galton, Pearson and some other pioneers more than one hundred years ago (cf., Stanton 2001), the true regression function is assumed *linear*. Linear regression models are later generalized to various *parametric* regression models, such as the Box-Cox regression model (Box and Cox 1964). Since late 1950's, parametric regression models are further generalized to *nonparametric* regression models (e.g., Nadaraya 1964, Parzen 1962, Rosenblatt 1956, 1969, Watson 1964). A typical two dimensional (2-D) nonparametric regression model can be written as follows:

$$Z_{ij} = f(x_i, y_j) + \varepsilon_{ij},\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2, \tag{1.1}$$

where x and y are two explanatory variables, $\{Z_{ij},\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2\}$ are observations of the response variable Z observed at design points $\{(x_i, y_j),\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2\}$ in design space $[0, 1] \times [0, 1]$, f is a bivariate regression function denoting a true regression surface, and $\{\varepsilon_{ij},\ i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2\}$ are independent and identically distributed (i.i.d.) random errors with mean 0 and unknown variance σ^2.

In conventional nonparametric regression analysis, f is often assumed to be a *continuous* function. In model (1.1), the continuity assumption implies that the true regression surface f can only change a small amount when the values of x and y change a small amount, which is reasonable in many applications. Several conventional smoothing procedures have been proposed in the literature for estimating a continuous regression function from noisy data, which include the local polynomial kernel smoothing procedures, smoothing splines, regression splines, wavelet transformation methods, and so forth. For nice overviews of these methods, see books, such as Fan and Gijbels (1996), Härdle (1990), Müller (1988), Qiu (2005), and Wahba (1990).

In some applications, however, the regression function f may have jumps in the design space, which often represent structural changes of the related process. For instance, the image intensity function of an image is discontinuous at the outlines of objects, and the equi-temperature surfaces in high sky or deep ocean are often discontinuous. It is easy to check that estimated regression functions by most conventional nonparametric regression procedures would not converge to the true regression function f at the jump positions, which implies that conventional nonparametric regression analysis can not handle the regression problem properly when f has jumps. To handle such a problem properly, some new statistical tools are necessary.

In recent years, some smoothing procedures have been suggested in the statistical literature for estimating jump regression functions from noisy data. For 1-D methodologies, see Gijbels and Goderniaux (2004), Gijbels *et al.* (2007), Hall and Titterington (1992), McDonald and Owen (1986), Müller (1992), Qiu (1991, 1994, 2003), Qiu *et al.* (1991), Qiu and Yandell (1998), Wang (1995), Wu and Chu (1993), and the references cited therein. In this paper, we focus on 2-D problems since they are directly related to image analysis, as discussed below. We provide a review of some recent methodologies on jump surface estimation.

An important application of jump surface estimation is image processing. A monochrome digital image, generated by a uniform sampling scheme in digitization of the spatial location, consists of pixels regularly spaced in rows and columns. It can be well described by the 2-D regression model (1.1). More specifically, in model (1.1), x_i denotes the i-th row of pixels, y_j denotes the j-th column, f is the image intensity function, $f(x_i, y_j)$ and Z_{ij} are the true and observed image intensity levels at the (i, j)-th pixel, and ε_{ij} denotes the random noise at that pixel. The image intensity function f has jumps at various places which are called *edges* in the image processing literature. Therefore, edge detection and edge-preserving image restoration in image processing are essentially the same problems as jump detection and jump-preserving surface estimation in regression analysis, although two different sets of terminologies are used in the two research areas. In this paper, we also provide a review of major edge detection and image restoration procedures.

Our discussion here is kept concise. We mainly address the main ideas of some fundamental methodologies in the related areas, their major strengths and limitations, connections and differences between jump surface estimation methods in the statistical literature and the related methods in the image processing literature, and some important open problems for future research. More detailed background introduction and detailed discussion about traditional methodologies on jump curve/surface estimation and image analysis can be found in Qiu (2005).

The remaining part of the article is organized as follows. Jump detection in 2-D regression surfaces is discussed in Section 2. Jump-preserving surface estimation is discussed in Section 3. In Sections 4, 5, and 6, we discuss edge detection, image denoising, and image deblurring problems, respectively. Some remarks conclude the article in Section 7.

2 Estimation of Jump Location Curves of 2-D Surfaces

Jumps in a regression surface often represent an important structure of the surface. Therefore, it is important to detect their positions accurately from observed data. In 2-D cases, jump positions usually form curves in the design space, which are called the *jump location curves (JLCs)* by Qiu (1997). For a formal definition of JLCs, see Qiu (1998). In this paper, for simplicity, we assume that design points are regularly spaced in the design space $[0,1] \times [0,1]$, as specified in model (1.1). Most existing procedures introduced in this section and the next section, however, can also work well in cases when design points are irregularly spaced with some homogeneity properties and when the design space is a general connected region in R^2. This section is organized in two parts. Section 2.1 introduces some early jump detection procedures requiring the assumption that the number of JLCs is known. Section 2.2 discusses some procedures for detecting arbitrary JLCs.

2.1 Jump detection when the number of JLCs is known

Early methodologies proposed in the statistical literature for estimating JLCs treat JLCs as curves in the design space and try to estimate them by some other curves in the same space. They usually assume that the number of JLCs is known and the JLCs satisfy certain smoothness conditions. For instance, when it is assumed that there is a single JLC which has the expression $y = \phi(x)$, for $x \in [0,1]$, Korostelev and Tsybakov (1993) suggest a minimax estimator of the JLC, by approximating the JLC with a piecewise polynomial function and by estimating the polynomial coefficients with maximum likelihood estimation (MLE). In such cases, Rudemo and Stryhn (1994) suggest estimating the function ϕ by a step function with its coefficients estimated by MLE. Both papers make some generalizations to cases with multiple JLCs that all have mathematical expressions. In the case when there is a single JLC and the JLC is a "smooth, closed and simple" curve, O'Sullivan and Qian (1994) suggest estimating the JLC by the minimizer of a contrast statistic, defined by the difference between observation averages inside and outside a candidate JLC, searched in a sufficiently rich class of candidate JLCs, including the true JLC as a member.

To have a rough idea how such methods work, next, we introduce two methods by Müller and Song (1994a) and Qiu (1997) in more details. Both methods assume that there is a single JLC. Müller and Song's jump detection criterion $\Delta(\theta, x, y)$ is defined by a difference of two averages of the observations located in two one-sided square-shaped neighborhoods of a given point (x, y) along a direction θ. The two neighborhoods are required to be a certain distance apart. The absolute jump magnitudes along the true JLC Γ are assumed to be bounded above 0. Then, their estimator $\widehat{\Gamma}$ of Γ is defined by the maximizer of

$$\sup_{\Gamma' \in \Xi} \left[\inf_{(x,y) \in \Gamma'} \left(\sup_{\theta \in [-\pi/2, \pi/2]} |\Delta(\theta, x, y)| \right) \right], \tag{2.1}$$

where Ξ denotes a sufficiently rich class of candidate JLCs, including Γ as a member. When the JLC has the expression $y = \phi(x)$, Qiu suggests a so-called *rotational difference kernel estimator (RDKE)* of the JLC, using two *rotational kernel functions* $K_j(\theta, x, y; h_n, p_n)$, for $j = 1, 2$, obtained by rotating the supports $[-h_n/2, h_n/2] \times [0, p_n]$ and $[-h_n/2, h_n/2] \times [-p_n, 0)$ of two one-sided kernel functions $K_j^*(x/h_n, y/p_n)$, for $j = 1, 2$, an angle θ counterclockwise about the origin, where h_n and p_n are two bandwidth parameters and $n = n_1 n_2$. Then, the jump detection criterion is defined by

$$\begin{aligned} M_{RDKE}(\theta, x, y) &= \frac{1}{n_1 n_2 h_n p_n} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} Z_{ij} \left[K_2\left(\theta, x_i - x, y_j - y; h_n, p_n\right) - \right. \\ & \quad \left. K_1\left(\theta, x_i - x, y_j - y; h_n, p_n\right) \right], \end{aligned} \tag{2.2}$$

where $(x, y) \in [b_n, 1 - b_n] \times [b_n, 1 - b_n]$ and $b_n = \sqrt{h_n^2/4 + p_n^2}$. The RDKE of $\phi(x)$ is defined by

$$\theta^*(x, y) = \arg \max_{\theta \in [-\pi/2, \pi/2]} \left| M_{RDKE}\left(\theta, x, y\right) \right|; \quad \widehat{\phi}(x) = \arg \max_{b_n \leq y \leq 1 - b_n} \left| M_{RDKE}\left(\theta^*(x, y), x, y\right) \right|.$$

It should be pointed out that procedures (2.1) and (2.2) could be generalized in several different ways. For instance, when multiple JLCs are present, in order to use Müller and Song's procedure (2.1), the design space can be partitioned into a sufficiently large number of disjoint regions so that Müller and Song's procedure can be used in each region. To lessen its computational burden, instead of searching all possible directions at a given point for a gradient direction (cf., expression (2.1)), we could search only the x- and y-axis directions. See Müller and Song (1994b) for related discussions. For procedure (2.2), Garlipp and Müller (2004) correct a mistake in Qiu (1997), and propose a robust version using M-kernel estimation. Such robust jump detection procedures are useful when there are outliers or "salt-and-pepper" noise in observed data. Garlipp and Müller (2006) further extend that robust jump detection procedure for detecting linearly or circularly shaped objects.

