Edge Detection and Image Restoration of Blurred Noisy Images Using Jump Regression Analysis

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yicheng Kang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Peihua Qiu, Adviser

August 2013

Acknowledgements

Many thanks to the School of Statistics at the University of Minnesota for creating free and friendly academic environment. I have enjoyed my time in Minneapolis and this will be an unforgettable memory.

Special thanks to my adviser, Peihua Qiu, for constant encouragement and personal help. His sharp intuition and deep thoughts have inspired me to explore all kinds of image processing techniques and related statistical problems. With his expertise and friendship, I have leaned more statistics than I ever imagined.

I would also like to express gratitude to my committee members, Professor Douglas Hawkins, Professor Yuhong Yang, and Associate Professor Haitao Chu, who provided encouraging and constructive feedback. It is no easy task, reviewing a thesis, and I am grateful for their thoughtful and detailed comments.

Most of all, I would like to thank my wife, Celia, for her great support and understanding when I spent days and nights working on this dissertation in front of computer. My three years of graduate study would have been impossible without her excellent job at managing our household.

Abstract

We consider the problem of edge-preserving image restoration when images are degraded by spatial blur and pointwise noise. When the spatial blur described by a point spread function (psf) is not completely specified beforehand, this is a challenging "ill-posed" problem, because (i) theoretically, the true image can not be uniquely determined by the observed image when the psf is unknown, even in cases when the observed image contains no noise, and (ii) practically, besides blurring, observed images often contain noise, which can cause numerical instability in many existing image deblurring procedures. In the literature, most existing deblurring procedures are developed under the assumption that the psf is completely specified, or that the psf follows a parametric form with one or more unknown parameters. In this dissertation, we propose blind image deblurring (BID) methodologies that do not require such restrictive conditions on the psf. They even allow the psf to change over location. This dissertation has three chapters. Chapter 1 introduces some motivating applications for image processing along with presenting the overall scope of the dissertation. In Chapter 2, the problem of step edge detection in blurred noisy images is studied. In Chapter 3, a BID procedure based on edge detection is proposed. In Chapter 4, an efficient BID procedure without explicitly detecting edges is presented. Both theoretical justifications and numerical studies show that our proposed procedures work well in applications.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Images are everywhere in our daily life. This is not only because image is a widely used medium of communications, but also because it is an easy and compact way to represent the physical world. For example, medical professionals make substantial use of medical images for diagnostic purpose, e.g., measuring the size and the shape of a tumor via medical imaging. Satellite images are used to track and quantify changes in forests, water supplies, pollution and so forth. These and similar examples demonstrate that image processing plays an important role in the modern society.

In practice, however, images are not always faithful representations of the scenes that we see. Degradations can occur during the entire process of image acquisition. 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. See Figure 1.1 for a better understanding of the two types of degradations. This is a toy example of an image that contains a single edge. It can be seen that noise alters individual gray level in a random pattern but alterations of individual gray levels caused by blur are spatially correlated. In a spatially blurred image, edge structures are often blurred by some smoothing process.

The mechanism behind this kind of smoothing process is often unknown and can be variant over location in applications (e.g., atmospheric turbulence that degrades satellite images causes location variant blur because direction intensity of turbulence often differ from location to location), making spatial degradations relatively difficult to handle. Other types of degradations involve chromatic or temporal effects. See Bates and McDonnell (1986) for a detailed discussion about formation of various degradations.

In many applications, it is of interest to extract edge information from images because edges often represent outlines of the object or abrupt structural changes in the related 2-D process represented by the image (e.g, places where equi-temperature surfaces in high sky or deep ocean change dramatically). In such cases, the major challenge is to detect edges properly based on the observed image, which is the goal of *edge detection* in image processing. In some other applications, it is often important to remove noise and blur (i.e., denoising and deblurring) from the observed image for better human interpretation and machine perception. Thus, *image restoration* is another important research topic in image processing.

Jump regression analysis (JRA) provides a statistical tool for jump detection and surface estimation of regression surfaces with discontinuities. The intensity function of the image gray levels can be regarded as a 2-D regression surface and the edges correspond to the jumps of the regression function and its derivatives of various orders (e.g., step edges correspond to jumps in the regression function and roof/valley edges correspond to jumps in the first order derivatives of the regression function). Then the problem of edge detection and image restoration can fit well under the framework of JRA. See a detailed discussion about the connections and differences between image processing and JRA in Qiu (2005). Most JRA procedures, however, assume either there is no spatial blur involved in the observed image or the blurring mechanism is known to some extent. In this dissertation, JRA methodologies are extended to

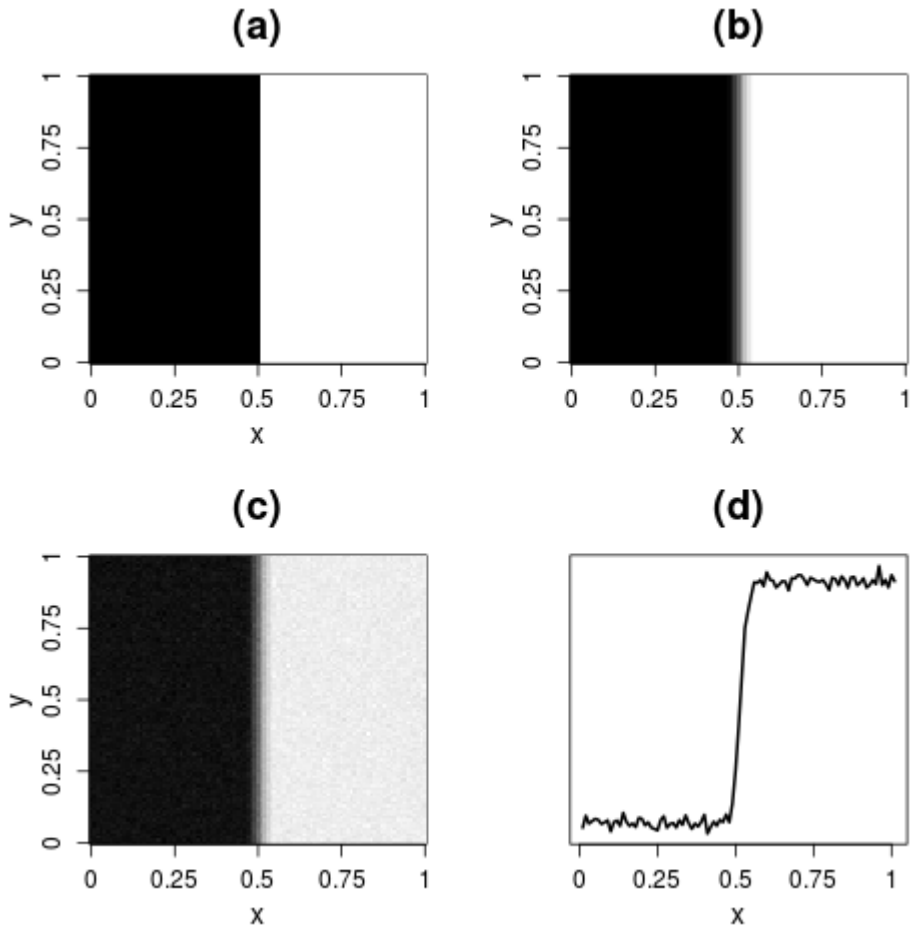

Figure 1.1: (a)-(c): Original image with a single edge, a blurred version and a noisy blurred version, respectively. (d): The cross section at the line $y = 0.5$ of the image shown in (c).

handle the problem of edge detection and image restoration when both pointwise degradation and spatial blur are present without assuming restrictive conditions on the blurring mechanism.

The remaining part of this dissertation is organized as follows. Edge detection in blurred noisy images is discussed in Chapter 2. A BID procedure based on edge detection is proposed in Chapter 3. And in Chapter 4, an efficient BID procedure without detecting edges explicitly is presented.

Chapter 2

Edge Detection In Blurred Noisy Images

2.1 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 step edges of the related image intensity surface in the image processing literature (cf., Gonzalez and Woods (2002), Qiu (2005)). Step edges in an image often convey much information of an image, as discussed in the previous chapter. Therefore, it is important to detect the step edges accurately from the observed images. This chapter focuses on step edge detection in images when both spatial blur and pointwise noise are present. Throughout this chapter, edges refer to step edges if there is no further specification.

In the literature, most existing edge 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 intensity functions. See, for instance, Chu et al. (2012), Garlipp and Müller (2006), Garlipp and Müller (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 (1984) in the image processing literature.

In this chapter, we propose four edge detectors. The idea behind our proposed methods can be explained intuitively as follows. First, although spatial blur would alter the image structure, especially around the true step edge curves, the blurred image would change most dramatically at the true step edge curves along their normal directions. Therefore, appropriate estimators of the gradients of the image intensity can be used for edge detection. Second, spatial blur is a smoothing process; it smooths the image at the true step edges. If we can recover the edge structure to a certain degree when removing noise, it may help detect edges. Our proposed edge detectors make use of these properties of spatial blur in different ways. One major feature of these edge detectors is that they do not require restrictive assumptions on either the psf that describes the blurring mechanism or the true image. Their theoretical and numerical properties are studied and compared. Further, in this chapter, we propose a new quantitative metric for measuring the performance of an edge detector. A data-driven bandwidth selection procedure via bootstrap is suggested as well.

The remaining part of the chapter is organized as follows. In the next section, our proposed edge detectors are described in detail. Some of their statistical properties are presented in Section 2.3. Their numerical performance is evaluated in Section 2.4. Some discussions are presented in Section 2.5. Technical details are provided in Section 2.6.

2.2 Proposed Edge Detectors

In this section, we first describe our proposed edge detectors in detail in Subsection 2.2.1, and then propose a new metric for measuring the edge detection performance

in Subsection 2.2.2. Based on the proposed performance measure, a bootstrap bandwidth selection procedure is suggested in that subsection as well.

2.2.1 Four edge detectors based on local kernel smoothing

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

$$Z(x,y) = H\{f\}(x,y) + \varepsilon(x,y), \qquad \text{for } (x,y) \in \Omega, \tag{2.1}$$

where $H\{f\}(x,y) = \int\int_{\mathbf{R}^2} h(u,v;x,y) f(x-u, y-v)\, dudv$ denotes the convolution between a 2-D *point spread function* (psf) h and the true image f, $\varepsilon(x,y)$ is the pointwise noise at (x,y), and Ω is the design space of the image. In model ((2.1)), it is assumed that the true image f is degraded spatially by h and pointwise by ε, that the spatial blur is linear, and that the pointwise noise is additive. In most references, people further assume that the psf h, which describes the spatial blurring mechanism, is location invariant. That is, $h(u,v;x,y)$ does not depend on (x,y). See Hall and Qiu (2007b) for a related discussion.

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

$$Z_{ij} = H\{f\}(x_i, y_j) + \varepsilon_{ij}, \qquad \text{for } i,j = 1,2,\cdots,n, \tag{2.2}$$

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

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

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

$$\min_{a,b,c} \sum_{i^2+j^2 \le k^2} \{Z(x+i/n, y+j/n) - [a + b(i/n) + c(j/n)]\}^2 K\left(\frac{i}{k}, \frac{j}{k}\right), \qquad (2.3)$$

where $Z(x+i/n, y+j/n)$ denotes the observed intensity at pixel $(x+i/n, y+j/n)$ 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 (2.3) be denoted as $\{\widehat{a}(x,y), \widehat{b}(x,y), \widehat{c}(x,y)\}$. Then, $(\widehat{b}(x,y), \widehat{c}(x,y))$ is an estimator of the gradient of the true intensity function 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 $(\widehat{b}(x,y), \widehat{c}(x,y))$. 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 step edge segment. Let

$$\begin{aligned} \widehat{f}_{LCK,+}(x,y) &= \frac{\sum_{U_n(x,y)} Z(x+i/n, y+j/n) K\left(\frac{i}{k}, \frac{j}{k}\right)}{\sum_{U_n(x,y)} K\left(\frac{i}{k}, \frac{j}{k}\right)}, \\ \widehat{f}_{LCK,-}(x,y) &= \frac{\sum_{V_n(x,y)} Z(x+i/n, y+j/n) K\left(\frac{i}{k}, \frac{j}{k}\right)}{\sum_{V_n(x,y)} K\left(\frac{i}{k}, \frac{j}{k}\right)} \end{aligned} \qquad (2.4)$$

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, $\widehat{f}_{LCK,+}(x,y) - \widehat{f}_{LCK,-}(x,y)$ 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 edge detection criterion is defined by its standardized version

$$\mathrm{LCK}_n(x,y) = \frac{\widehat{f}_{LCK,+}(x,y) - \widehat{f}_{LCK,-}(x,y)}{\sqrt{\frac{\sum_{U_n(x,y)} K\left(\frac{i}{k}, \frac{j}{k}\right)^2}{\left[\sum_{U_n(x,y)} K\left(\frac{i}{k}, \frac{j}{k}\right)\right]^2} + \frac{\sum_{V_n(x,y)} K\left(\frac{i}{k}, \frac{j}{k}\right)^2}{\left[\sum_{V_n(x,y)} K\left(\frac{i}{k}, \frac{j}{k}\right)\right]^2}}}, \qquad (2.5)$$

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

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

$$\mathrm{LCK}_n(x, y) \xrightarrow{d} N\left(0, \sigma^2\right), \text{ as } n \longrightarrow \infty \tag{2.6}$$

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

$$\mathrm{LCK}_n(x, y) \xrightarrow{P} \infty, \text{ as } n \longrightarrow \infty. \tag{2.7}$$

Therefore, the criterion $\mathrm{LCK}_n(x, y)$ does contain useful information for edge detection. Our proposed edge detector LCK can be summarized as follows.