Other methods for estimating JLCs when the number of JLCs is assumed known include several tracking algorithms proposed by Peter Hall and co-authors (e.g., Hall, Peng and Rau 2001, Hall, Qiu and Rau 2007, Hall and Rau 2000, 2002), methods based on wavelet transformations (e.g., Wang 1998), methods by smoothing splines (e.g., Shiau 1985), and so forth.

2.2 Detection of arbitrary JLCs

A major reason why some methods mentioned in the previous subsection require the number of JLCs to be known and some other quite restrictive assumptions, we think, is due to the philosophy that JLCs are curves and they should be estimated by curves. We know that curves have the *global* nature that any two points on a curve are connected by some other points on the same curve. Because of that, conditions imposed on the JLCs by these methods are also global, in the sense that they should be satisfied at all points on the JLCs. An alternative, but more flexible description of the JLCs is that points on the JLCs constitute a *pointset* in the design space, which can be estimated by another pointset in the same design space. Since a pointset needs not form curves, it is more flexible to describe and estimate. Using this concept, several procedures have been proposed in the literature for estimating arbitrary JLCs, some of which are described below.

Qiu and Yandell (1997) suggest estimating arbitrary JLCs based on local linear estimation. At any design point (x_i, y_j), for $\ell + 1 \leq i \leq n_1 - \ell$ and $\ell + 1 \leq j \leq n_2 - \ell$, we consider a square-shaped neighborhood $N(x_i, y_j)$ with width $k = 2\ell + 1$, where $0 < \ell \ll \min(n_1, n_2)$ is an integer. A least squares (LS) plane is then fitted in this neighborhood, and the fitted LS plane is denoted by

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

Then, the gradient direction of the fitted plane is $\vec{v}_{ij} = (\widehat{\beta}_1^{(i,j)}, \widehat{\beta}_2^{(i,j)})$. On two sides of (x_i, y_j) along this direction, let (x_{N1}, y_{N1}) and (x_{N2}, y_{N2}) be two design points whose neighborhoods $N(x_{N1}, y_{N1})$ and $N(x_{N2}, y_{N2})$ are just next to the neighborhood $N(x_i, y_j)$. Then, a jump detection criterion is defined by

$$\delta_{ij} = \min\left\{\|\vec{v}_{ij} - \vec{v}_{N1}\|, \|\vec{v}_{ij} - \vec{v}_{N2}\|\right\}, \tag{2.3}$$

where $\vec{v}_{N1}$ and $\vec{v}_{N2}$ are gradient vectors of the fitted LS planes in $N(x_{N1}, y_{N1})$ and $N(x_{N2}, y_{N2})$, respectively, and $\|\cdot\|$ is the Euclidean norm. The set of design points $\{(x_i, y_j) : \delta_{ij} > b,\ i = (3k+1)/2, \ldots, n_1 - (3k-1)/2, j = (3k+1)/2, \ldots, n_2 - (3k-1)/2\}$ is then used as an estimator of the JLCs, where b is a threshold parameter, properly chosen by some hypothesis testing arguments. Due to the nature of local smoothing and thresholding, on which the above procedure is based, there may be two kinds of deceptive jumps detected. They are either close to the true JLCs or scattered in the whole design space. To delete them, Qiu and Yandell also suggest two modification procedures.

Qiu (2002) proposes an alternative estimator of the JLCs, which tries to simplify the computation involved in the RDKE procedure (2.2). Notice that the RDKE procedure detects a possible jump at a given point by searching all possible directions, which is computationally expensive. To overcome this limitation, Qiu (2002) suggests searching only the x- and y-directions at any point (x, y) by the criterion

$$M_n(x, y) = \max\left\{|M_{RDKE}(0, x, y)|, |M_{RDKE}(\pi/2, x, y)|\right\},$$

where M_{RDKE} is defined in (2.2). Then, the pointset

$$\widehat{D}_n = \{(x_i, y_j) : M_n(x_i, y_j) > u_n\} \tag{2.4}$$

is used as an estimator of the pointset of the true JLCs, denoted as D, where u_n is a threshold parameter. Qiu proves that, as long as the two bandwidths h_n and p_n (cf., equation (2.2)) are chosen such that $\lim_{\min(n_1,n_2)\to\infty} h_n/p_n = 0$ and other regularity conditions are satisfied, $\widehat{D}_n$ is a strong consistent estimator of D, in Hausdorff distance defined by

$$d_H(\widehat{D}_n, D) = \max\left\{\sup_{\mathbf{s}_1\in\widehat{D}_n}\inf_{\mathbf{s}_2\in D}\|\mathbf{s}_1 - \mathbf{s}_2\|, \sup_{\mathbf{s}_1\in D}\inf_{\mathbf{s}_2\in\widehat{D}_n}\|\mathbf{s}_1 - \mathbf{s}_2\|\right\}.$$

Regarding performance measurement of a pointset estimator, Qiu (2002) points out two limitations of the Hausdorff distance for that purpose. One is that it is difficult to compute, and the second one is that it is sensitive to individual points in the related pointsets, due to the supremums/infimums involved in its definition. To overcome these limitations, Qiu suggests an alternative performance measure $d^*(\widehat{D}_n, D)$, which is a weighted average of the proportion of the detected continuity points among all continuity design points and the proportion of the true jump points missed by the jump detection procedure. However, this alternative measure has its own limitations. For instance, it depends on the two proportions mentioned above; but, it does not depend on relative distances between points in the two pointsets. In the extreme case when $\widehat{D}_n$ and D are disjoint, $d^*(\widehat{D}_n, D)$ would be a constant, no matter how far away $\widehat{D}_n$ is from D, which is obviously unreasonable. Due to the importance of a good performance measure in comparing different jump detectors, much more future research is needed on this topic.

The idea to estimate JLCs by a pointset allows us to handle arbitrary JLCs. As long as the points in the pointset are close enough to the true JLCs and the points are dense enough as well, the visual effect of the estimator should be reasonably good. Otherwise, the pointset estimator would not be acceptable. Furthermore, in certain applications, it is better to estimate JLCs by curves instead of pointsets. For instance, in bioinformatics, before generating gene expression data from spotted microarray images, spot boundaries separating foregrounds from backgrounds should be estimated first (cf., e.g., Qiu and Sun 2007, Young *et al.* 2002). If spot boundaries are estimated by pointsets, then foreground and background pixels would not be properly defined even after jump detection; consequently, gene expression data can not be generated. For these reasons, it might be an interesting research problem to replace the pointset estimator by some curves in certain cases, after the pointset estimator is obtained. In the context of image segmentation for microarray images, Qiu and Sun (2006) suggest a post-smoothing algorithm for this purpose.

We notice that almost all existing jump detection procedures in the statistical literature detect jumps based on estimation of the first-order derivatives of the true surface f in one way or another. As discussed in Section 4 below, second-order derivatives of f also include useful information for jump detection. It might be an interesting problem to make use of this information properly in jump detection, together with the useful information included in the first-order derivatives. Recently, Sun and Qiu (2007) propose such a jump detection procedure using both the first-order and second-order derivatives; but, more research is needed in this direction. A related problem is to detect jumps in derivatives of f, which has not been properly addressed yet in the statistical literature. For instance, jumps in the first-order derivatives correspond to sharp angles in the surface, which are often called roof edges in image analysis (cf., some related discussion in Section 4). Detection of such jumps, or detection of such jumps together with jumps in f, should be important in certain applications.

3 Jump-Preserving Surface Estimation

In some applications, our ultimate goal is to estimate the underlying regression surface from noisy data. When the surface has jumps, conventional 2-D smoothing procedures are not appropriate to use, since they would blur jumps when removing noise, as discussed in Section 1. Therefore, new smoothing procedures are necessary for jump-preserving surface estimation.

There are two potential approaches to estimation of surfaces with jumps preserved. By the first approach, jumps are first detected by a jump detector, as described in Section 2, and then the surface is estimated properly after accommodating the detected jumps. This type of jump surface estimation methods is referred to as the *direct* methods in this paper. By the second approach, jumps are not detected explicitly; smoothing procedures adapt to the underlying jump structure automatically. Such methods are referred to as the *indirect* methods. In a specific application, if specification of jump locations is important, besides surface estimation, then the direct methods are appropriate to use. The microarray image analysis problem mentioned in Section 2 belongs to this scenario. If our ultimate goal is surface estimation, it is desirable to preserve jumps in the estimation process because they are important structures of the true surface, but it is not our major research interest in knowing the jump locations, then the indirect methods might be more convenient to use. Many image denoising applications can be classified into this category. See Section 5 for related discussion.