- At a given pixel (x_i, y_j), for $i, j = 1, 2, \cdots, n$, solve the minimization problem (2.3) 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 (2.4).

- The pixel (x_i, y_j) is flagged as a detected edge pixel if

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

  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 edge detector LCK does not take into account the possible blurring in the observed image. In cases when blurring is present, if a pixel 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 edge detection criterion is defined by

$$\mathrm{LC2K}_n(x,y) = \frac{\widehat{f}_{LC2K,+}(x,y) - \widehat{f}_{LC2K,-}(x,y)}{\sqrt{\frac{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right)^2 L(d_{ij}/(k/n))^2}{\left[\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L(d_{ij}/(k/n))\right]^2} + \frac{\sum_{V_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right)^2 L(d_{ij}/(k/n))^2}{\left[\sum_{V_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L(d_{ij}/(k/n))\right]^2}}}, \tag{2.8}$$

where

$$\begin{aligned}
\widehat{f}_{LC2K,+}(x,y) &= \frac{\sum_{U_n(x,y)} Z(x+i/n, y+j/n) K\left(\frac{i}{k},\frac{j}{k}\right) L(d_{ij}/(k/n))}{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L(d_{ij}/(k/n))}, \\
\widehat{f}_{LC2K,-}(x,y) &= \frac{\sum_{V_n(x,y)} Z(x+i/n, y+j/n) K\left(\frac{i}{k},\frac{j}{k}\right) L(d_{ij}/(k/n))}{\sum_{V_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L(d_{ij}/(k/n))},
\end{aligned} \tag{2.9}$$

L is a univariate increasing density kernel function with support $[0,1]$, and d_{ij} is the Euclidean distance from the pixel (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 edge detector is labeled as LC2K, where LC2K denotes local constant smoothing with 2 kernel functions. The edge detection criterion $\mathrm{LC2K}_n(x,y)$ has similar asymptotic results to (2.6) and (2.7). Also, a numerical algorithm of the edge detector LC2K can be developed in a similar way to that of the

edge detector LCK.

In cases when the true intensity function f is quite steep but continuous around a given point (x, y), by the edge detectors LCK and LC2K, (x, y) can be falsely detected as an edge pixel because both their edge detection criteria $\mathrm{LCK}_n(x, y)$ and $\mathrm{LC2K}_n(x, y)$ 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., (2.4)), we consider fitting local planes by solving the following local linear kernel (LLK) smoothing problems:

$$\min_{a_0,a_1,a_2} \sum_{U_n(x,y)} \{Z(x + i/n, y + j/n) - [a_0 + a_1(i/n) + a_2(j/n)]\}^2 K\left(\frac{i}{k}, \frac{j}{k}\right), \quad (2.10)$$

and

$$\min_{a_0,a_1,a_2} \sum_{V_n(x,y)} \{Z(x + i/n, y + j/n) - [a_0 + a_1(i/n) + a_2(j/n)]\}^2 K\left(\frac{i}{k}, \frac{j}{k}\right). \quad (2.11)$$

The solutions to a_0 of (2.10) and (2.11) are denoted as $\widehat{f}_{LLK,+}(x, y)$ and $\widehat{f}_{LLK,-}(x, y)$, respectively, and they have the expressions

$$\begin{aligned} \widehat{f}_{LLK,+}(x, y) &= \frac{\sum_{U_n(x,y)} w_{ij}(x, y) Z(x + i/n, y + j/n)}{\sum_{U_n(x,y)} w_{ij}(x, y)} \\ \widehat{f}_{LLK,-}(x, y) &= \frac{\sum_{V_n(x,y)} w'_{ij}(x, y) Z(x + i/n, y + j/n)}{\sum_{V_n(x,y)} w'_{ij}(x, y)}, \end{aligned} \quad (2.12)$$

where

$$\begin{aligned} w_{ij}(x, y) &= A_1(x, y) + A_2(x, y)(x_i - x) + A_3(x, y)(y_j - y), \\ A_1(x, y) &= r_{20}(x, y) r_{02}(x, y) - r_{11}(x, y) r_{11}(x, y), \end{aligned}$$

$$\begin{aligned} A_2(x,y) &= r_{01}(x,y)r_{11}(x,y) - r_{10}(x,y)r_{02}(x,y), \\ A_3(x,y) &= r_{10}(x,y)r_{11}(x,y) - r_{01}(x,y)r_{20}(x,y), \\ r_{s_1,s_2}(x,y) &= \sum_{U_n(x,y)} (i/n)^{s_1}(j/n)^{s_2} K\left(\frac{i}{k},\frac{j}{k}\right), \qquad \text{for } s_1, s_2 = 0,1,2, \end{aligned}$$

and $w'_{ij}(x,y)$ are defined in the same way as $w_{ij}(x,y)$ except that $U_n(x,y)$ in the definition should be replaced by $V_n(x,y)$. Then, we define our third edge detection criterion by

$$\mathrm{LLK}_n(x,y) = \frac{\widehat{f}_{LLK,+}(x,y) - \widehat{f}_{LLK,-}(x,y)}{\sqrt{\frac{\sum_{(U_n(x,y)} w_{ij}(x,y)^2}{[\sum_{U_n(x,y)} w_{ij}(x,y)]^2} + \frac{\sum_{V_n(x,y)} w'_{ij}(x,y)^2}{[\sum_{V_n(x,y)} w'_{ij}(x,y)]^2}}}. \tag{2.13}$$

It can be checked that, in cases when f is continuous at (x,y), the value of $|\mathrm{LLK}_n(x,y)|$ would be small, even when f is steep around (x,y), because the slope effect has been accommodated in fitting the local planes in (2.10) and (2.11). See, for instance, Qiu (2004) for a related discussion.

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

$$\mathrm{LL2K}_n(x,y) = \frac{\widehat{f}_{LL2K,+}(x,y) - \widehat{f}_{LL2K,-}(x,y)}{\sqrt{\frac{\sum_{U_n(x,y)} \tilde{w}_{ij}(x,y)^2}{\left[\sum_{U_n(x,y)} \tilde{w}_{ij}(x,y)\right]^2} + \frac{\sum_{V_n(x,y)} \tilde{w}'_{ij}(x,y)^2}{\left[\sum_{V_n(x,y)} \tilde{w}'_{ij}(x,y)\right]^2}}}, \tag{2.14}$$

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:

$$\min_{a_0,a_1,a_2} \sum_{U_n(x,y)} \{Z(x+i/n, y+j/n) - [a_0 + a_1(i/n) + a_2(j/n)]\}^2 K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right),$$

(2.15)

$$\min_{a_0,a_1,a_2} \sum_{V_n(x,y)} \{Z(x+i/n, y+j/n) - [a_0 + a_1(i/n) + a_2(j/n)]\}^2 K\left(\frac{i}{k}, \frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right), \tag{2.16}$$

$$\begin{aligned}
\tilde{w}_{ij}(x,y) &= \left[\tilde{A}_1(x,y) + \tilde{A}_2(x,y)(i/n) + \tilde{A}_3(x,y)(j/n)\right] K\left(\frac{i}{k}, \frac{j}{k}\right) L(d_{ij}/h_n), \\
\tilde{A}_1(x,y) &= \tilde{r}_{20}(x,y)\tilde{r}_{02}(x,y) - \tilde{r}_{11}(x,y)\tilde{r}_{11}(x,y), \\
\tilde{A}_2(x,y) &= \tilde{r}_{01}(x,y)\tilde{r}_{11}(x,y) - \tilde{r}_{10}(x,y)\tilde{r}_{02}(x,y), \\
\tilde{A}_3(x,y) &= \tilde{r}_{10}(x,y)\tilde{r}_{11}(x,y) - \tilde{r}_{01}(x,y)\tilde{r}_{20}(x,y), \\
\tilde{r}_{s_1,s_2}(x,y) &= \sum_{U_n(x,y)} (i/n)^{s_1}(j/n)^{s_2} K\left(\frac{i}{k}, \frac{j}{k}\right) L(d_{ij}/h_n), \text{ for } s_1, s_2 = 0, 1, 2,
\end{aligned}$$

and $\tilde{w}'_{ij}(x,y)$ are defined in the same way as $\tilde{w}_{ij}(x,y)$ except that $U_n(x,y)$ should be replaced by $V_n(x,y)$.

For the edge detection criteria $\mathrm{LLK}_n(x,y)$ and $\mathrm{LL2K}_n(x,y)$, we can derive their asymptotic results similar to those in (2.6) and (2.7). Their corresponding edge 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 $i, j = 1, 2, \cdots, n$, solve the minimization problem (2.3) 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 that is perpendicular to the estimated gradient vector $\{\widehat{b}(x_i, y_j), \widehat{c}(x_i, y_j)\}$. Compute $\widehat{f}_{LLK,+}(x,y)$ and $\widehat{f}_{LLK,-}(x,y)$ (or, $\widehat{f}_{LL2K,+}(x,y)$ and $\widehat{f}_{LL2K,-}(x,y)$) from (2.10) and (2.11) (or, (2.15) and (2.16)).

- The pixel (x_i, y_j) is flagged as a detected edge pixel if the edge detector LLK is used and

$$\mathrm{LLK}_n(x_i, y_j) > Z_{1-\alpha_n}\widehat{\sigma}$$

or, the edge detector LL2K is used and

$$\mathrm{LL2K}_n(x_i, y_j) > Z_{1-\alpha_n}\widehat{\sigma}$$

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

2.2.2 Selection of the bandwidth

The proposed edge detectors described in Subsection 1.2.1 have the bandwidth parameter k involved. In 1-D cases, Gijbels and Goderniaux (2004) has 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 k 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 point set S of the true edge pixels and its estimator $\widehat{S}_n$ by an edge detector with a given bandwidth k. Qiu (2002) investigated this performance measure problem carefully, and suggested the following performance measure:

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

where $0 \le \omega \le 1$ 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 edge 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 0.5.

In mathematics, a popular metric for measuring the distance between two point sets 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 point sets, 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 edge curve is the line $y = 0.5$ and the detected edge pixels 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 edge 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:

$$d_{KQ}(\widehat{S}_n, S; k) = \frac{\omega}{|\widehat{S}_n|}\sum_{(x',y')\in\widehat{S}_n} d_E((x',y'),S) + \frac{1-\omega}{|S|}\sum_{(x,y)\in S} d_E((x,y),\widehat{S}_n), \quad (2.17)$$

where $0 \le w \le 1$ is a weight, and $d_E((x',y'),S)$ denotes the Euclidean distance from the point (x',y') to the point set S. Obviously, d_{KQ} is a weighted average of two averages, where the first average in (2.17) is the average Euclidean distance from the individual detected edge pixels to the set of true edge pixels and the second average is the average Euclidean distance from the individual true edge pixels to the set of detected edge pixels. 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 edge curve 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 edge pixels are shown in Figure 2.1(a) by the dark dots. It can be checked that

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

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 2.1(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 edge pixel at $(0.5, 0.5 + r)$, $d_Q(\widehat{S}_n, S)$ does not depends 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 edge pixel moved when r changes. In this example, it is obvious that $d_H(\widehat{S}_n, S)$ does not reflect the edge 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 edge pixels are those design points on the line at $y = 0.5 + t$ (shown in Figure 2.1(c)), where $0 \leq t \leq 0.5$ is a constant, and the remaining set up is the same as those in the first example. In this example, it can be checked that the three performance measures are

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

When t changes from 0 to 0.5, these performance measures are shown in Figure 2.1(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 edge 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 edge 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 only in practice or numerical study due to the reason that the mathematically well-defined Hausdorff distance is highly sensitive to individual point as demonstrated in Figure 2.1. Thus Hausdorff distance still needs to be adopted for characterizing asymptotic properties of edge detectors in Section 1.3 but d_{KQ} is used in Section 1.4 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, k can be chosen by minimizing $d_{KQ}(\widehat{S}_n, S; k)$. In practice, however, S is often unknown. In such cases, we propose a bootstrap procedure described below for choosing k.

Bootstrap Procedure For Choosing k

1. Apply an edge detector to the original dataset $\{(x_i, y_j, Z_{ij}), i, j = 1, 2, \ldots, n\}$. The set of detected edge pixels is denoted as $\widehat{S}_n$.

2. Construct an estimator $\widehat{f}(x, y)$ of the true intensity function $f(x, y)$ using a jump-preserving surface estimation method, and define the residuals by $\widehat{r}_{ij} = Z_{ij} - \widehat{f}(x_i, y_j)$.

3. Draw with replacement a bootstrap sample of size n^2, denoted as $\{\widehat{r}^*_{ij}, i, j = 1, 2, \cdots, n\}$, from the residual set $\{\widehat{r}_{ij}, i, j = 1, 2, \cdots, n\}$, and then obtain a bootstrap dataset $\{(x_i, y_j, Z^*_{ij}), i, j = 1, 2, \cdots, n\}$, where $Z^*_{ij} = \widehat{f}(x_i, y_j) + \widehat{r}^*_{ij}$.

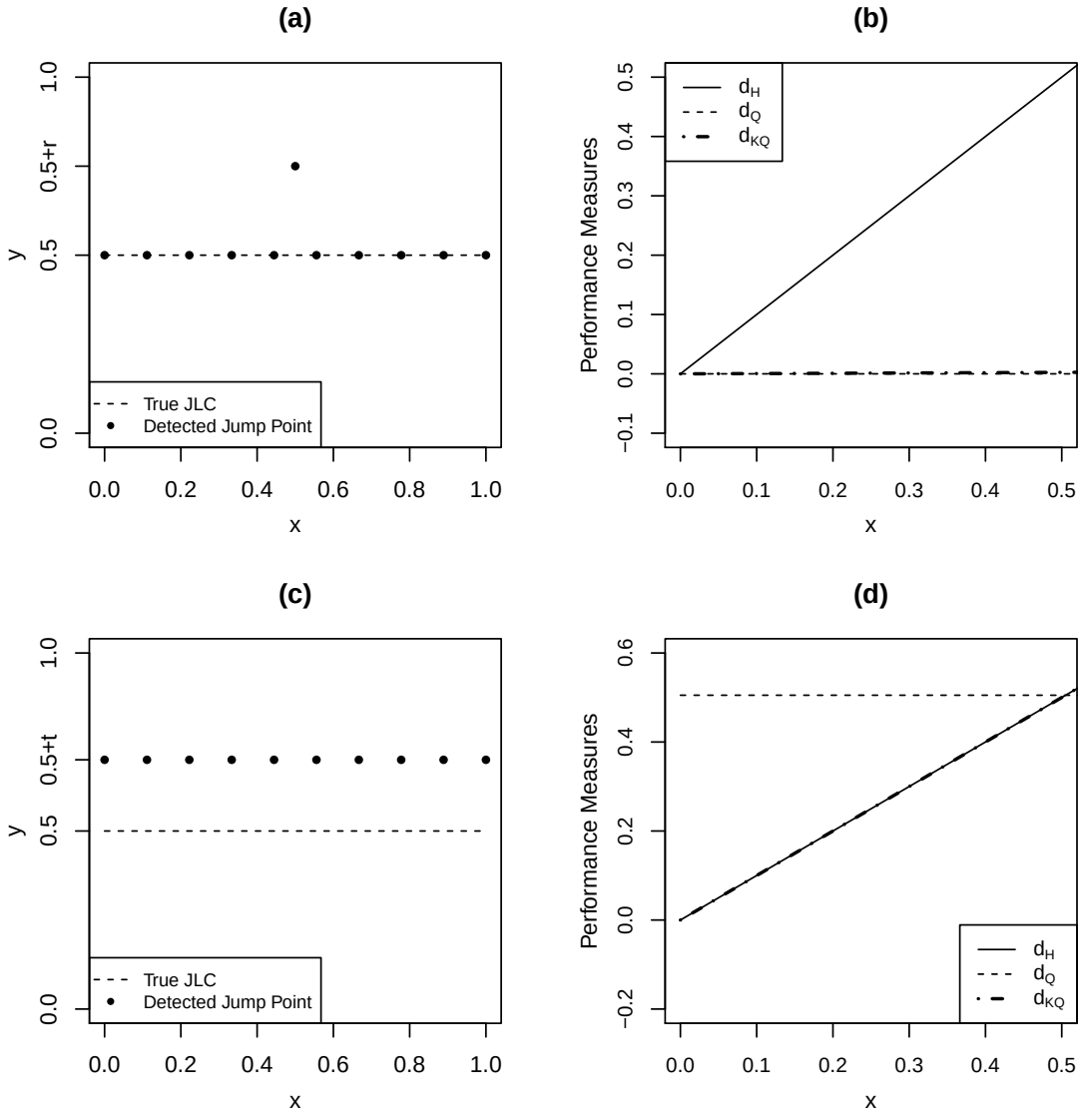

Figure 2.1: (a) Detected edge pixels are shown by the dark dots in the first example. (b) Three metrics for measuring the performance of the detected edge pixels in plot (a) when r changes from 0 to 0.5. (c) Detected edge pixels are shown by the dark dots in the second example. (d) Three metrics for measuring the performance of the detected edge pixels in plot (c) when t changes from 0 to 0.5. In both examples, the true edge curves are the line $y = 0.5$ shown by the dashed lines in plots(a) and (c).

4. The edge detector with the same bandwidth as that in step 1 is then applied to the bootstrap dataset, and a set of detected edge pixels from this bootstrap data set can be obtained.

5. Steps 3 and 4 are repeated for N times. The sets of the detected edge pixels from the N bootstrap datasets are denoted as $\widehat{S}_1^*, \widehat{S}_2^*, \ldots, \widehat{S}_N^*$, respectively. Then, the bootstrap estimator of $d_{KQ}(\widehat{S}_n, S; k)$ is defined by

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

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

In step 2 of the above procedure, $\widehat{f}(x, y)$ 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.

2.3 Statistical Properties

We discuss some statistical properties of the proposed edge detectors. In the literature, the psf $h(u, v; x, y)$ in model (2.2) 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 image. This conventional assumption is also adopted here. Further, we assume h 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) $h(u, v; x, y) \geq 0$,

for all $(u, v) \in \mathbf{R^2}$, (ii) $\int\int_{\mathbf{R^2}} h(u, v; x, y)\ dudv = 1$,(iii) $\int\int_{\mathbf{R^2}} uh(u, v; x, y)\ dudv = \int\int_{\mathbf{R^2}} vh(u, v; x, y)\ dudv = 0$. and (iv) $h(u, v; x, y) = 0$ if $\sqrt{u^2 + v^2} > r_n(x, y)/n$, where $r_n(x, y)$ is a positive integer indicating the number rows (or columns) of pixels that are affected by the blur incurred at (x, y), i.e. $r_n(x, y)$ is the blurring extent at (x, y). Let $R_n = \sup_{x,y} r_n(x, y)$. Then, we have the following result about the proposed edge detectors.

Theorem 2.1

Assume that the true image intensity function f has piece-wisely continuous second-order derivatives in each closed subset of $[0, 1] \times [0, 1]$ where f and its first-order derivatives are continuous; at the boundary curves of the pieces, f has uniformly bounded, directional second-order derivatives from any direction in a single piece; $E(\varepsilon_{11}^3) < \infty$; the kernel function K is a Lipschitz-1 continuous, circularly symmetric, density function; the kernel function L is a Lipschitz-1 continuous increasing density function supported on $[0, 1]$; the psf $h(u, v; x, y)$ is bounded uniformly with respect to (u, v) and (x, y); the blurring extent $R_n = \sup_{i,j} r_n(x_i, y_j)$, the bandwidths k, and sample size n satisfy the conditions that $k/n = o(1)$, $1/k = o(1)$, $R_n/k = o(1)$, and $n^2 \log(n)/k^3 = o(1)$; α_n satisfy the conditions that $k^2/(nZ_{1-\alpha_n}) = o(1)$ and $Z_{1-\alpha_n}/k = o(1)$. Then, we have

$$d_H(\widehat{S}_n \cap \Omega_k \cap J_{S,k}^C, S \cap \Omega_k \cap J_{S,k}^C) = O(k/n),\ a.s.,$$

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

From Theorem 2.1, each of the four proposed edge 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, if the blurring extent R_n is constant, it can be checked that the conditions in Theorem 2.1 are satisfied. Actually the assumptions in Theorem 2.1 allow R_n to increase to infinity when n increases. Namely, when the resolution of the observed image 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 2.1 is given in Section 2.6.

2.4 Numerical Examples

In this section, we present some numerical examples concerning the numerical performance of the four proposed edge detectors. Throughout this section, if there is no further specification, the kernel function used in (2.3) is chosen to be the truncated 2-D Gaussian density function $\frac{1}{2\pi-3\pi\exp(-0.5)}[\exp(-(x^2+y^2)/2-\exp(-0.5))]I_{x^2+y^2\le 1}$, the 2-D kernel function used in constructing each edge detection criterion (cf., e.g., (2.4)) is chosen to be the Epanechnikov kernel function $\frac{2}{\pi}(1-x^2-y^2)I_{x^2+y^2\le 1}$, and the 1-D kernel function L used in (2.8) and (2.14) is chosen to be $\frac{1}{1.194958}\exp(x^2/2)I_{0\le x\le 1}$.

In the first example, the following true image intensity function is considered:

$$f_1(x,y)=\begin{cases} 3, & \text{if } (x-0.5)^2+(y-0.5)^2\le .04\\ 2, & \text{if } .04<(x-0.5)^2+(y-0.5)^2\le .09\\ -20(\sqrt{(x-0.5)^2+(y-0.5)^2}-0.4), & \text{if } .09<(x-0.5)^2+(y-0.5)^2\le .16\\ 0, & \text{otherwise.}\end{cases}$$

It is shown in Figure 2.2, from which it can be seen that f_1 has one circular edge curve, and its surface is steep in the region $\{(x,y): 0.09<(x-0.5)^2+(y-0.5)^2<$

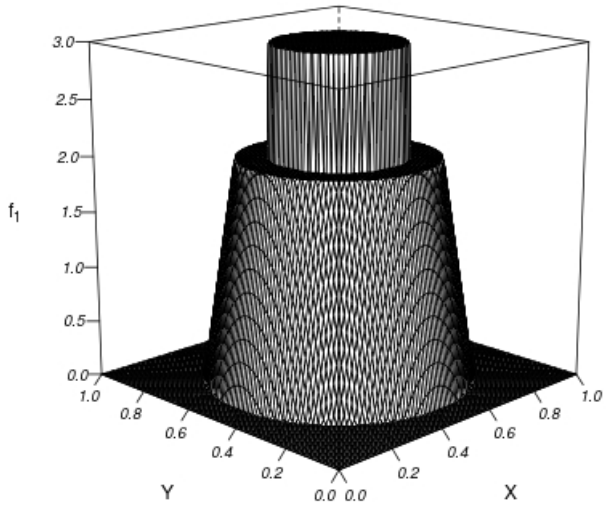

Figure 2.2: The true image intensity function $f_1(x, y)$.

0.16}. In model (2.2), the psf is chosen to be

$$h(u, v; x, y) = \frac{3}{\pi(r_n(x, y)/n)^2} \left(1 - \frac{\sqrt{u^2 + v^2}}{r_n(x, y)/n}\right) I_{u^2+v^2 \le (r_n(x,y)/n)^2}, \tag{2.18}$$

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 $r_n(x, y)$. Let $\rho_n(x, y) := r_n(x, y)/n$ denote the *blurring-extent-sample-size-ratio* (BSR) at (x, y). In this example, we simply choose $\rho_n(x, y) \equiv 0.02$. In such cases, $h(u, v; x, y)$ is location invariant. Next, we try to measure the performance of the related edge detectors quantitatively using the proposed performance measure $d_{KQ}(\widehat{S}_n, S; k)$, in which ω is fixed at 0.5 in this section. We consider two sample sizes $n = 100$ and 200, and three σ 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 $n = 200$. Simulation results based on 100 replications

are presented in Table 2.1. In the simulation, for each combination of n and σ and for each edge detector, the optimal bandwidth k is chosen by minimizing the averaged value of the 100 values of $d_{KQ}(\widehat{S}_n, S; k)$ that are obtained from 100 replicated simulations. Bandwidths chosen by our proposed bootstrap procedure are also presented in the table.