In the simple case when there is a single JLC with expression $y = \phi(x)$, the jump surface can be estimated easily by one of the following two direct methods. (1) The JLC is first estimated with a curve, as discussed in Section 2.1, and then the surface is estimated as usual in design subregions separated by the estimated JLC. (2) The JLC and the corresponding jump magnitude function $C(x)$ are first estimated by univariate functions $\widehat{\phi}(x)$ and $\widehat{C}(x)$, then the modified data $\{Z_{ij} - \widehat{C}(x_i)I(y_j > \widehat{\phi}(x)),\ i = 1, 2, \ldots, n_1; j = 1, 2, \ldots, n_2\}$ are computed, and finally the continuity part of the surface is estimated as usual from the modified data, where $I(\cdot)$ is the indicator function and the data $\{(x_i, y_j, Z_{ij}),\ i = 1, 2, \ldots, n_1; j = 1, 2, \ldots, n_2\}$ are assumed to be from model (1.1). Obviously, such direct methods can also be applied to some more general cases, such as the one when multiple JLCs exist, each JLC can be expressed by a univariate function of x, and all JLCs do not cross each other.

In a general case with arbitrary JLCs, the JLCs are often estimated by a pointset which may not form curves in the design space, as discussed in Section 2.2. In such cases, neither one of the two direct methods mentioned above would work. To overcome this difficulty, Qiu (1998) suggests a three-stage direct method, briefly described below. In the first stage, jump candidate points are detected using a jump detector, such as (2.3). Then, in the second stage, if there are enough detected jump points in a neighborhood of a given design point, a local principal component (PC) line is fitted through them, which provides a first-order approximation to a possible JLC in the neighborhood. In the third stage, observations on the same side of the PC line as the given point are combined using a weighted average procedure to estimate the surface at the given point. By this method, if there are no jump candidate points in the neighborhood, then all observations in the neighborhood are actually used in surface estimation. On the other hand, if a JLC exists in the neighborhood, then only those observations on one specific side of the PC line are used. Thus, blurring is avoided around the jump locations.

In the statistical literature, there are several indirect methods proposed recently. The so-called sigma filter and M-smoother, discussed by Chu *et al.* (1998), are both based on robust estimation. The surface estimator of the sigma filter is defined by

$$\widehat{f}_S(x_i, y_j) = \frac{\sum_{s=1}^{n_1}\sum_{t=1}^{n_2} Z_{st}K((x_s - x_i)/h_n, (y_t - y_j)/h_n)L((Z_{st} - Z_{ij})/g_n)}{\sum_{s=1}^{n_1}\sum_{t=1}^{n_2} K((x_s - x_i)/h_n, (y_t - y_j)/h_n)L((Z_{st} - Z_{ij})/g_n)}, \tag{3.1}$$

where K and L are two density kernel functions, and h_n and g_n are two bandwidths. Clearly, $\widehat{f}_S(x_i, y_j)$ is a weighted average of all observations; the weights are determined not only by the distance from individual design points (x_s, y_t) to the given design point (x_i, y_j), but also by the difference between individual observations Z_{st} and the observation Z_{ij} at (x_i, y_j). If (x_s, y_t) and (x_i, y_j) are on two different sides of a JLC, then the absolute difference between Z_{st} and Z_{ij} would be large. Consequently, Z_{st} would receive small weights in computing $\widehat{f}_S(x_i, y_j)$. Thus, jumps are preserved to some degree by (3.1). The surface estimator of the M-smoother is defined by the solution of

$$\min_{a \in R} \sum_{s=1}^{n_1}\sum_{t=1}^{n_2} \rho(Z_{st} - a)K\left(\frac{x_s - x_i}{h_n}, \frac{y_t - y_j}{h_n}\right),$$

where $\rho(z)$ is a robust loss function chosen to grow slowly when z tends to $\pm\infty$.

The basic idea of both sigma filter and M-smoother is to assign small weights to observations located on the other side of a JLC around a given design point. However, those observations still receive *some* weights in the weighted average; therefore, certain blurring is inevitable by these methods, although the degree of blurring has been greatly reduced. This issue is especially important when the noise level in the observed data is high, or, when we try to preserve small jumps. Polzehl and Spokoiny (2000) suggest an iterative version of the sigma filter, by which the above mentioned problem might be alleviated to a certain degree after some iterations. However, so far, we do not know much about the theoretical properties of that iterative procedure yet, which requires much future research.

Recently, Qiu (2004) suggests an indirect method based on local piecewisely linear kernel smoothing. At a given point, a square-shaped neighborhood is considered. Then, four surface estimators are obtained by local linear kernel smoothing in the four quadrants of the neighborhood, respectively. If the range of the four estimators is larger than a threshold value, then the final surface estimator is defined by the one with the smallest weighted residual sum of squares of the related fitted local plane. Otherwise, the final surface estimator is defined by the simple average of the four estimators. This procedure preserves jumps well; but, its surface estimator is a bit noisy in continuity regions of the surface. Several improvements have been proposed recently. For instance, Gijbels *et al.* (2006) suggest fitting a plane in a circular neighborhood $N(x,y)$ of a given point (x,y) by local linear kernel smoothing, obtaining a conventional surface estimator $\widehat{a}_c(x,y)$ and a gradient estimator $G(x,y)$. Then, the neighborhood $N(x,y)$ is divided into two halves by a line passing through (x,y) and perpendicular to the gradient direction $G(x,y)$. Two one-sided surface estimators $\widehat{a}_1(x,y)$ and $\widehat{a}_2(x,y)$ are obtained, respectively, in the two halves of $N(x,y)$, by local linear kernel smoothing. The final surface estimator is defined by

$$\widehat{f}(x,y) = \begin{cases} \widehat{a}_c(x,y), & \text{if } \mathrm{dif}(x,y) \leq u \\ \widehat{a}_1(x,y), & \text{if } \mathrm{dif}(x,y) > u \text{ and } \mathrm{WRMS}_1(x,y) < \mathrm{WRMS}_2(x,y) \\ \widehat{a}_2(x,y), & \text{if } \mathrm{dif}(x,y) > u \text{ and } \mathrm{WRMS}_1(x,y) > \mathrm{WRMS}_2(x,y) \\ \frac{\widehat{a}_1(x,y)+\widehat{a}_2(x,y)}{2}, & \text{if } \mathrm{dif}(x,y) > u \text{ and } \mathrm{WRMS}_1(x,y) = \mathrm{WRMS}_2(x,y), \end{cases} \tag{3.2}$$

where $\mathrm{dif}(x,y) = \max\{\mathrm{WRMS}_c(x,y) - \mathrm{WRMS}_1(x,y), \mathrm{WRMS}_c(x,y) - \mathrm{WRMS}_2(x,y)\}$, $\mathrm{WRMS}_c(x,y)$, $\mathrm{WRMS}_1(x,y)$, and $\mathrm{WRMS}_2(x,y)$ denote the weighted residual mean squares of the corresponding fitted local planes, and u is a threshold parameter. An alternative, two-stage, surface estimator is suggested by Qiu (2006). In the first stage, the fitted surface is defined by the one of $\widehat{a}_c(x,y), \widehat{a}_1(x,y)$, and $\widehat{a}_2(x,y)$ with the smallest WRMS value. In the second stage, estimated surface values at the design points obtained in the first stage are used as new data, and the above procedure is applied to this data in the same way except that one of the three estimators is selected based on their estimated variances in the second stage. It is shown that the final surface estimator preserves jumps well and removes noise efficiently.

The surface estimator (3.2) and the alternative estimator by Qiu (2006) still have much room for improvement. For instance, estimator (3.2) uses a constant threshold value u in the entire design space, which may not be reasonable in cases when the noise level changes with location. Both procedures divide the circular neighborhood of a given point into two halves for preserving jumps in the entire design space. To preserve corners of JLCs, estimator (3.2) is generalized in Gijbels *et al.* (2006) by using two opposite sectors of constant size in the circular neighborhood when constructing the two one-sided estimators. In my opinion, an ideal surface estimation procedure should have the ability to adapt to not only the presence of jumps but also the shape of the underlying JLCs, which has not been achieved yet by most existing procedures.

Another issue that has not been well addressed in the literature is about preservation of other important features of the regression surface, including peaks and valleys, in surface estimation. So far, most existing procedures discuss preservation of jumps in f only. In some applications, especially those with reasonably large sample sizes, such as image analysis, preservation of peaks, valleys, and other important features should be important and also possible to achieve.

4 Edge Detection in Image Processing

In the computer science literature, many different image processing techniques have been proposed for various purposes (cf., e.g., Gonzalez and Woods 1992, Rosenfeld and Kak 1982). In this and the next two sections, we focus on three types of image processing techniques: those for edge detection, those for image denoising, and those for image deblurring, since they are widely used in applications and closely related to the jump surface estimation methods discussed in the previous two sections.

In the literature, there are several different types of *edges* defined (cf., Qiu 2005, Chapter 6). *Step edges* refer to places in an image at which the image intensity function has jumps, and *roof edges* refer to places at which the first-order derivatives of the image intensity function have jumps. Other types of edges include ramp edges, spike edges, line edges, texture edges, and so forth. Most edge detection techniques in the literature are for detecting step edges, because they often represent outlines of objects or some sudden structural changes of the related process. For this reason, edges mentioned in this and the next two sections are referred to step edges if there is no further specification.