Table 2.1: This table presents some numerical results about example 1 based on 100 replicated simulations. In each entry, the first number in parenthesis is k/n when k is the optimal bandwidth, the second number in parenthesis is k/n when k is chosen by the proposed bootstrap procedure, and the number in the second line is the averaged value of $d_{KQ}(\widehat{S}_n, S; k)$ when k 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) 0.00885	(0.09, 0.13) 0.0125	(0.10, 0.14) 0.0297
	LL2K	(0.06, 0.10) 0.0112	(0.07, 0.11) 0.0127	(0.08, 0.12) 0.0218
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) 0.00444	(0.06, 0.06) 0.00549	(0.07, 0.07) 0.00609
	LL2K	(0.04, 0.04) 0.00628	(0.05, 0.05) 0.00637	(0.06, 0.06) 0.00784

From Table 2.1, it can be seen that: (1) the edge 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) in this example, (2) generally speaking, bandwidth for each edge detector should be chosen larger for noisier data, (3) performance of the edge detectors based on the local constant kernel

smoothing does not change much when noise level or sample size changes, (4) the edge detectors based on the local linear kernel smoothing perform better when the sample size gets larger, (5) the bootstrap procedure tends to select the bandwidth slightly larger than the optimal one, and (6) the bootstrap procedure gives close-to-optimal bandwidths when the sample size increases. The result (1) is intuitively reasonable because the intensity function is steep in certain regions in this example (cf., Figure 2.2), the edge detectors based on the local linear kernel smoothing can accommodate such a case well, but the edge detectors based on the local constant kernel smoothing would have many false detections due to the steep intensity surface. The result (3) is also caused by the false detections of the edge detectors based on the local constant kernel smoothing.

In cases when $n = 100$ and $\sigma = 0.3$, the detected edges by the four edge detectors with their optimal bandwidths presented in Table 2.1 are shown in Figure 2.3, where the results are from the simulation whose $d_{KQ}(\widehat{S}_n, S; k)$ value is the median of the 100 $d_{KQ}(\widehat{S}_n, S; k)$ values computed from the 100 replications. From the plots in this figure, it can be seen that (1) the detected edge pixels in both plots (a) and (b) contain some false detections in the region where the true intensity surface is steep, (2) both the edge detectors LCK and LC2K fail to detect some true edge pixels, and (3) the detected edge pixels in plot (d) seem to be a little more variable than those in plot (c). All these results are consistent with those in Table 2.1 and with our theoretical justifications discussed before. This example demonstrates that the edge detectors LLK and LL2K can accommodate the slope effect of the true image well, while the edge detectors LCK and LC2K can not.

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

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

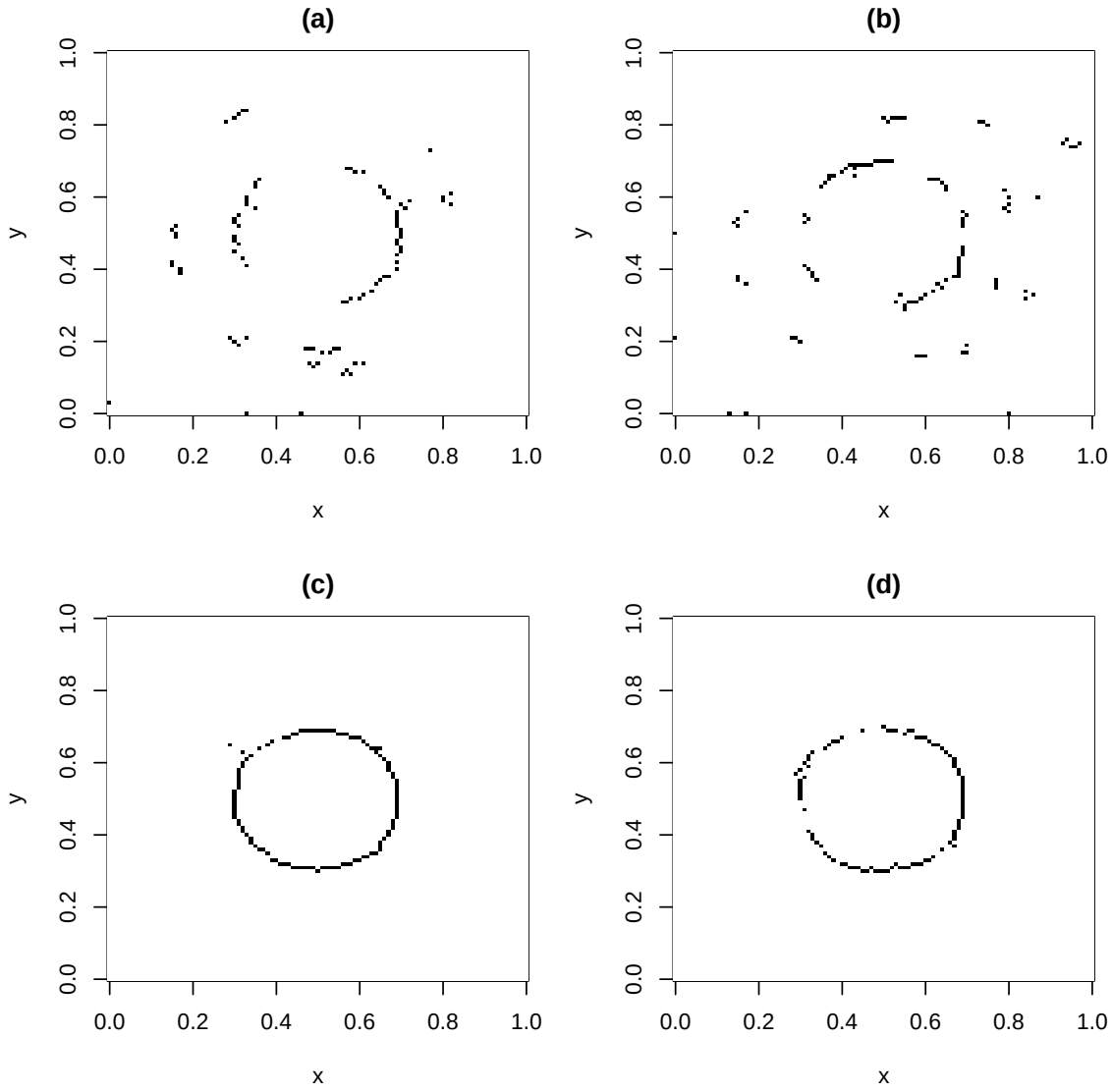

Figure 2.3: (a)-(d): Detected edge pixels by edge detector LCK, LC2K, LLK, and LL2K, respectively.

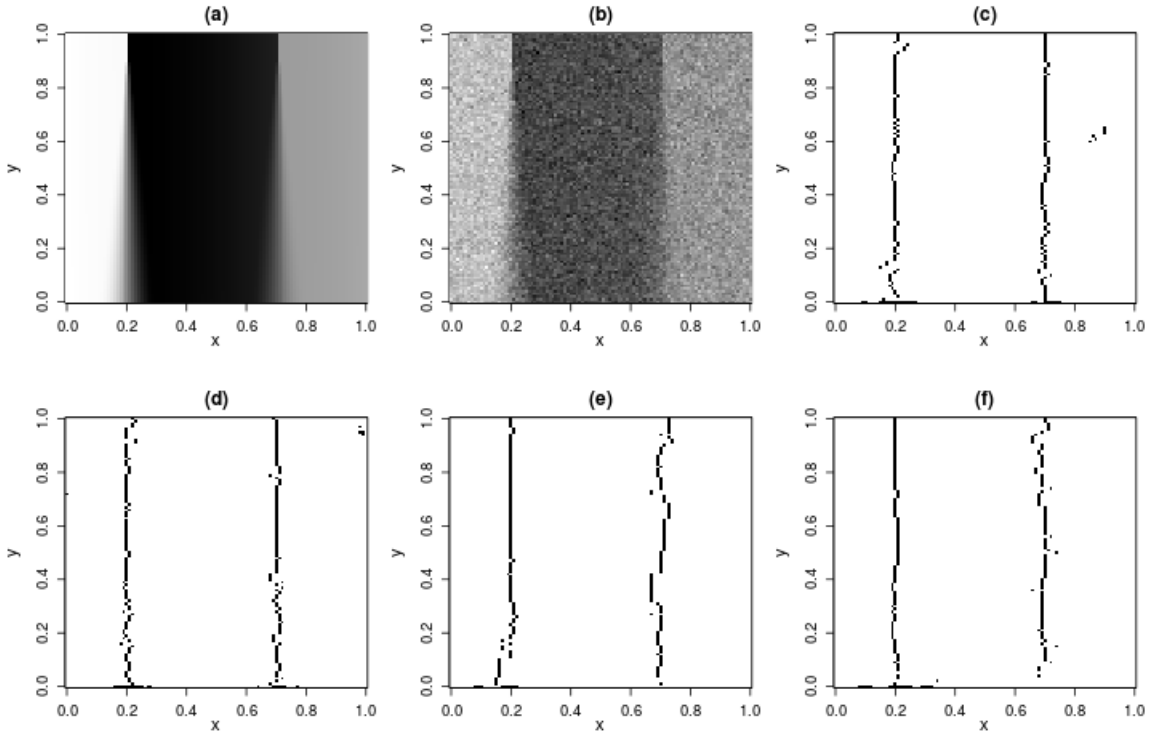

Figure 2.4: (a): The blurred version of f_2; (b): A noisy and blurred version of f_2 with $n = 100$ and $\sigma = 0.3$; (c)-(f): Detected edges by edge detector LCK, LC2K, LLK and LL2K, respectively

and the psf is defined by (2.18) with $\rho_n(x, y) = 0.1(1 - y)$. Therefore, the true intensity function f_2 has two edge curves, both are parallel to the y-axis at $x = 0.2$ and $x = 0.7$, and the jump size is larger (to be 2) at $x = 0.2$ and smaller (to be 1) at $x = 0.7$. The psf is location variant in this example, with blurring extent increasing when y gets smaller. The blurred version of f_2 is shown in Figure 2.4(a). A noisy and blurred version of f_2 with $n = 100$ and $\sigma = 0.3$ is shown in Figure 2.4(b).

Then, in the same setup as that in Table 1, the optimal bandwidths, the corresponding $d_{KQ}(\widehat{S}_n, S; k)$ 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 (1) the four edge detectors all perform better when the sam-

ple size increases, (2) the edge detectors LLK and LL2K are more sensitive to the noise level in the sense that their $d_{KQ}(\widehat{S}_n, S; k)$ values increase more rapidly as the noise level increases, compared to the edge detectors LCK and LC2K, (3) for all four edge detectors, a larger bandwidth is chosen when the noise level is larger, and (4) the bandwidths chosen by the bootstrap procedure are all quite close to the optimal bandwidths. The result (2) 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 (1) and (3) are intuitively reasonable.

Table 2.2: This table presents some numerical results about example 2 based on 100 replicated simulations. In each entry, the first number in parenthesis is k/n when k is the optimal bandwidth, the second number in parenthesis is k/n when k is chosen by the proposed bootstrap procedure, and the number in the second line is the averaged value of $d_{KQ}(\widehat{S}_n, S; k)$ when k is chosen to be the optimal bandwidth.

n	Method	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
100	LCK	(0.03, 0.05) 0.0163	(0.05, 0.06) 0.0175	(0.05, 0.06) 0.0184
	LC2K	(0.03, 0.05) 0.0171	(0.05, 0.06) 0.0185	(0.05, 0.07) 0.0193
	LLK	(0.14, 0.13) 0.0149	(0.15, 0.15) 0.0197	(0.16, 0.16) 0.0232
	LL2K	(0.12, 0.14) 0.0155	(0.14, 0.14) 0.0193	(0.15, 0.15) 0.0238
200	LCK	(0.03, 0.04) 0.0125	(0.03, 0.05) 0.0138	(0.04, 0.05) 0.0152
	LC2K	(0.03, 0.04) 0.0138	(0.03, 0.05) 0.0146	(0.04, 0.05) 0.0159
	LLK	(0.11, 0.13) 0.00977	(0.13, 0.14) 0.0111	(0.14, 0.15) 0.0137
	LL2K	(0.10, 0.11) 0.0107	(0.11, 0.11) 0.0124	(0.12, 0.12) 0.0140

Figure 2.4(c)-(f) show the detected edges by the four edge detectors in the case

when $n = 100$, $\sigma = 0.3$, and their optimal bandwidths presented in Table 2.2 are used, where the results are from the simulation whose $d_{KQ}(\widehat{S}_n, S; k)$ value is the median of the 100 $d_{KQ}(\widehat{S}_n, S; h_n)$ values computed from the 100 replications. From the plots, it can be seen that (1) all four edge detectors detect the edge segment of larger jump magnitude at $x = 0.2$ reasonably well, (2) the edge detectors LCK and LC2K that are based on the local constant kernel smoothing detectors perform better than the edge detectors LLK and LL2K that are based on the local linear kernel smoothing when detecting the edge segment of smaller jump magnitude at $x = 0.7$, (3) by comparing plots (c) and (d), it seems that the edge detector LC2K is slightly more robust to the blurring than the edge detector LCK, and (4) by comparing plots (e) and (f), the edge detector LL2K seems more robust to the blurring than the edge detector LLK. This example demonstrates that (i) the edge detectors LC2K and LL2K using 2 kernel functions in their construction seem more robust to the blurring, compared to the other two edge detectors, and (ii) the results by the edge detectors LCK and LC2K seem less variable, compared to the edge detectors LLK and LL2K.

At the end of this section, we consider a real test image shown in Figure 2.5(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 (2.18) with $\rho_n(x, y) \equiv 0.02$ and by the i.i.d. noise from $N(0, 5^2)$ is shown in Figure 2.5(b). The detected edges by the four edge detectors LCK, LC2K, LLK, and LL2K are presented in Figure 2.5(c)-(f), respectively. In all four edge detectors, α_n is fixed at 0.001, and their bandwidths (i.e., values of k in (2.3)) are chosen by the bootstrap procedure to be 3, 3, 11 and 9. From the plots, it can be seen that (1) all four edge detectors detect the major edge curves reasonably well, (2) the edge curves detected by LCK and LC2K are thinner and more straight than those detected by LLK and LL2K; but, some false edges are detected by them here and there, (3) the edge detector LLK fails to detect certain edge curves (e.g.,

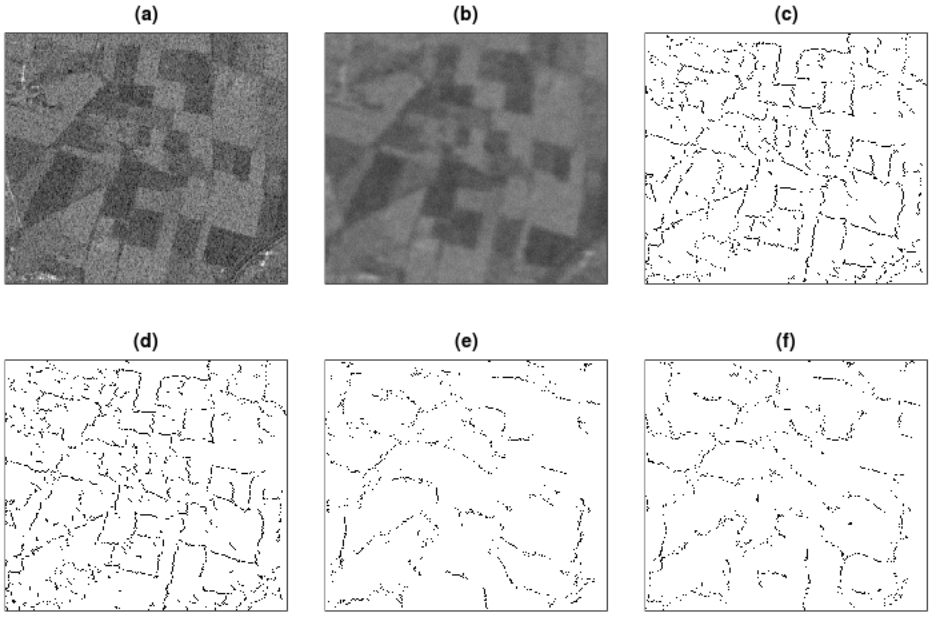

Figure 2.5: (a): Original SAR image; (b): A noisy and blurred version of theSAR image; (c)-(f): Detected edges by LCK, LC2K, LLK and LL2K, respectively.

the edges located in the upper-right portion of the image), and (4) the edge detector LL2K detects more true edge pixels than the edge detector LLK, which is consistent with our analysis in section 1.2.

2.5 Discussions

We have presented four edge detectors for detecting step edges in blurred noisy images. All four edge 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 an edge 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.

Numerical examples presented in the previous section show that these edge detectors have their own strengths and limitations when handling various different situations. These results can be summarized as follows. (1) The edge detectors based on local constant kernel smoothing (i.e., LCK and LC2K) are more robust to noise and their detected edge curves tend to be thinner. However, they cannot accommodate the slope effect of the true intensity surface well and would detect many false edges at places where the true intensity surface is steep. (2) The edge detectors based on local linear kernel smoothing (i.e., LLK and LL2K) can accommodate the slope effect of the true intensity surface well; but they are more sensitive to noise. (3) The edge detectors using 2 kernel functions (i.e., LC2K and LL2K) are more robust to blurring; however, their detected edges tend to have a larger variability. Considerable future research will be required to find an appropriate way to combine the strengths of the four edge detectors and avoid their major limitations at the same time.

2.6 Technical Details

For convenience of mathematical presentation, let $h_n = k/n$. Then the asymptotic conditions on k in Theorem 2.1 become $h_n = o(1)$, $1/(nh_n) = o(1)$, $nR_n/h_n = o(1)$, and $\log(n)/(nh_n^3) = o(1)$; α_n satisfy the conditions that $nh_n^2/Z_{1-\alpha_n} = o(1)$ and $Z_{1-\alpha_n}/(nh_n) = o(1)$.

Lemma 2.1

Let $\phi(\cdot,\cdot)$ be any continuous function, $K(\cdot,\cdot)$ be a Lipschitz-1 continuous bivariate density kernel function with support $\{(u,v) : u^2 + v^2 \le 1\}$, and ε_{ij} be i.i.d. random errors from model (2.2) with mean 0 and variance σ^2. Then, if the bandwidth h_n used

in procedure (2.3) satisfies the condition that $h_n = o(1)$ *and* $1/(nh_n) = o(1)$*, we have*

$$\frac{1}{nh_n} \sum_{(x_i,y_j)\in O_n(x,y)} \varepsilon_{ij}\phi\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) \xrightarrow{d} N\left(0, \widetilde{\sigma}^2\right), \text{ as } n \to \infty,$$

where $\widetilde{\sigma}^2 = \sigma^2 \int_{u^2+v^2\le 1} \phi^2(u,v)K^2(u,v)\, dudv$ *and* (x_i, y_j)*,* $O_n(x,y)$ *are defined to be the same as those in (2.3).* □

Remark 2.1

A direct conclusion of Lemma 2.1 is that

$$\frac{1}{n^2h_n^2} \sum_{(x_i,y_j)\in O_n(x,y)} \varepsilon_{ij}\phi\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) = o\left(\frac{\log(n)}{nh_n}\right) \ a.s.$$ □

Proof 2.1 (Lemma 2.1)

This is a simple application of Lindeberg-Feller conditions. In fact, the terms in the summation are all independent and have the mean 0. Also, we observe that

$$\begin{aligned} \sum_{(x_i,y_j)\in O_n(x,y)} & \mathrm{E}\{\varepsilon_{ij}^2\}\phi\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)^2 K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)^2 \frac{1}{n^2h_n^2} \\ \to \quad & \sigma^2 \int_{u^2+v^2\le 1} \phi^2(u,v)K^2(u,v)\, dudv, \text{ as } n \to \infty. \end{aligned}$$

Next, for any $\delta > 0$, we have

$$\begin{aligned} \sum_{(x_i,y_j)\in O_n(x,y)} & \phi\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)^2 K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)^2 \frac{1}{n^2h_n^2} \cdot \\ & \mathrm{E}\left\{\varepsilon_{ij}^2 I_{\left\{\frac{1}{nh_n}\left|\phi\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)\varepsilon_{ij}\right|>\delta\right\}}\right\} \\ \le \quad & \frac{C}{n^2h_n^2} \sum_{(x_i,y_j)\in O_n(x,y)} \mathrm{E}\left\{\varepsilon_{ij}^2 I_{\left\{|\varepsilon_{ij}|>\frac{\delta nh_n}{C}\right\}}\right\} \\ \le \quad & \frac{C}{n^2h_n^2}\frac{h_n^2}{1/n^2}\mathrm{E}\left\{\varepsilon_{11}^2 I_{\left\{|\varepsilon_{11}|>\frac{\delta nh_n}{C}\right\}}\right\} \\ \to \quad & 0, \qquad \text{as } n \to \infty, \end{aligned} \tag{2.19}$$

where C is some constant. Thus, all the Lindeberg-Feller conditions are satisfied, and the desired result follows immediately. □

Lemma 2.2

Under the condition of Theorem 2.1, the estimated gradient $(\widehat{b}(x,y), \widehat{c}(x,y))$ *obtained from local linear kernel smoothing procedure (2.3) has the following properties:*

(i) If (x,y) *is not on any step edge segment, then*

$$(\widehat{b}(x,y), \widehat{c}(x,y)) \to (f'_x(x,y), f'_y(x,y)), \ \ a.s, \qquad as\ n \to \infty. \tag{2.20}$$

(ii) If (x,y) *is a non-singular point on a step edge segment and the the step edge segment has a unique tangent line at* (x,y)*, then*

$$\frac{(\widehat{b}(x,y), \widehat{c}(x,y))}{\sqrt{\widehat{b}(x,y)^2 + \widehat{c}(x,y)^2}} \to (-\sin\theta, \cos\theta), \ \ a.s, \ , \qquad as\ n \to \infty, \tag{2.21}$$

where θ *is the angle formed by the tangent line of the step edge segment at* (x,y) *and the x-axis.*

(iii) If (x,y) *is a non-singular point on a step edge segment and the step edge segment has two one-sided tangent lines at* (x,y)*, then*

$$\frac{(\widehat{b}(x,y), \widehat{c}(x,y))}{\sqrt{\widehat{b}(x,y)^2 + \widehat{c}(x,y)^2}} \to \left(\cos\left(\frac{\theta_1+\theta_2}{2}\right), \sin\left(\frac{\theta_1+\theta_2}{2}\right)\right), \ \ a.s, \ , \qquad as\ n \to \infty, \tag{2.22}$$

where θ_1 *and* θ_2 *are angles formed by the two one-sided tangent lines and the x-axis respectively.* □

Proof 2.2 (Lemma 2.2)

First, it is not difficult to verify that the solution of procedure (2.3) has the expressions

$$\widehat{b}(x,y) = \frac{1}{r_{20}} \sum_{(x_i,y_j)\in O_n(x,y)} (x_i - x) Z_{ij} K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right), \tag{2.23}$$

$$\widehat{c}(x,y) = \frac{1}{r_{02}} \sum_{(x_i,y_j)\in O_n(x,y)} (y_j - y) Z_{ij} K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right), \tag{2.24}$$

where $r_{s_1 s_2} = \sum_{(x_i,y_j)\in O_n(x,y)} (x_i - x)^{s_1} (y_j - y)^{s_2} K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)$, for $s_1, s_2 = 0, 1, 2$.

To prove result (2.20), we notice that, for a given point (x, y), if (x, y) is not on any step edge segment, then

$$\mathrm{E}(\widehat{b}(x,y)) = \frac{1}{r_{20}} \sum_{(x_i,y_j)\in O_n(x,y)} H\{f\}(x_i, y_j)(x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right), \tag{2.25}$$

where

$$\begin{aligned} H\{f\}(x_i, y_j) &= \int\int_{u^2+v^2\le\rho_n^2} h(u, v; x_i, y_j) f(x_i - u, y_j - v)\, du dv \\ &= \int\int_{u^2+v^2\le\rho_n^2} h(u, v; x_i, y_j) \left[f(x_i, y_j) - f'_x(x_i, y_j)u \right. \\ &\quad \left. - f'_y(x_i, y_j)v + O(\rho_n^2)\right]\, du dv \\ &= f(x_i, y_j) + O(\rho_n^2). \end{aligned} \tag{2.26}$$

In the last equation of (3.26), we have used the symmetry of h. By (3.25) and (3.26), we have

$$\begin{aligned} &\mathrm{E}(\widehat{b}(x,y)) \\ = \ & \frac{1}{r_{20}} \sum_{(x_i,y_j)\in O_n(x,y)} [f(x_i, y_j) + O(\rho_n^2)](x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{r_{20}} \sum_{(x_i,y_j)\in O_n(x,y)} \Big[f(x,y) + f'_x(x,y)(x_i - x) + f'_y(x,y)(y_j - y) + \\
&\quad \frac{1}{2} f''_{xx}(x,y)(x_i - x)^2 + f''_{xy}(x,y)(x_i - x)(y_j - y) + \frac{1}{2} f''_{yy}(x,y)(y_j - y)^2 + O(h_n^3) \Big] \\
&\quad (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n} \right) \\
&\quad + \frac{1}{r_{20}} \sum_{(x_i,y_j)\in O_n(x,y)} O(\rho_n^2)(x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n} \right) \\
&= f'_x(x,y) + O(\rho_n^2/h_n) + O(h_n^2). \qquad (2.27)
\end{aligned}
$$

In the last equation of the above expression, we have used the results that $r_{s_1,s_2} = 0$, for $s_1, s_2 = 0, 1, 2$ with $s_1 + s_2$ being odd, using the circular symmetry of K, the equal spacing of the design points, and the properties that $r_{20} = O(n^2 h_n^4)$, which can be proved similarly to expression (23) in Proposition 2 of Qiu (2009). Then, by Lemma Lemma 2.1, we have

$$
\frac{1}{n^2 h_n^2} \sum_{(x_i,y_j)\in O_n(x,y)} \varepsilon_{ij} \phi\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n} \right) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n} \right) = o\left(\frac{\log n}{n h_n} \right), \ a.s., \quad (2.28)
$$

where $\phi(u, v)$ is any continuous function defined in the region $\{(u, v) : u^2 + v^2 \leq 1\}$. By (3.25) and the fact that $r_{20} = O(n^2 h_n^4)$, we have

$$
\begin{aligned}
&\widehat{b}(x,y) - \mathrm{E}(\widehat{b}(x,y)) \\
&= \frac{1}{r_{20}} \sum_{(x_i,y_j)\in O_n(x,y)} \varepsilon_{ij}(x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n} \right) \\
&= o\left(\frac{\log(n)}{n h_n^2} \right), \ a.s. \qquad (2.29)
\end{aligned}
$$

Similarly,

$$
\widehat{c}(x,y) - \mathrm{E}(\widehat{c}(x,y))
$$

$$= \frac{1}{r_{02}} \sum_{(x_i,y_j)\in O_n(x,y)} \varepsilon_{ij}(y_j - y) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right)$$
$$= o\left(\frac{\log(n)}{nh_n^2}\right), \ a.s. \tag{2.30}$$

(2.20) is then obtained, after combining (3.29) and (3.30).