Most existing edge detectors are based on estimation of the first-order and second-order derivatives of the image intensity function, because they include useful information about edges, as explained below. Figure 4.1(a) displays a 1-D profile of a step edge which is slightly blurred during image acquisition. Its first- and second-order derivatives are displayed in Figure 4.1(b)–(c). It can be seen that (1) the first-order derivative peaks at the step edge, (2) the second-order derivative equals zero at the step edge, and (3) the second-order derivative changes its sign on two sides of the edge. In the literature, the second and third properties are called *zero-crossing* properties of the second-order derivative. In the 2-D setup, the directional first- and second-order derivatives in the direction perpendicular to a given edge segment have similar properties.

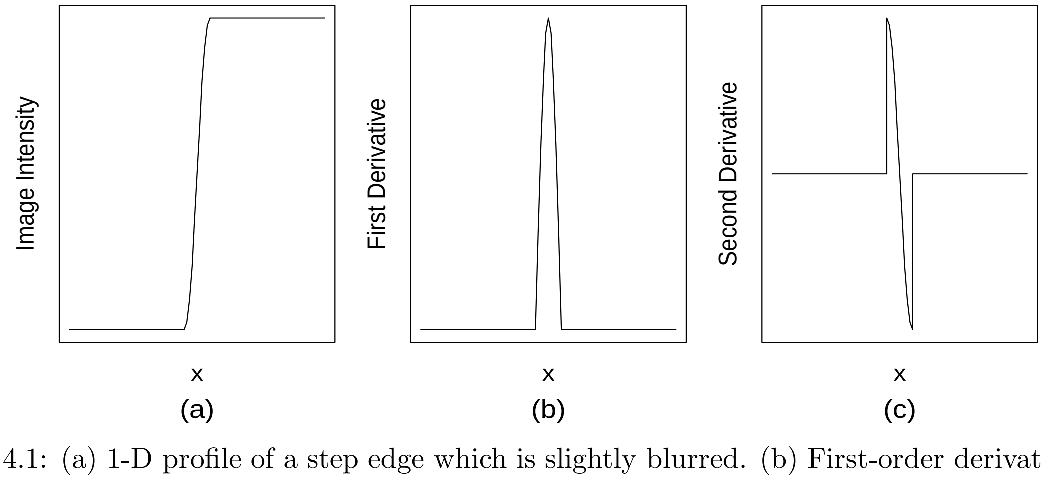

Figure 4.1: (a) 1-D profile of a step edge which is slightly blurred. (b) First-order derivative of the 1-D profile. (c) Second-order derivative of the 1-D profile.

Edge detectors based on first-order derivatives usually work as follows. Let $f(x,y)$ be the image intensity function. The magnitude of its gradient vector

$$M_f(x,y) = \sqrt{[f'_x(x,y)]^2 + [f'_y(x,y)]^2}, \tag{4.1}$$

should be large if the pixel (x,y) is on an edge segment. So, $M_f(x,y)$ can be used for edge detection, after $f_x'(x,y)$ and $f_y'(x,y)$ are estimated properly. Many masks have been proposed in the literature for estimating f_x' and f_y' using directional differences, which include Sobel masks, Roberts operators, Prewitt masks, Frei-Chen masks, truncated pyramid operators, derivative of Gaussian (DoG) operators, and so forth (cf., e.g., Gonzalez and Woods 1992, Chapter 4). After $M_f(x,y)$ is estimated, we still need to choose a threshold value for identifying detected edge pixels. In applications, we usually try a sequence of threshold values and then choose the one giving results with the best visual impression.

Obviously, edge detectors mentioned above are similar in nature to the jump detectors discussed in Section 2, because all of them are based on estimation of first-order derivatives of f. But, there are several important differences between the two groups of methods. First, edge detectors usually use masks of fixed sizes. Commonly used sizes are 3×3, 5×5, and so forth. With such small sizes, edge detectors are easy to compute and use; but their ability to remove noise is limited. As a comparison, window widths of most jump detectors in the statistical literature are chosen by data-driven procedures, such as the cross-validation and bootstrap procedures. Second, most masks used in the image processing literature are square-shaped. However, as pointed out by Qiu (2002), to successfully detect edges with different curvatures, the x-direction and y-direction masks should be narrow along the x- and y-axis directions, respectively (cf., related discussion about the jump detector (2.4) in Section 2.2). Third, most edge detectors have made use of the spatial regularity of pixel locations. Therefore, they can not be applied to applications with irregular design points. As a comparison, most jump detectors in statistics do not have this limitation. Other differences between the two groups of methods include weight assignments, threshold selection, and so forth.

We now discuss edge detection based on second-order derivatives. A commonly used operator for this purpose is the following Laplacian operator:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}.$$

It has the zero-crossing properties around a given edge pixel (x_0, y_0) in the following sense: $\nabla^2 f(x,y)$ would be positive on one side of the edge segment, negative on the other side, and zero at (x_0, y_0) or at some place(s) between (x_0, y_0) and its neighboring pixels. Compared to edge detectors based on the first-order derivatives of f, the Laplacian operator has the advantage in localizing the detected edges well, i.e., its detected edges are usually thinner than those detected by edge detectors based on the first-order derivatives. However, it is quite sensitive to noise. To balance localization of detected edges and denoising, Marr and Hildreth (1980) propose the *Laplacian of Gaussian (LoG)* edge detector, by which the image is pre-smoothed by a weighted averaging procedure with the weights determined by a 2-D Gaussian density $G(x,y;s)$ where s is a scale parameter, and then edges are detected by applying the Laplacian operator to the pre-smoothed image. For related discussions, see Haralick (1984), Nalwa and Binford (1986), Qiu and Bhandarkar (1996), and the references cited therein. It should be pointed out that, although LoG is less sensitive to noise, compared to the Laplacian operator, it has its own limitations. One major limitation is caused by the pre-smoothing step. At the same time when noise is removed in that step, edges especially those with small sizes would be blurred. Consequently, localization of the detected edges would be sacrificed to a certain degree. To overcome this limitation, in my opinion, the Gaussian pre-smoother should be replaced by an edge-preserving smoother. For this purpose, some jump-preserving surface estimation methods discussed in Section 3 might be helpful.

Canny (1986) suggests a benchmark edge detector based on the following three criteria for measuring edge detection performance: (i) there should be a low probability of missing real edge pixels, and a low probability of detecting false edge pixels, (ii) detected edge pixels should have good localization, and (iii) there should not be multiple responses to a single true edge pixel. After describing these criteria with mathematical expressions, Canny derives an optimal edge detector for detecting step edges, using the so-called *variational approach.* It is shown that this detector can be well approximated by the DoG and LoG operators mentioned above. In the literature, there are several modifications and generalizations of Canny's edge detection criteria. For instance, Petrou and Kittler (1991) adapt Canny's criteria for detecting roof edges and ramp edges. Demigny and Kamlé (1997) propose a discrete version for Canny's criteria. More recent discussion about performance assessment of edge detectors can be found in Heath *et al.* (1997, 1998), Shin *et al.* (2001), Yitzhaky and Peli (2003), and the references cited there.

Tan *et al.* (1992) formulate the edge detection problem as a problem of cost minimization. Their edge estimator is defined to be the edge configuration that minimizes a cost function, defined as a weighted average of the following five cost factors: edge curvature, dissimilarity of the regions separated by the edges, number of edge pixels, fragmentation of the edges, and edge thickness. After these five cost factors are properly defined, Tan *el al.* suggest searching for the optimal edge estimator among all possible candidates by a simulated annealing algorithm. An alternative, genetic algorithm-based optimization procedure for finding the optimal edge estimator is suggested by Bhandarkar *et al.* (1994).

Besides the edge detectors described above, there are many others in the literature. For instance, as a by-product, Markov random field (MRF) modeling approaches and diffusion approches, which are mainly for image restoration and will be described in more detail in the next section, can also generate detected edge images (cf., Hansen and Elliot 1982, Perona and Malik 1990). Shiau (1985, 1987) and Shiau *et al.* (1986) suggest using the *partial smoothing spline* method for estimating certain discontinuous curves and surfaces. This general method has been applied to several image analysis problems, including estimation of object boundaries and detection of boundary corners (cf., Chen and Chin 1993). We believe that it has a great potential to be used for edge detection.

At the end of this section, we would like to mention that several other image analysis problems, including image segmentation, image enhancement, object boundary detection, pattern recognition, shape analysis, and so forth, are closely related to edge detection. Methodologies used for solving these problems might also be helpful for edge detection. For instance, *active contour models (ACMs)*, or *snakes*, and geodesic ACMs are used mainly for shape analysis (cf., e.g., Caselles *et al.* 1997, Chan and Luminita 2001, Kass *et al.* 1988, Malladi *et al.* 1995). The basic idea of these methods is to evolve a curve, subject to constraints from a given image, in order to detect object boundaries in that image. As an example, by the snake model, the boundary of an object in the image I is estimated by the solution of

$$\min_C \left\{ \alpha \int_0^1 |C'(s)|^2 \, ds + \beta \int_0^1 |C''(s)| \, ds - \lambda \int_0^1 | \bigtriangledown I(C(s))|^2 \, ds \right\},$$

where $C(s) : [0,1] \to R^2$ is a candidate, parameterized, boundary curve, $\bigtriangledown I(C(s))$ is the image gradient assigned at the boundary curve, and α, β and λ are positive parameters. In the above expression, the first two terms control the smoothness of the candidate boundary curve and the third term attracts the candidate boundary curve toward the object in the image. In the literature, such a minimization problem is often handled using the *level set method* by solving a partial differential equation (cf., e.g., Caselles *et al.* 1993, Osher and Sethian 1988).