To prove (2.21), assume that (x, y) is a non-singular point on a step edge segment. Then, $O_n(x, y)$ consists of the following three disjoint parts $O_{n,l}(x, y)$, $O_{n,c}(x, y)$, and $O_{n,r}(x, y)$, where $O_{n,c}(x, y)$ is a band of width $2\rho_n$ containing the step edge segment, and $O_{n,l}(x, y)$ and $O_{n,r}(x, y)$ are two neighborhoods on its different sides. Since the step edge has a unique tangent line at (x, y), difference between the curve and the tangent line will be negligible. Thus, we may assume that the step edge segment is a straight line in $O_n(x, y)$ and it forms an angle, denoted by θ, with the x-axis. Then,

$$\begin{aligned}
&\mathrm{E}(\widehat{b}(x,y)) \\
= \ & \frac{1}{r_{20}} \left(\sum_{O_{n,l}(x,y)} + \sum_{O_{n,c}(x,y)} + \sum_{O_{n,r}(x,y)} \right) H\{f\}(x_i, y_j)(x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) \\
= \ & \frac{1}{r_{20}} \sum_{O_{n,l}(x,y)} [f(x_i, y_j) + O(\rho_n^2)](x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + \\
& \frac{1}{r_{20}} \sum_{O_{n,c}(x,y)} H\{f\}(x_i, y_j)(x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + \\
& \frac{1}{r_{20}} \sum_{O_{n,r}(x,y)} [f(x_i, y_j) + O(\rho_n^2)](x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) \\
= \ & \frac{1}{r_{20}} \sum_{O_{n,l}(x,y)} [f_-(x, y) + O(h_n) + O(\rho_n^2)](x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + \\
& O\left(\frac{\rho_n}{h_n^2}\right) + \frac{1}{r_{20}} \sum_{O_{n,r}(x,y)} [f_+(x, y) + O(h_n) + O(\rho_n^2)](x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) \\
= \ & \frac{1}{r_{20}} f_-(x, y) \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) -
\end{aligned}$$

$$
\begin{aligned}
= & \frac{1}{r_{20}} f_-(x,y) \sum_{O_{n,r}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + O\left(\frac{\rho_n^2}{h_n}\right) - \\
& \frac{1}{r_{20}} f_-(x,y) \sum_{O_{n,c}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + \\
& \frac{1}{r_{20}} f_+(x,y) \sum_{O_{n,r}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + O(1) + O\left(\frac{\rho_n}{h_n^2}\right) \\
= & \frac{f_+(x,y) - f_-(x,y)}{r_{20}} \sum_{O_{n,r}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + \\
& O(1) + O\left(\frac{\rho_n}{h_n^2}\right). \qquad (2.31)
\end{aligned}
$$

In the second equation of (3.33), (3.26) is used. In the third equation , we have used the results that $r_{20} = O(n^2 h_n^4)$, $H\{f\}(x_i, y_j)$ are uniformly bounded when $(x_i, y_j) \in O_{n,c}(x,y)$, and the fact that the ratio of the area of $O_{n,c}(x,y)$ to the area of $O_n(x,y)$ is of order $O(\rho_n/h_n)$. In the fourth equation, we have used the results that $\sum_{O_{n,r}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) = O(n^2 h_n^3)$, $\sum_{O_{n,l}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) = O(n^2 h_n^3)$, $r_{20} = O(n^2 h_n^4)$. In the last equation, we have used the results that $r_{10} = 0$ and $\frac{1}{r_{20}} \sum_{O_{n,c}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) = O\left(\frac{\rho_n}{h_n^2}\right)$. By (3.29), we have

$$
\begin{aligned}
\widehat{b}(x,y) = & \frac{f_+(x,y) - f_-(x,y)}{r_{20}} \sum_{O_{n,r}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + \\
& O(1) + O\left(\frac{\rho_n}{h_n^2}\right) + o\left(\frac{\log(n)}{n h_n^2}\right), \ a.s. \qquad (2.32)
\end{aligned}
$$

Similarly, we can check that

$$
\begin{aligned}
\widehat{c}(x,y) = & \frac{f_+(x,y) - f_-(x,y)}{r_{02}} \sum_{O_{n,r}(x,y)} (y_j - y) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + \\
& O(1) + O\left(\frac{\rho_n}{h_n^2}\right) + o\left(\frac{\log(n)}{n h_n^2}\right), \ a.s. \qquad (2.33)
\end{aligned}
$$

Notice the following two facts:

$$\begin{aligned}\frac{h_n}{r_{20}}\sum_{O_{n,r}(x,y)}(x_i-x)K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right) &\to \frac{\int_\theta^{\theta+\pi}\,d\varphi\int_0^1 r^2\cos\varphi K(r)\,dr}{\int_0^{2\pi}\,d\varphi\int_0^1 r^3\cos^2\varphi K(r)\,dr}\\ &= \frac{-2\int_0^1 r^2K(r)\,dr}{\pi\int_0^1 r^3K(r)\,dr}\sin\theta. \qquad (2.34)\end{aligned}$$

$$\begin{aligned}\frac{h_n}{r_{02}}\sum_{O_{n,r}(x,y)}(y_j-y)K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right) &\to \frac{\int_\theta^{\theta+\pi}\,d\varphi\int_0^1 r^2\sin\varphi K(r)\,dr}{\int_0^{2\pi}\,d\varphi\int_0^1 r^3\sin^2\varphi K(r)\,dr}\\ &= \frac{2\int_0^1 r^2K(r)\,dr}{\pi\int_0^1 r^3K(r)\,dr}\cos\theta. \qquad (2.35)\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{(\widehat{b}(x,y),\widehat{c}(x,y))}{\sqrt{\widehat{b}(x,y)^2+\widehat{c}(x,y)^2}} &= \frac{(h_n\widehat{b}(x,y),h_n\widehat{c}(x,y))}{\sqrt{h_n^2\widehat{b}(x,y)^2+h_n^2\widehat{c}(x,y)^2}}\\ &\to (-\sin\theta,\cos\theta),\ a.s,\end{aligned}$$

which completes the proof of (2.21).

Next, assume that (x,y) is a non-singular point on a step edge, and there exist two one-sided tangent lines of the step edge at (x,y), forming angles θ_1 and θ_2, respectively, with the x-axis. See Figure Figure 2.6 for a demonstration. The difference between the polygonal line and the step edge segment in $O_n(x,y)$ is negligible when n is sufficiently large. Hence, we may assume that the step edge segment is the same as the polygonal line in $O_n(x,y)$ without loss of generality. By the same arguments in (3.33) and (3.34), we can show that

$$\widehat{b}(x,y) = \frac{f_+(x,y)-f_-(x,y)}{r_{20}}\sum_{O_{n,r}(x,y)}(x_i-x)K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)+$$

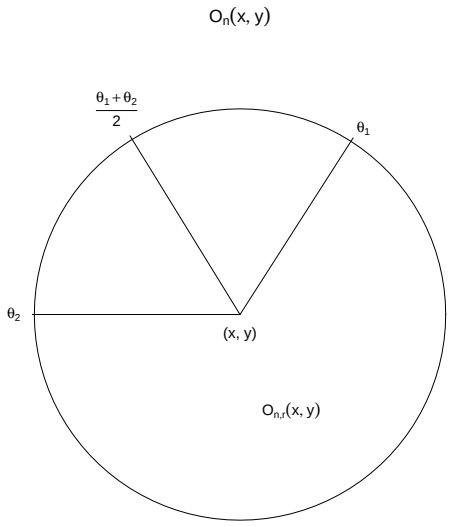

Figure 2.6: A demonstration for the case when (x, y) is on a step edgethat has two one-sided tangent lines at (x, y).

$$O(1) + O\left(\frac{\rho_n}{h_n^2}\right) + o\left(\frac{\log(n)}{nh_n^2}\right), \ a.s. \tag{2.36}$$

$$\widehat{c}(x,y) = \frac{f_+(x,y) - f_-(x,y)}{r_{02}} \sum_{O_{n,r}(x,y)} (y_j - y) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) + O(1) + O\left(\frac{\rho_n}{h_n^2}\right) + o\left(\frac{\log(n)}{nh_n^2}\right), \ a.s. \tag{2.37}$$

Also, we observe the following facts:

$$\begin{aligned} & \frac{h_n}{r_{20}} \sum_{O_{n,r}(x,y)} (x_i - x) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) \\ \to \ & \frac{\int_{\theta_2}^{\theta_1 + 2\pi} d\varphi \int_0^1 r^2 \cos\varphi K(r)\, dr}{\int_0^{2\pi} d\varphi \int_0^1 r^3 \cos^2\varphi K(r)\, dr} = \frac{\int_0^1 r^2 K(r)\, dr}{\pi \int_0^1 r^3 K(r)\, dr} (\sin\theta_1 - \sin\theta_2) \\ = \ & \frac{2\int_0^1 r^2 K(r)\, dr}{\pi \int_0^1 r^3 K(r)\, dr} \sin\left(\frac{\theta_1 - \theta_2}{2}\right) \cos\left(\frac{\theta_1 + \theta_2}{2}\right). \end{aligned} \tag{2.38}$$

$$\begin{aligned}
& \frac{h_n}{r_{02}} \sum_{O_{n,r}(x,y)} (y_j - y) K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) \\
\to \quad & \frac{\int_{\theta_2}^{\theta_1+2\pi} d\varphi \int_0^1 r^2 \sin\varphi K(r)\, dr}{\int_0^{2\pi} d\varphi \int_0^1 r^3 \sin^2\varphi K(r)\, dr} = \frac{\int_0^1 r^2 K(r)\, dr}{\pi \int_0^1 r^3 K(r)\, dr} (\cos\theta_2 - \cos\theta_1) \\
= \quad & \frac{2\int_0^1 r^2 K(r)\, dr}{\pi \int_0^1 r^3 K(r)\, dr} \sin\left(\frac{\theta_1 - \theta_2}{2}\right) \sin\left(\frac{\theta_1 + \theta_2}{2}\right). \qquad (2.39)
\end{aligned}$$

Therefore, it follows after combining (2.36)– (2.39) that

$$\begin{aligned}
\frac{(\widehat{b}(x,y), \widehat{c}(x,y))}{\sqrt{\widehat{b}(x,y)^2 + \widehat{c}(x,y)^2}} &= \frac{(h_n\widehat{b}(x,y), h_n\widehat{c}(x,y))}{\sqrt{h_n^2\widehat{b}(x,y)^2 + h_n^2\widehat{c}(x,y)^2}} \\
&\to \left(\cos\left(\frac{\theta_1 + \theta_2}{2}\right), \sin\left(\frac{\theta_1 + \theta_2}{2}\right)\right), \ a.s,
\end{aligned}$$

which finishes the proof of (2.22). □

Proof 2.3 (Theorem 2.1)