5 Image Denoising

Observed images are often degraded versions of their true images. If *pointwise* noise and *spatial* blur are the degradations of our major concern, then the relationship between an observed image Z and its true image f can be described by the following model:

$$Z(x,y) = (h \otimes f)(x,y) + \varepsilon(x,y), \text{ for } (x,y) \in \Omega, \tag{5.1}$$

where $\varepsilon(x,y)$ denotes noise at (x,y), Ω is the design space of the image, and $h \otimes f$ is a spatially degraded version of f, defined by the convolution between a *point spread function (psf)* h and the true image f. *Image restoration* techniques try to recover the true image f from the observed one Z. Since edges are important structures of the true image, they should be preserved in the process of image restoration.

Generally speaking, image restoration is challenging, due mainly to the fact that spatial blur and pointwise noise often mix up and it is difficult to remove them simultaneously. For ease of presentation, we roughly classify all image restoration methods into two categories: those mainly for *denoising* and those mainly for *deblurring*. Denoising methods, discussed in this section, are often used in applications in which there is little blurring in Z. Some denoising methods can also be used in applications when there is substantial blurring in Z but the blurring mechanism is known. When blurring is our major concern, deblurring methods should be used, which are discussed in Section 6.

Image restoration by MRF modeling is an active research area in the image processing literature. Geman and Geman (1984) provide a general framework for this approach. Suppose that a given image has n_1 rows and n_1 columns of pixels, $\mathbf{X} = \{X_{ij}; i,j = 1,2,\ldots,n_1\}$ is the matrix of observed image intensities at the pixel sites $\mathcal{S} = \{(i,j); i,j = 1,2,\ldots,n_1\}$, and $\mathbf{F} = \{F_{ij}, i,j = 1,2,\ldots,n_1\}$ is the matrix of true image intensities. Between two neighboring pixels in either horizontal or vertical direction, it is assumed that there is an unobservable binary edge element, with 1 denoting an existing edge between the two pixels and 0 denoting no edge. The matrix of all such edge elements is called a *line process*, denoted by $\mathbf{L}$. Geman and Geman assume that $\mathbf{Y} = (\mathbf{F}, \mathbf{L})$ is a MRF, or equivalently, its *prior* distribution is the following Gibbs distribution:

$$P(\mathbf{F} = \mathbf{f}, \mathbf{L} = \mathbf{l}) = \frac{1}{A} \exp\left(-\frac{U(\mathbf{f}, \mathbf{l})}{T}\right),$$

where $U(\mathbf{f}, \mathbf{l})$ is an energy function, A is a normalizing constant, and T is a *temperature* parameter. Under the assumption that pointwise noise is i.i.d. with normal distribution, it is proved that the *posterior* distribution of $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}$ is also a Gibbs distribution with energy function $U^P(\mathbf{f}, \mathbf{l})$, where the superscript "P" denotes "posterior". The restored image is defined by one of the maximizers of the posterior distribution, or, equivalently, one minimizer of the energy function $U^P(\mathbf{f}, \mathbf{l})$. This estimator is called the maximum *a posteriori* (MAP) estimator. By this approach, edges are detected and preserved by using the line process $\mathbf{L}$. However, to search for the MAP estimator directly from all image configurations is almost impossible numerically, even for a moderate-sized gray-scale image. To overcome the computational difficulty, Geman and Geman suggest using the *Gibbs sampler* scheme and an *annealing* algorithm in obtaining the MAP estimator.

To further simplify the computation involved in the MAP procedure, Besag (1986) suggests the *iterated conditional modes (ICM)* algorithm for image restoration. The ICM algorithm works iteratively. Its updated estimator of F_{ij} in each iteration is defined by the maximizer of the *local* conditional probability $P(F_{ij}|\mathbf{X}, \widehat{F}_s, s \in \mathcal{S}\backslash\{(i,j)\})$, instead of the *global* posterior distribution as in the MAP procedure, where $\widehat{F}_s$ is obtained from the previous iteration. In the literature, there are some other generalizations and modifications of the MAP procedure. See, for instance, Fessler *et al.* (2000), Godtliebsen and Sebastiani (1994), and Marroquin *et al.* (2001). Li (1995) provides a nice overview on this topic.

Usually, MRF methods work well in cases when there is blurring involved in the observed image and the blurring mechanism described by the psf h is known. Because of their iterative nature, MRF methods are often relatively difficult to compute, although a great amount of effort has been made in the literature for simplifying their computation. Due to the same reason, it is also difficult for us to study their theoretical properties. Many numerical studies in the literature show that they work well in various applications, though the MRF assumption and some other model assumptions may not be valid in certain applications. See the references cited above for related discussion.

The image restoration problem can also be formulated as a *regularization* problem, in which the restored image is defined by the minimizer of the following energy function in an appropriate function space:

$$U(g, d) + U(g),$$

where the first term $U(g, d)$ measures the *fidelity* of a candidate estimator g of the true image intensity function f to the observed image intensity function d (i.e., the *data*), and $U(g)$ is a *smoothness* measure of g, which is called a *regularizer*. To preserve edges, a line process is often used in $U(g)$. Poggio *et al.* (1985) give a nice review on some early regularization procedures. More recent methods can be found in Blake and Zisserman (1987), Green (1990), Lange (1990), Bouman and Sauer (1993), Stevenson *et al.* (1994), Li (1998), and the references cited there.

Local *median filtering* is a popular pre-smoothing tool in image processing because it has a certain ability of preserving edges while removing noise (Gallagher and Wise 1981, Huang 1981). By conventional local median filtering, the restored image intensity at a given pixel equals the sample median of the observed image intensities in a neighborhood. Usually, square-shaped neighborhoods are used in applications. Sometimes, cross-shaped, X-shaped, and other types of neighborhoods are also used for better preserving *line edges*, angles, and other image details. There are some generalizations of the conventional local median filtering. Among them, *weighted median filtering* is found useful in several applications (e.g., Arce 1991, Haavisto *et al.* 1991). From statistical viewpoint, we know that median filters are sensitive to noise. Therefore, it is difficult for them to preserve small edges in the presence of noise. As pointed out earlier, for pre-smoothing purposes, some jump-preserving surface estimation procedures discussed in Section 3 might be more reasonable to use.

Saint-Marc *et al.* (1991) propose another local smoothing filter, called *adaptive smoothing filter*, based on the idea that edge structure of the image should be taken into account when the image is restored. That filter works iteratively as follows. Let $Z(x, y)$ be observed image intensity at pixel (x, y), and $S^{(0)}(x, y) = Z(x, y)$ be the image intensity before smoothing. Then, the smoothed image intensity at (x, y) in the $(t + 1)$-th iteration, for any $t \geq 0$, is defined by

$$S^{(t+1)}(x,y) = \frac{1}{N^{(t+1)}(x,y)} \sum_{i=-1}^{1} \sum_{j=-1}^{1} S^{(t)}(x+i, y+j) w^{(t)}(x+i, y+j), \tag{5.2}$$

where $w^{(t)}(x + i, y + j) = \exp(-(d^{(t)}(x + i, y + j))^2/(2\sigma_D^2))$ are weights, $d^{(t)}(x + i, y + j)$ is the magnitude of image gradient at $(x + i, y + j)$, $\sigma_D > 0$ is a scale parameter, and $N^{(t+1)}(x, y) =$

$\sum_{i=-1}^{1}\sum_{j=-1}^{1} w^{(t)}(x+i, y+j)$. By (5.2), $S^{(t+1)}(x,y)$ is defined by a weighted average of the restored image intensities in the 3×3 neighborhood of (x,y) obtained in the previous iteration, pixels closer to edges would receive less weights because their image gradient magnitudes are usually large, and consequently edges are preserved to a certain degree. However, a small amount of blurring is inevitable in (5.2) due to the fact that pixels located on the different side of the edge, different from the given pixel (x,y), still receive some weights, although such weights are relatively small. Iterations might help eliminate blurring; but, so far we have not found any theoretical justifications about this property yet. This phenomenon of blurring might become worse when larger neighborhoods are used.