Let us first prove the theorem for the edge detector LL2K. Let $\widehat{S}_n$ be the set of detected edge pixels by the edge detector LL2K. For any $(x,y) \in \Omega_{h_n}$, we have

$$\begin{aligned}
\widehat{f}_{LL2K,+}(x,y) &= \frac{\sum_{(x_i,y_j)\in U_n(x,y)} \widetilde{w}_{ij}(x,y) Z_{ij}}{\sum_{(x_i,y_j)\in U_n(x,y)} \widetilde{w}_{ij}(x,y)} \\
&= \frac{\sum_{U_n} H\{f\}(x_i,y_j)\widetilde{w}_{ij}(x,y)}{\sum_{U_n} \widetilde{w}_{ij}(x,y)} + \frac{\sum_{U_n} \varepsilon_{ij}\widetilde{w}_{ij}(x,y)}{\sum_{U_n} \widetilde{w}_{ij}(x,y)} \\
&=: I_1(x,y) + I_2(x,y), \qquad (2.40)
\end{aligned}$$

where $\sum_{U_n}$ denotes $\sum_{(x_i,y_j)\in U_n}$, U_n is the upper half of $O_n(x,y)$ divided by a line perpendicular to the estimated gradient direction

$$\widehat{G}(x,y) = \left(\frac{\widehat{c}(x,y)}{\sqrt{\widehat{b}(x,y)^2 + \widehat{c}(x,y)^2}}, \frac{-\widehat{b}(x,y)}{\sqrt{\widehat{b}(x,y)^2 + \widehat{c}(x,y)^2}}\right).$$

Let $S_{h_n} = \{(x,y) \in \Omega : d_E((x,y), S) \le h_n\}$. Then, for any $(x,y) \in \Omega_{h_n} \setminus S_{h_n}$, $O_n(x,y)$

does not contain any edge pixel. Let $\widetilde{U}_n(x,y)$ be the half of the $O_n(x,y)$ separated by a line passing (x,y) in the direction perpendicular to the asymptotic direction of $(\widehat{b}(x,y),\widehat{c}(x,y))$, which is discussed in Lemma 2.2, and $\widetilde{d}_{ij}$ be the Euclidean distance from (x_i,y_j) to the asymptotic dividing line (thus, $\widetilde{d}_{ij}$ is non-random). For a function ϕ satisfying the condition that $\sup_{u^2+v^2\le 1}|\phi(u,v)| \le b_\phi < \infty$, we have

$$
\begin{aligned}
& \left| \sum_{U_n(x,y)} \phi(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) K(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) L(d_{ij}/h_n)\frac{1}{n^2h_n^2} - \right. \\
& \left. \sum_{\widetilde{U}_n(x,y)} \phi(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) K(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) L(\widetilde{d}_{ij}/h_n)\frac{1}{n^2h_n^2} \right| \\
\le\ & \frac{1}{n^2h_n^2} \left| \sum_{U_n(x,y)} \phi(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) K(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) L(\widetilde{d}_{ij}/h_n) - \right. \\
& \left. \sum_{\widetilde{U}_n(x,y)} \phi(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) K(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}) L(\widetilde{d}_{ij}/h_n) \right| + O\left(\frac{|d_{ij}-\widetilde{d}_{ij}|}{h_n}\right) \\
\le\ & b_\phi \|K\|_\infty \|L\|_\infty \left| \frac{1}{n^2h_n^2} \sum_{U_n(x,y)\triangle\widetilde{U}_n(x,y)} 1 \right| + O\left(\frac{|d_{ij}-\widetilde{d}_{ij}|}{h_n}\right) \\
=\ & O(\theta_n) = o(1),\ a.s., \qquad (2.41)
\end{aligned}
$$

where θ_n denotes the acute angle between $(\widehat{b}(x,y),\widehat{c}(x,y))$ and its asymptotic direction and $U_n'(x,y) \triangle \widetilde{U}_n'(x,y) = (U_n'(x,y) \setminus \widetilde{U}_n'(x,y)) \cup (\widetilde{U}_n'(x,y) \setminus U_n'(x,y))$. In the first inequality of (3.39), we have used the Lipschitz-1 continuity of L. In the last equation, Lemma 2.2 has been applied. Now, let

$$
\begin{aligned}
\widetilde{b}_{i,j}(x,y) &= [\widetilde{B}_1(x,y) + \widetilde{B}_2(x,y)(x_i-x) + \widetilde{B}_3(x,y)(y_j-y)] \\
&\quad K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right) L(\widetilde{d}_{ij}/h_n), \\
\widetilde{B}_1(x,y) &= \widetilde{t}_{20}(x,y)\widetilde{t}_{02}(x,y) - \widetilde{t}_{11}(x,y)\widetilde{t}_{11}(x,y),
\end{aligned}
$$

$$\begin{aligned}
\widetilde{B}_2(x,y) &= \widetilde{t}_{01}(x,y)\widetilde{t}_{11}(x,y) - \widetilde{t}_{10}(x,y)\widetilde{t}_{02}(x,y), \\
\widetilde{B}_3(x,y) &= \widetilde{t}_{10}(x,y)\widetilde{t}_{11}(x,y) - \widetilde{t}_{01}(x,y)\widetilde{t}_{20}(x,y), \\
\widetilde{t}_{s_1,s_2}(x,y) &= \sum_{\widetilde{U}_n(x,y)} (x_i - x)^{s_1}(y_j - y)^{s_2} K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) L(\widetilde{d}_{ij}/h_n).
\end{aligned}$$

Then, by using similar arguments to those in (3.39), we can check that

$$I_1(x,y) = \frac{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y) H\{f\}(x_i,y_j)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} + O(\theta_n), \ a.s. \tag{2.42}$$

By using (3.39), we have

$$\begin{aligned}
I_2(x,y) &= \sum_{U_n(x,y)} \frac{\widetilde{w}_{ij}(x,y)\frac{1}{n^4h_n^8}}{\frac{1}{n^6h_n^{10}}\sum_{U_n(x,y)} \widetilde{w}_{ij}(x,y)} \frac{1}{n^2h_n^2}\varepsilon_{ij} \\
&= \sum_{U_n(x,y)} \frac{\frac{1}{n^4h_n^8}\widetilde{b}_{ij}(x,y) + O(\theta_n)}{\frac{1}{n^6h_n^{10}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y) + O(\theta_n)} \frac{1}{n^2h_n^2}\varepsilon_{ij} \\
&= \sum_{U_n(x,y)} \left(\frac{\frac{1}{n^4h_n^8}\widetilde{b}_{ij}(x,y)}{\frac{1}{n^6h_n^{10}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} + O(\theta_n)\right) \frac{1}{n^2h_n^2}\varepsilon_{ij} \\
&= \sum_{U_n(x,y)} \frac{\widetilde{b}_{ij}(x,y)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)}\varepsilon_{ij} + \frac{1}{n^2h_n^2}\sum_{U_n(x,y)} O(\theta_n)\varepsilon_{ij} \\
&= \sum_{U_n(x,y)} \frac{\widetilde{b}_{ij}(x,y)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)}\varepsilon_{ij} + O(\theta_n), \ a.s.
\end{aligned} \tag{2.43}$$

In the second equation of (2.43), we have used the results that $\widetilde{B}_1(x,y) = O(n^4h_n^8)$, $\widetilde{B}_2(x,y) = O(n^3h_n^7)$, $\widetilde{B}_3(x,y) = O(n^3h_n^7)$ and $\widetilde{t}_{s_1,s_2}(x,y) = O\left(n^2h_n^{s_1+s_2+2}\right)$ for $s_1, s_2 = 0, 1$. Similar arguments to those in Lemma 2.1 can be applied to $\sum_{U_n(x,y)} \frac{\widetilde{b}_{ij}(x,y)\varepsilon_{ij}}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)}$

, since $\widetilde{b}_{ij}(x,y)$ is deterministic. Consequently, we have

$$\sum_{U_n(x,y)} \frac{\widetilde{b}_{ij}(x,y)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} \varepsilon_{ij} \overset{asy.}{\sim} N\left(0, \sum_{U_n(x,y)} \frac{\widetilde{b}_{ij}^2(x,y)}{\left[\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)\right]^2}\right). \tag{2.44}$$

By (3.26), we have that

$$\begin{aligned}
& \frac{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y) H\{f\}(x_i,y_j)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} \\
= & \frac{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)(f(x_i,y_j)+O(\rho_n^2))}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} \\
= & \frac{\widetilde{B}_1(x,y)}{|\widetilde{\triangle}|} \sum_{\widetilde{U}_n(x,y)} (f(x,y)+f'_x(x,y)(x_i-x)+f'_y(x,y)(y_j-y)+O(h_n^2)+ \\
& O(\rho_n^2))K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) L(\widetilde{d}_{ij}/h_n)+ \\
& \frac{\widetilde{B}_2(x,y)}{|\widetilde{\triangle}|} \sum_{\widetilde{U}_n(x,y)} (f(x,y)+f'_x(x,y)(x_i-x)+f'_y(x,y)(y_j-y)+O(h_n^2)+ \\
& O(\rho_n^2))(x_i-x)K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) L(\widetilde{d}_{ij}/h_n)+ \\
& \frac{\widetilde{B}_3(x,y)}{|\widetilde{\triangle}|} \sum_{\widetilde{U}_n(x,y)} (f(x,y)+f'_x(x,y)(x_i-x)+f'_y(x,y)(y_j-y)+O(h_n^2)+ \\
& O(\rho_n^2))(y_j-y)K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) L(\widetilde{d}_{ij}/h_n) \\
= & f(x,y)+\frac{f'_x(x,y)}{|\widetilde{\triangle}|}(\widetilde{B}_1(x,y)\widetilde{t}_{10}(x,y)+\widetilde{B}_2(x,y)\widetilde{t}_{20}(x,y)+\widetilde{B}_3(x,y)\widetilde{t}_{11}(x,y))+ \\
& \frac{f'_y(x,y)}{|\widetilde{\triangle}|}(\widetilde{B}_1(x,y)\widetilde{t}_{01}(x,y)+\widetilde{B}_2(x,y)\widetilde{t}_{11}(x,y)+\widetilde{B}_3(x,y)\widetilde{t}_{02}(x,y))+O(h_n^2)+O(\rho_n^2) \\
= & f(x,y)+O(h_n^2)+O(\rho_n^2),
\end{aligned} \tag{2.45}$$

where $|\widetilde{\triangle}| = \widetilde{t}_{00}(x,y)\widetilde{t}_{20}(x,y)\widetilde{t}_{02}(x,y) + \widetilde{t}_{10}(x,y)\widetilde{t}_{01}(x,y)\widetilde{t}_{11}(x,y) + \widetilde{t}_{10}(x,y)\widetilde{t}_{01}(x,y)\widetilde{t}_{11}(x,y) - \widetilde{t}_{01}(x,y)^2\widetilde{t}_{20}(x,y) - \widetilde{t}_{11}(x,y)^2\widetilde{t}_{00}(x,y) - \widetilde{t}_{10}(x,y)^2\widetilde{t}_{02}(x,y)$. In the second

equation of (3.42), we have used (3.26). In the last equation, we have used the results that $|\widetilde{\triangle}| = \widetilde{B}_1(x,y)\widetilde{t}_{00}(x,y) + \widetilde{B}_2(x,y)\widetilde{t}_{10}(x,y) + \widetilde{B}_3(x,y)\widetilde{t}_{01}(x,y)$, $\widetilde{t}_{11}(x,y) = 0$ by the symmetry of K and L, $\widetilde{B}_1(x,y)\widetilde{t}_{10}(x,y)+\widetilde{B}_2(x,y)\widetilde{t}_{20}(x,y)+\widetilde{B}_3(x,y)\widetilde{t}_{11}(x,y)=0$, $\widetilde{B}_1(x,y)\widetilde{t}_{01}(x,y)+\widetilde{B}_2(x,y)\widetilde{t}_{11}(x,y)+\widetilde{B}_3(x,y)\widetilde{t}_{02}(x,y) = 0$, and that $\widetilde{B}_1(x,y) = O(n^4h_n^8)$, $\widetilde{B}_2(x,y) = O(n^4h_n^7)$, $\widetilde{B}_3(x,y) = O(n^4h_n^7)$, $|\widetilde{\triangle}| = O(n^6h_n^{10})$, $\widetilde{t}_{s_1,s_2}(x,y) = O\left(n^2h_n^{s_1+s_2+2}\right)$, for $s_1, s_2 = 0, 1$. All these results can be proved similarly to the result (23) in Proposition 2 in Qiu (2009). Now, after combining (2.42), (2.43), (2.44) and (3.42), we have the following result:

$$\widehat{f}_{LL2K,+}(x,y) = f(x,y) + O(h_n^2) + O(\rho_n^2) + O(\theta_n) + \xi_n, \tag{2.46}$$

where $\xi_n \overset{asy.}{\sim} N\left(0, \frac{\sum_{U_n(x,y)} \widetilde{b}_{ij}^2(x,y)}{\left[\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)\right]^2}\right)$. Similarly, we have

$$\widehat{f}_{LL2K,-}(x,y) = f(x,y) + O(h_n^2) + O(\rho_n^2) + O(\theta_n) + \eta_n, \tag{2.47}$$

where $\eta_n \overset{asy.}{\sim} N\left(0, \frac{\sum_{V_n(x,y)} \widetilde{b'}_{ij}^2(x,y)}{\left[\sum_{\widetilde{V}_n(x,y)} \widetilde{b'}_{ij}(x,y)\right]^2}\right)$, $\widetilde{b'}_{ij}(x,y)$ is defined similarly to $\widetilde{b}_{ij}(x,y)$. From the proof of Lemma 2.2, we know that, if (x,y) is not an edge pixel, then

$$\theta_n = O(\rho_n^2) + O(h_n^2) + o\left(\frac{\log(n)}{nh_n^2}\right). \tag{2.48}$$