A generalization of the above adaptive smoothing filter is the *bilateral filtering* procedure suggested by Tomasi and Manduchi (1998). By this procedure, the restored image intensity at the position $\mathbf{s} = (x,y)$ is defined by:

$$\widehat{f}(\mathbf{s}) = \frac{1}{N(\mathbf{s})}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Z(\mathbf{s}^*)K_D(\mathbf{s}^*,\mathbf{s})K_R(Z(\mathbf{s}^*), Z(\mathbf{s}))d\mathbf{s}^*, \tag{5.3}$$

where $\mathbf{s}^* = (x^*, y^*)$ is a nearby point of $\mathbf{s}$, K_D is a non-negative bivariate kernel function, K_R is a non-negative univariate kernel function, and $N(\mathbf{s})$ is a normalizing constant. Usually, the two kernel functions in (5.3) are chosen to be probability density functions of Gaussian distributions. So, the restored image intensity $\widehat{f}(\mathbf{s})$ is a weighted average of the image intensities at nearby pixels, and the weights are determined by the distance from an individual pixel site $\mathbf{s}^*$ to the given pixel site $\mathbf{s}$ and by the difference between the observed image intensities at $\mathbf{s}^*$ and $\mathbf{s}$. By comparing equation (5.3) with equation (3.1), it can be seen that the bilateral filtering procedure discussed in the image processing literature is identical to the sigma filter discussed in the statistical literature, although two different terminologies are used in the two disciplines. Thus, related discussion in Section 3 about the sigma filter also applies to the bilateral filtering procedure here.

Another popular denoising technique in the computer science literature involves *diffusion equations*, such as the following one:

$$\begin{cases} \frac{\partial Z(x,y,t)}{\partial t} = \text{div}\ (c\, G_Z(x,y,t)) \\ \text{subject to: } Z(x,y,0) = Z_0(x,y), \end{cases} \tag{5.4}$$

where "div" is the *divergence* operator, $Z_0(x,y)$ is an observed image, c is a constant, and $G_Z(x,y,t)$ denotes the gradient of $Z(x,y,t)$. Koenderink (1984) and Hummel (1987) point out that the solution of (5.4) is equivalent to the smoothed image $Z(x,y,t) = (Z_0 \otimes \phi)(x,y;t)$, where $\phi(x,y;t)$ denotes the probability density function of the 2-D Gaussian distribution $N((0,0)', t^2 I)$ and $Z_0 \otimes \phi$ denotes the convolution between Z_0 and ϕ. In physics, equation (5.4) is often used for describing diffusion processes that equilibrate concentration differences without creating or destroying mass. The constant c is called the *diffusion conductance* or *diffusivity.* People find that the degree of smoothing is controlled by the diffusivity. In the case of (5.4), all locations in the image, including edge pixels, are smoothed equally in the diffusion process. In such cases, the diffusion equation is *homogeneous.* To preserve edges during smoothing, or, to smooth *within* regions separated by edges instead of *between* such regions, the diffusivity should be ideally 0 at edge positions and 1 within separated regions. In applications, edge positions can be roughly indicated by the image gradient magnitude $|G_Z(x,y,t)|$. Therefore, the following *non-homogeneous* diffusivity can be applied to (5.4) for edge-preserving denoising:

$$c(x,y,t) = g\left(|G_Z(x,y,t)|\right),$$

where g is a decreasing function with $g(0) = 1$. In the literature, there are several generalizations and modifications of this procedure. See, e.g., Catté *et al.* (1992), Charbonnier *et al.* (1994), Weickert (1998), and Andreu *et al.* (2001) for related discussion. Relationship between non-homogeneous diffusion filtering and both adaptive smoothing and bilateral filtering is discussed by Barash (2002). Relationship between non-homogeneous diffusion filtering and wavelet shrinkage methods (cf., Chang *et al.* 2000) is studied by Mrázek *et al.* (2003).

6 Image Deblurring

The image deblurring problem, described by model (5.1), is generally ill-posed in the sense that (i) there might be many different sets of h and f corresponding to the same observed image, and (ii) the inverse problem to estimate f from Z often involves some numerical singularities (cf., equation (6.1) and the related discussion below). Therefore, *it seems impossible to estimate both h and f properly from Z alone*, without using any extra information about either f or h or both.

Early image deblurring methods assume that h is known. This assumption is reasonable in some cases, because h can be specified (at least approximately) based on our knowledge about the image acquisition device (e.g., camera). For instance, the *linear blur model* is appropriate for describing h when blurring is mainly caused by relative motion between the image acquisition device and the object. The *Gaussian blur model* is often used for describing blurring caused by atmospheric turbulence in remote sensing and aerial imaging. See Bates and McDonnell (1986) for detailed description of these models. After h is specified, f can be estimated based on the relationship that

$$\mathcal{F}\{Z\}(u,v) = \mathcal{F}\{h\}(u,v)\mathcal{F}\{f\}(u,v) + \mathcal{F}\{\varepsilon\}(u,v), \text{ for } (u,v) \in R^2, \tag{6.1}$$

where $\mathcal{F}\{f\}$ denotes the Fourier transformation of f. By equation (6.1), many methods have been proposed in the literature for estimating f, including some non-iterative methods, e.g., inverse filtering, Wiener filtering, and constrained least squares filtering procedures (cf., e.g., Gonzalez and Woods 1992, Chapter 5), and several iterative methods, e.g., Lucy-Richardson procedure, Landweber procedure, Tikhonov-Miller procedure, maximum *a posteriori* (MAP) procedure, maximum entropy procedure, procedures based on EM algorithm, and so forth (cf., e.g., Skilling 1989, Carraso 1999, Figueiredo and Nowak 2003). The Wiener filter, for instance, defines the restored image by

$$\widehat{f}(x,y) = \frac{1}{(2\pi)^2}\mathcal{R}\left\{\int\int \frac{\overline{\mathcal{F}\{h\}(u,v)}}{|\mathcal{F}\{h\}(u,v)|^2 + \alpha(u^2+v^2)^{\beta/2}}\mathcal{F}\{Z\}(u,v)\exp\{i(ux+vy)\}\,dudv\right\}, \tag{6.2}$$

where $\overline{\mathcal{F}\{h\}(u,v)}$ denotes the complex conjugate of $\mathcal{F}\{h\}(u,v)$, $\mathcal{R}\{C\}$ denotes the real part of the complex number C, and $\alpha, \beta > 0$ are two parameters.

In equation (6.1), if the noise term is ignored, then the restored image by the Wiener filter should be defined by (6.2) without the term $\alpha(u^2 + v^2)^{\beta/2}$ in the denominator of the integrant. However, in such cases, noise effect would dominate the image estimator because $\mathcal{F}\{f\}(u,v)$ and $\mathcal{F}\{h\}(u,v)$ usually converge to zero rapidly, as $u^2 + v^2$ tends to infinity, and $\mathcal{F}\{\varepsilon\}(u,v)$ converges to zero much less rapidly. How to diminish the noise effect turns out to be a major challenge for image restoration in cases when h is known. Inclusion of $\alpha(u^2 + v^2)^{\beta/2}$ in (6.2) is mainly for this purpose. It has been proved that the Wiener filter is optimal in minimizing the mean integrated square error of the restored image when h is assumed known, when the noise is assumed Gaussian, and when some other regularity conditions are satisfied (cf., Gonzalez and Woods 1992, Chapter 5).

In many applications, however, it is difficult to specify psf h completely, based on our prior knowledge about the image acquisition device. Image restoration when h is unknown is often referred to as the *blind image restoration* problem. In the literature, a number of procedures have been proposed for solving this problem, which can be grouped roughly into two categories. One type of such procedures assumes that h can be described by a parametric model with one or more unknown parameters, and the parameters together with the true image are estimated by some algorithms, most of which are iterative (cf., e.g., Cannon 1976, Katsaggelos and Lay 1990, Rajagopalan and Chaudhuri 1999, Carasso 2001, Joshi and Chaudhuri 2005). For instance, Carasso (2001) assumes that h belongs to the symmetric Lévy "stable" density family whose Fourier transformation is defined by

$$\mathcal{F}\{h\}(u,v) = \exp\left\{-\xi(u^2+v^2)^\eta\right\}, \text{ for } (u,v) \in R^2,$$

where $\xi > 0$ and $0 < \eta \leq 1$ are two parameters. See Hall and Qiu (2007a) for a related discussion. The other type of procedures assumes that the true image comprises of an object with known finite support, the background is uniformly black, gray, or white, and the psf h satisfies various conditions (e.g., Yang *et al.* 1994, Kundur and Hatzinakos 1998). For instance, Kundur and Hatzinakos (1998) assumes that h has its inverse, both h and its inverse have finite integrations over the real plane, and both the true image f and the psf h are *irreducible* in the sense that each of them can not be expressed as a convolution of two or more component images with finite support.

Apparently, validity of parametric models used by the first group of methods for describing psf h should be checked in a specific application. If such models do not describe the underlying blurring mechanism well, then the quality of the restored images is in question because numerical studies have shown that restored images are quite sensitive to correct specification of h (cf., e.g., Carasso 2001, Hall and Qiu 2007a). So far, it is still an open problem how to make such a goodness-of-fit model checking. Regarding the second group of methods, some of their model assumptions may not be realistic. For instance, some commonly used h functions, including the Gaussian blurring function, do not have integrable inverses.

In the literature, there are some other blind image deblurring procedures. One group of such methods is based on the *total variation (TV)* minimization method first proposed by Rudin *et al.* (1992). For instance, Chan and Wong (1998) formulate the blind deblurring problem as

$$\min_{\widetilde{f},\widetilde{h}} \left\{ \frac{1}{2}\int_\Omega [(\widetilde{h}\otimes\widetilde{f})(x,y) - Z(x,y)]^2\, dxdy + \alpha_1 \int_\Omega |\bigtriangledown \widetilde{f}(x,y)|\, dxdy + \alpha_2 \int_\Omega |\bigtriangledown \widetilde{h}(x,y)|\, dxdy \right\}, \tag{6.3}$$

where Ω is the image design space, and α_1 and α_2 are two positive parameters. Solutions of (6.3) are used as estimators of f and h. Clearly, in (6.3), the first term measures the goodness-of-fit of the estimators, and the second and third terms regularize their total variations. Chan and Wong solve the minimization problem by an iterative algorithm, after α_1 and α_2 are properly selected. See You and Kaveh (1996) for a similar procedure. Recently, Hall and Qiu (2007b) suggest estimating the psf h from an observed test image of an imaging device, and then restore other observed images by the same imaging device using the estimated h. The true test image considered in the paper has a square block in the middle with a uniform background. The idea in the paper, however, can also be applied to cases when no such test images are available but the observed image to deblur has one or more parts in which the true image has step edges at known locations with uniform image intensities on either side of any of those step edges.

7 Some Concluding Remarks

In the previous sections, we provided a review of recent methodologies on jump surface estimation, edge detection, and image restoration. From the review, it can be seen that, although different terminologies are used in the two disciplines, jump detection and jump surface estimation in statistics are essentially the same problems as edge detection and image restoration in image processing. Therefore, it can benefit both research areas if the strengths of the two groups of methods can be combined in a proper way.

Besides statistics, computer sciences, and electronic engineering, surface estimation is a research topic in several other disciplines, including numerical mathematics (e.g., Cheney and Kincaid 1994), computer-aided design (e.g., Farin 1993), meteorology (e.g., Bai and Feng 2003), geology(e.g., Bonan *et al.* 2001), and so forth. Some curve/surface estimation techniques proposed in these areas might be helpful in handling jump surface estimation and image analysis. For instance, Bezier curves, commonly used in computer-aided design, might be more flexible than the PC lines used in the second stage of the three-stage jump surface estimation procedure discussed in Section 3, and Bezier surfaces have a great potential to be used locally for detecting and preserving jumps in surface estimation.

From model (5.1), it can be seen that the image deblurring problem has a similar nature to the deconvolution problem (e.g., Carroll and Hall 1988, Fan 1991) and the error-in-variables regression problem (e.g., Hall and Qiu 2005, Delaigle *et al.* 2006), both of which are well discussed in the statistical literature. Although formulations of the three problems are different in one way or another, all of them have convolutions involved and belong to the so-called inverse problems. Therefore, existing methods for handling the deconvolution problem and the error-in-variables regression problem might be helpful in solving the image deblurring problem.

Acknowledgments: The author would like to thank two reviewers for some constructive comments and suggestions which greatly improved the quality of the paper. This research is supported in part by an NSA grant and an NSF grant.

References

Andreu, F., Ballester, C., Caselles, V., and Mazón, J.M. (2001), "Minimizing total variation flow," *Differential and Integral Equations*, **14**, 321–360.

Arce, G.R. (1991), "Multistage order statistic filters for image sequence," *IEEE Transactions on Signal Processing*, **39**, 1146–1163.

Bai, Z., and Feng, Y. (2003), "GPS water vapor estimation using interpolated surface meteorological data from Australian automatic weather stations," *Journal of Global Positioning Systems*, **2**, 83–89.

Barash, D. (2002), "A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 844–847.

Bates, R.H.T., and McDonnell, M.J. (1986), *Image Restoration and Reconstruction,* Oxford: Clarendon Press.

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

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

Blake, A., and Zisserman, A. (1987), *Visual Reconstruction*, Cambridge, MA: MIT Press.

Bonan, G.B., Oleson, K.W., Vertenstein, M., Levis, S., Zeng, X., Dai, Y., Dickinson, R.E., and Yang, Z.-L. (2002), "The land surface climatology of the Community Land Model coupled to the NCAR Community Climate Model," *Journal of Climate*, **15**, 3123–3149.

Bouman, C., and Sauer, K. (1993), "A generalized Gaussian image model for edge preserving MAP estimation," *IEEE Transactions on Image Processing*, **2**, 296–310.

Box, G.E.P., and Cox, D.R. (1964), "An analysis of transformations," *Journal of the Royal Statistical Society (Series B)*, **26**, 211–246.

Cannon, M. (1976), "Blind deconvolution of spatially invariant image blurs with phase," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **24**, 58–63.

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

Carasso, A.S. (1999), "Linear and nonlinear image deblurring – A documented study." *SIAM Journal on Numerical Analysis*, **36**, 1659–1689.

Carasso, A.S. (2001), "Direct blind deconvolution," *SIAM Journal on Applied Mathematics*, **61**, 1980–2007.

Carroll, R.J., and Hall, P. (1988), "Optimal rates of convergence for deconvolving a density," *Journal of the American Statistical Association*, **83**, 1184–1186.

Caselles, V., Catté, F., Coll, T., and Dibos, F. (1993), "A geometric model for active contours in image processing," *Numerische Mathematik*, **66**, 1–31.

Caselles, V., Kimmel, R., and Sapiro, G. (1997), "On geodesic active contours," *International Journal of Computer Vision*, **22**, 61–79.

Catté, F., Lions, P.L., Morel, J.M., and Coll, T. (1992), "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM Journal on Numerical Analysis*, **29**, 182-193.

Chan, T.F., and Luminita, A.V. (2001), "Active contour without edges," *IEEE Transactions on Image Processing*, **10**, 266–277.

Chan, T.F., and Wong, C.K. (1998), "Total variation blind deconvolution," *IEEE Transactions on Image Processing*, **7**, 370–375.

Chang, G.S., Yu, B., and Vetterli, M. (2000), "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on Image Processing*, **9**, 1522–1531.

Charbonnier, P., Blanc-Féraud, L., Aubert, G., and Barlaud, M. (1994), "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proceedings of IEEE International Conference on Image Processing*, **2**, 168–172, Austin, TX.

Chen, M.H., and Chin, R.T. (1993), "Partial smoothing splines for noisy plus boundaries with corners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1208–1216.

Cheney, W., and Kincaid, D. (1994), *Numerical Mathematics and Computing (3rd ed.)*, Pacific Grove, CA: Brooks/Cole Publishing Company.

Chu, C.K., Glad, I.K., Godtliebsen, F., and Marron, J.S. (1998), "Edge-preserving smoothers for image processing (with discussion)", *Journal of the American Statistical Association*, **93**, 526–556.

Delaigle, A., Hall, P., and Qiu, P. (2006), "Nonparametric methods for solving the Berkson error-in-variables problem," *Journal of the Royal Statistical Society (Series B)*, **68**, 201–220.

Demigny, D., and Kamlé (1997), "A discrete expression of Canny's criteria for step edge detector performances evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 1199–1211.

Duda, R.O., and Hart, P.E. (1973), *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons.

Fan, J. (1991), "On the optimal rates of convergence for nonparametric deconvolution problems," *The Annals of Statistics*, **19**, 1257–1272.

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

Farin, G. (1993), *Curves and Surfaces for Computer Aided Geometric Design (3rd ed.)*, New York: Academic Press, Inc..

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

Figueiredo, M.A.T., and Nowak, R.D. (2003), "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, **12**, 906–916.

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

Garlipp, T., and Müller, Ch.H. (2004), "Robust jump detection in regression surface," (submitted), http://www.mathematik.unikassel.de/~cmueller/publikation.html.

Garlipp, T., and Müller, Ch.H. (2006), "Detection of linear and circular shapes in image analysis," *Computational Statistics and Data Analysis,* **51**, 1479–1490.

Geman, S., and Geman, D. (1984), “Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721–741.

Gijbels, I., and Goderniaux, A.C. (2004), “Bandwidth selection for changepoint estimation in nonparametric regression,” *Technometrics*, **46**, 76-86.

Gijbels, I., Lambert, A., and Qiu, P. (2006), “Edge-preserving image denoising and estimation of discontinuous surfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 1075–1087.

Gijbels, I., Lambert, A., and Qiu, P. (2007), “Jump-preserving regression and smoothing using local linear fitting: a compromise,” *Annals of the Institute of Statistical Mathematics*, (in press).

Godtliebsen, F., and Sebastiani, G. (1994), “Statistical methods for noisy images with discontinuities,” *Journal of Applied Statistics*, **21**, 459–477.

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

Green, P. (1990), “Bayesian reconstructions from emission tomography data using a modified EM algorithm,” *IEEE Transactions on Medical Imaging*, **9**, 84–93.

Haavisto, P., Gabbouj, M., and Neuvo, Y. (1991), “Median based idempotent filters,” *Journal of Circuits, Systems, and Computers*, **1**, 125–148.

Hall, P., Peng, L., and Rau, C. (2001), “Local likelihood tracking of fault lines and boundaries,” *Journal of the Royal Statistical Society (Series B)*, **63**, 569–582.

Hall, P., and Qiu, P. (2005), “Discrete transform approach to errors-in-variables problems,” *Biometrika*, **92**, 135–148.

Hall, P., and Qiu, P. (2007a), “Blind deconvolution and deblurring in image analysis,” *Statistica Sinica*, (in press).

Hall, P., and Qiu, P. (2007b), “Nonparametric estimation of a point spread function in multivariate problems,” *The Annals of Statistics*, (in press).

Hall, P., Qiu, P., and Rau, C. (2007), “Tracking edges, corners and vertices in an image,” *Scandinavian Journal of Statistics*, (in press).

Hall, P., and Rau, C. (2000), “Tracking a smooth fault line in a response surface”, *The Annals of Statistics*, **28**, 713–733.

Hall, P., and Rau, C. (2002), “Likelihood-based confidence bands for fault lines in response surfaces,” *Probability Theory and Related Fields*, **124**, 26–49.

Hall, P., and Titterington, M. (1992), “Edge-preserving and peak-preserving smoothing,” *Technometrics*, **34**, 429–440.

Hansen, F.R., and Elliot, H. (1982), “Image segmentation using simple Markov field models,” *Computer Graphics and Image Processing*, **20**, 101–132.

Haralick, R.M. (1984), “Digital step edges from zero crossing of second directional derivatives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 58–68.

Härdle, W. (1990), *Smoothing Techniques: With Implementation in S*, New York: Springer–Verlag.

Heath, M., Sarkar, S., Sanocki, T., and Bowyer, K.W. (1997), “A robust visual method for assessing the relative performance of edge detection algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 1338–1359.

Heath, M., 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.

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

Hummel, A. (1987), “Representations based on zero-crossings in scale-space,” in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, 204–209.

Joshi, M.V., and Chaudhuri, S. (2005), “Joint blind restoration and surface recovery in photometric stereo,” *Journal of the Optical Society of America, Series A*, **22**, 1066–1076.

Kass, M., Witkin, A., and Terzopoulos, D. (1988), “Snakes: active contour models,” *International Journal of Computer Vision*, **1**, 321–331.

Katsaggelos, A.K., and Lay, K.-T. (1990), “Image identification and image restoration based on the expectation-maximization algorithm,” *Optical Engineering*, **29**, 436–445.

Koenderink, J. (1984), “The structure of images,” *Biological Cybernetics*, **50**, 363–370.

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

Kundur, D., and Hatzinakos, D. (1998), “A novel blind deconvolution scheme for image restoration using recursive filtering,” *IEEE Transactions on Signal Processing*, **46**, 375–390.

Lange, K. (1990), “Convergence of EM image reconstruction algorithm with Gibbs smoothing,” *IEEE Transactions on Medical Imaging*, **9**, 439–446.

Li, S.Z. (1995), *Markov random field modeling in computer vision,* New York: Springer–Verlag.

Li, S.Z. (1998), “Close-form solution and parameter selection for convex minimization-based edge-preserving smoothing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 916–932.

Malladi, R., Sethian, J.A., and Vemuri, B.C. (1995), “Shape modeling with front propagation: a level set approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 158–175.

Marr, D., and Hildreth, E. (1980), “Theory of edge detection,” *Proceedings of the Royal Society in London*, **B207**, 187–217.

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

McDonald, J.A., and Owen, A.B. (1986), "Smoothing with split linear fits," *Technometrics*, **28**, 195–208.

Mrázek, P., Weickert, J., and Steidl, G. (2003), "Correspondences between wavelet shrinkage and nonlinear diffusion," in *Scale-Space 2003* (L.D. Griffin and M. Lillholm, eds.), 101–116.

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

Müller, H.G. (1992), "Change-points in nonparametric regression analysis," *The Annals of Statistics*, **20**, 737-761.

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

Müller, H.G., and Song, K.S. (1994b), "Cube splitting in multidimensional edge estimation," in *Change-Point Problems* (edited by Carlstein, E., Müller, H.G., and Siegmund, D.), IMS Lecture Notes, **23**, 210–223.

Nadaraya, E.A. (1964), "On estimating regression," *Theory of Probability and Its Applications*, **9**, 141–142.

Nalwa, V.S., and Binford, T.O. (1986), "On detecting edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 699–714.

Osher, S., and Sethian, J.A. (1988), "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulation," *Journal of Computational Physics*, **79**, 12–49.

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.

Parzen, E. (1962), "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, **33**, 1065–1076.

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.

Petrou, M., and Kittler, J. (1991), "Optimal edge detectors for ramp edges," *Transactions on Pattern Analysis and Machine Intelligence*, **13**, 483–491.

Poggio, T., Torre, V., and Koch, C. (1985), "Computational vision and regularization theory," *Nature*, **317**, 314–319.

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

Qiu, P. (1991), "Estimation of a kind of jump regression functions," *Systems Science and Mathematical Sciences,* **4**, 1–13.

Qiu, P. (1994), "Estimation of the number of jumps of the jump regression functions," *Communications in Statistics-Theory and Methods*, **23**, 2141–2155.

Qiu, P. (1997), "Nonparametric estimation of the jump regression surface," *Sankhya (Series A)*, **59**, 268–294.

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

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

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

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

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

Qiu, P. (2006), "Jump-preserving surface reconstruction from noisy data," (manuscript).

Qiu, P., Asano, Chi., and Li, X. (1991), "Estimation of jump regression functions," *Bulletin of Informatics and Cybernetics*, **24**, 197–212.

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 Sun, J. (2006), "Using conventional edge detectors and post-smoothing for segmentation of spotted microarray images," *Technical Report #651*, School of Statistics, University of Minnesota.

Qiu, P., and Sun, J. (2007), "Local smoothing image segmentation for spotted microarray images," *Journal of the American Statistical Association*, (in press).

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

Qiu, P., and Yandell, B. (1998), "A local polynomial jump detection algorithm in nonparametric regression," *Technometrics*, **40**, 141–152.

Rajagopalan, A.N., and Chaudhuri, S. (1999), "MRF model-based identification of shift-variant point spread function for a class of imaging systems," *Signal Processing*, **76**, 285–299.

Rosenblatt, M. (1969), "Conditional probability density and regression estimates," in *Multivariate Analysis II* (Krishnaiah, ed.), 25–31, New York: Academic Press.

Rosenfeld, A., and Kak, A.C. (1982), *Digital Picture Processing (2nd edition),* 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.

Rudin, L.I., Osher, S., and Fatemi, E. (1992), "Nonlinear total variation based noise removal algorithms," *Physica D*, **60**, 259–268.

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.

Shiau, J. (1985), "Smoothing spline estimation of functions with discontinuities," *Technical Reports # 768*, Department of Statistics, University of Wisconsin.

Shiau, J. (1987), "A note on MSE coverage intervals in a partial Spline model," *Communications in Statistics – Theory and Methods*, **16**, 1851–1866.

Shiau, J., 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.

Shin, M.C., Goldgof, D., and Bowyer, K.W. (2001), "Comparison of edge detector performance through use in an object recognition task," *Computer Vision and Image Understanding*, **84**, 160–178.

Skilling, J. (1989, eds), *Maximum Entropy and Bayesian Methods,* Norwell, MA: Kluwer Academic.

Stanton, J.M. (2001), "Galton, Pearson, and the Peas: a brief history of linear regression for statistics instructors," in the electronic journal *Journal of Statistics Education*, **9(1)**, of the following web address: http://www.amstat.org/publications/jse/v9n3/stanton.html.

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*, (in press).

Stevenson, R.L., Schmitz, B.E., and Delp, E.J. (1994), "Discontinuity preserving regularization of inverse visual problems," *IEEE Transactions on Systems, Man, and Cybernetics*, **24**, 455–469.

Tan, H.L., Gelfand, S.B., and Delp, E.J. (1992), "A cost minimization approach to edge detection using simulated annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 3–18.

Tomasi, C., and Manduchi, R. (1998), "Bilateral Filtering for Gray and Color Images," *Proceedings of the 1998 IEEE International Conference on Computer Vision*, 839–846, Bombay, India.

Wahba, G. (1990), *Spline Models for Observational Data,* CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia, PA: SIAM.

Wang, Y. (1995), "Jump and sharp cusp detection by wavelets," *Biometrika*, **82**, 385–397.

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

Watson, G.S. (1964), "Smooth regression analysis," *Sankhya (Series A)*, **26**, 359–372.

Weickert, J. (1998), *Anisotropic Diffusion in Imaging Processing*, Stuttgart, Germany: Teubner.

Wu, J.S., and Chu, C.K. (1993), "Kernel type estimators of jump points and values of a regression function," *The Annals of Statistics*, **21**, 1545–1566.

Yang, Y.H., Buckley, M.J., Dudoit, S., and Speed, T. (2002), "Comparison of Methods for Image Analysis on cDNA Microarray Data," *Journal of Computational and Graphical Statistics*, **11**, 108–136.

Yang, Y., Galatsanos, N.P., and Stark, H. (1994), "Projection-based blind deconvolution," *Journal of the Optical Society of America A*, **11**, 2401–2409.

Yitzhaky, Y., and Peli, E. (2003), "A method for objective edge detection evaluation and detector parameter selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, 1027–1033.

You, Y.L., and Kaveh, M. (1996), "A regularization approach to joint blur identification and image restoration," *IEEE Transactions on Image Processing*, **5**, 416–428.