Therefore, for any design point $(x,y) \in \Omega_{h_n} \setminus S_{h_n}$, by (2.46), (2.47) and (2.48), we have

$$\begin{aligned} \widehat{f}_{\mathtt{LL2K},+} - \widehat{f}_{\mathtt{LL2K},-} &= O(h_n^2) + O(\rho_n^2) + o\left(\frac{\log(n)}{nh_n^2}\right) \\ &+ \gamma_n \cdot \sqrt{\frac{\sum_{U_n(x,y)} \widetilde{b}_{ij}^2(x,y)}{\left[\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)\right]^2} + \frac{\sum_{V_n(x,y)} \widetilde{b'}_{ij}^2(x,y)}{\left[\sum_{\widetilde{V}_n(x,y)} \widetilde{b'}_{ij}(x,y)\right]^2}}, \end{aligned} \tag{2.49}$$

where $\gamma_n \overset{asy.}{\sim} N(0,1)$. Also, by using similar arguments to those in (3.39) and the fact that $\widetilde{b}_{ij}(x,y) = O(n^4 h_n^8)$, we have

$$\begin{aligned}
& \frac{\sum_{U_n(x,y)} w_{ij}^2(x,y)}{[\sum_{U_n(x,y)} w_{ij}(x,y)]^2} \\
= & \frac{n^{10}h_n^{18}\frac{1}{n^{10}h_n^{18}}\sum_{U_n(x,y)} w_{ij}^2(x,y)}{\left[n^6 h_n^{10}\frac{1}{n^6 h_n^{10}}\sum_{U_n(x,y)} w_{ij}(x,y)\right]^2} \\
= & \frac{n^{10}h_n^{18}\left(\frac{1}{n^{10}h_n^{18}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}^2(x,y) + O(\theta_n)\right)}{n^{12}h_n^{20}\left[\left(\frac{1}{n^6 h_n^{10}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y) + O(\theta_n)\right)\right]^2} \\
= & \frac{1}{n^2 h_n^2}\frac{\frac{1}{n^{10}h_n^{18}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}^2(x,y) + O(\theta_n)}{\left[\frac{1}{n^6 h_n^{10}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y) + O(\theta_n)\right]^2} \\
= & \frac{1}{n^2 h_n^2}\left\{\frac{\frac{1}{n^{10}h_n^{18}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}^2(x,y)}{\left[\frac{1}{n^6 h_n^{10}}\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)\right]^2} + O(\theta_n)\right\},\ a.s. \qquad (2.50)
\end{aligned}$$

Then, it follows that

$$\sqrt{\frac{\sum_{U_n(x,y)} w_{ij}^2(x,y)}{[\sum_{U_n(x,y)} w_{ij}(x,y)]^2} + \frac{\sum_{V_n(x,y)} w_{ij}'^2(x,y)}{[\sum_{V_n(x,y)} w_{ij}'(x,y)]^2}} = O\left(\frac{1}{nh_n}\right),\ a.s. \qquad (2.51)$$

By ((2.42)), ((2.43)), ((3.42)), ((2.48)) and ((3.47)), we have

$$\begin{aligned}
\frac{\mathrm{LL2K}_n(x,y)}{Z_{1-\alpha_n}} &= O\left(\frac{nh_n\rho_n^2}{Z_{1-\alpha_n}}\right) + O\left(\frac{nh_n^3}{Z_{1-\alpha_n}}\right) + o\left(\frac{\log(n)}{h_n Z_{1-\alpha_n}}\right) + o\left(\frac{\log(n)}{Z_{1-\alpha_n}}\right) \\
&= O\left(\frac{nh_n^3}{Z_{1-\alpha_n}}\right),\ a.s., \qquad (2.52)
\end{aligned}$$

where we have used the conditions that $\frac{\rho_n}{h_n} = o(1)$, and $\frac{\log(n)}{nh_n^4} = o(1)$. Hence, if $\frac{nh_n^3}{Z_{1-\alpha_n}} = o(1)$, any point $(x,y) \in \Omega \setminus S_{h_n}$ will not be flagged as a jump candidate when n is sufficient large.

Now, let us consider a non-singular design point (x, y) on a step edge segment that has a unique tangent line at (x, y). As discussed in Lemma 2.2, we may assume the step edge segment is the same as the tangent line in a small neighborhood. Let S_{ρ_n} be a band of width $2\rho_n$ that contains S. Then we have

$$\begin{aligned}
& \frac{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y) H\{f\}(x_i, y_j)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} \\
= & \frac{\sum_{\widetilde{U}_n(x,y)\setminus S_{\rho n}} \widetilde{b}_{ij}(x,y) H\{f\}(x_i, y_j)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} + \frac{\sum_{\widetilde{U}_n(x,y)\bigcap S_{\rho n}} \widetilde{b}_{ij}(x,y) H\{f\}(x_i, y_j)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} \\
= & \frac{\sum_{\widetilde{U}_n(x,y)\setminus S_{\rho n}} \widetilde{b}_{ij}(x,y)(f_+(x,y) + O(h_n) + O(\rho_n^2))}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} + O\left(\frac{\rho_n}{h_n}\right) \\
= & [f_+(x,y) + O(h_n) + O(\rho_n^2))] \left(1 - O\left(\frac{\rho_n}{h_n}\right)\right) + O\left(\frac{\rho_n}{h_n}\right) \\
= & f_+(x,y) + O(h_n) + O(\rho_n/h_n),
\end{aligned}$$

where $f_+(x, y)$ denotes the limit of $f(u, v)$ as (u, v) approaching to (x, y) form $\widetilde{U}_n(x, y)$. In the second equation we have used the fact that the ratio of the area of $\widetilde{U}_n(x, y) \bigcap S_{\rho_n}$ to the area of $\widetilde{U}_n(x, y)$ is of order $\frac{\rho_n}{h_n}$. In the third equation, (3.26) has been used. So, we have

$$I_1(x,y) = f_+(x,y) + O(h_n^2) + O(\rho_n/h_n) + O(\theta_n). \tag{2.53}$$

By ((2.43)) and Lemma 2.1, we have

$$I_2(x,y) = \sum_{U_n(x,y)} \frac{\widetilde{b}_{ij}(x,y)}{\sum_{\widetilde{U}_n(x,y)} \widetilde{b}_{ij}(x,y)} \varepsilon_{ij} + O(\theta_n) = o\left(\frac{\log(n)}{nh_n}\right) + O(\theta_n) \text{ a.s.} \tag{2.54}$$

From the proof of Lemma 2.2, we know that, when (x, y) is a non-singular point,

$$\theta_n = O(h_n) + O(\rho_n/h_n) + o\left(\frac{\log(n)}{nh_n}\right).$$

Thus,

$$\widehat{f}_{\text{LL2K},+}(x, y) = f_+(x, y) + O(h_n) + O(\rho_n/h_n) + o\left(\frac{\log(n)}{nh_n}\right). \tag{2.56}$$

Similarly, we can derive the result that

$$\widehat{f}_{\text{LL2K},-}(x, y) = f_-(x, y) + O(h_n) + O(\rho_n/h_n) + o\left(\frac{\log(n)}{nh_n}\right), \tag{2.57}$$

where $f_-(x, y)$ is defined similarly to $f_+(x, y)$. Then, a direct conclusion from (2.56), (2.57) and (3.47) is that

$$\begin{aligned}\frac{\text{LL2K}_n(x, y)}{Z_{1-\alpha_n}} &= O\left(\frac{nh_n(f_+(x, y) - f_-(x, y))}{Z_{1-\alpha_n}}\right) + \\ &\quad O(n\rho_n/Z_{1-\alpha_n}) + O\left(\frac{nh_n^2}{Z_{1-\alpha_n}}\right) + O\left(\frac{\log(n)}{h_n Z_{1-\alpha_n}}\right) \\ &= O\left(\frac{nh_n(f_+(x, y) - f_-(x, y))}{Z_{1-\alpha_n}}\right), \ a.s., \end{aligned} \tag{2.58}$$

where we have used the results that $h_n = o(1)$, $\frac{\rho_n}{h_n} = o(1)$, and $\frac{\log(n)}{nh_n^2} = o(1)$. Thus, in the case when (x, y) is a non-singular point and the step edge has a unique tangent line at (x, y), LL2K would detect (x, y) successfully when n is sufficiently large. The parallel result to ((2.58)) can be derived for the case when the jump location curve has two one-sided tangent lines at (x, y). Therefore, the LL2K edge detector can detect all points in $S \bigcap \Omega_{h_n} \bigcap \overline{J}_{S,h_n}$. And, all points whose Euclidean distances to S are larger than h_n would not be detected. So, when n is large enough, $S \bigcap \Omega_{h_n} \bigcap \overline{J}_{S,h_n}$ is included in $\widehat{S}_n$, and $\widehat{S}_n$ is included in the band of S with width h_n. By similar

arguments, it can be shown that this result also holds for the edge detectors LCK, LC2K and LLK. Thus, all results in the theorem are valid. □

Chapter 3

Blind Image Deblurring Based on Edge Detection

3.1 Introduction

Observed images generated by image acquisition devices are usually not exactly the same as the true images, but are degraded versions of their true images , as discussed in Chapter 1. *Image deblurring* is for estimating the true image f from its observed but degraded version Z. This problem is generally "ill-posed" for the following reasons. First, even in cases when no noise is contained in Z, there could be multiple sets of h and f that correspond to the same Z, when h is unspecified beforehand. Second, the inverse problem to estimate f from Z often involves numerical instability, caused mainly by noise. For instance, in cases when h is location invariant and completely specified, from (2.1), we have

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

where $\mathcal{F}\{f\}$ denotes the Fourier transformation of f. Then, intuitively, a reasonable estimator of f can be defined by the inverse Fourier transformation of $\mathcal{F}\{Z\}(u,v)/\mathcal{F}\{h\}(u,v)$. However, when u^2+v^2 gets larger, $\mathcal{F}\{h\}(u,v)$ converges to zero fast because h is usually a smooth function, but $\mathcal{F}\{Z\}(u,v)$ converges to zero much slower because

of the high-frequency noise contained in Z. Therefore, such an estimator would be numerically unstable.

In the literature, early image deblurring procedures are under the assumption that h is completely specified. In such cases, a major challenge is to overcome the numerical difficulty mentioned above. To this end, many proposals have been suggested, including inverse filtering, Wiener filtering, Lucy-Richardson algorithm, maximum *a posteriori* (MAP) procedure, procedures based on EM algorithm, total variation image deblurring, and so forth (e.g., Biggs and Andrews (1997), Figueiredo and Nowak (2003), Gonzalez and Woods (2002), Skilling (1989), Oliveira et al. (2009).) In applications, however, it is often difficult to specify h completely. Image deblurring when h is unspecified is referred to be *blind image deblurring (BID)* problem in the literature, which is challenging due to its "ill-posed" nature described above. There are a number of existing procedures to handle the BID problem. Some of them assume that h follows a parametric model with one or more unknown parameters, and estimate the parameters together with the true image f by certain algorithms (e.g., Carasso (2001), Hall and Qiu (2007a), Joshi and Chaudhuri (2005), Katsaggelos and Lay (1990)). Some others assume that the true image f has one or more regions with certain known edge structures (e.g., Hall and Qiu (2007b), Kundur and Hatzinakos (1998), Qiu (2008), Yang et al. (1994)). Several authors formulate the BID problem as a regularization problem with regularization measures on both h and f (e.g., Chan and Wong (1998), Rudin et al. (1992), You and Kaveh (1996)). Some others develop BID methods under the Bayesian framework (e.g., Fergus et al. (2006)).

In this chapter, we provide an alternative approach to the BID problem. We notice that blurring has the following hierarchical nature. It alters the image structure most significantly at step edges (i.e., places where f has jumps), less significantly at roof/valley edges (i.e., places where the first-order derivatives of f has jumps), and least significantly at places where f is straight. Therefore, our approach pays special

attention to regions around jumps in f and in its derivatives of different orders. In practice, jumps in the second or higher order derivatives of f is hardly visible. For this reason, only step and roof/valley edges are specially treated when deblurring in this chapter, although jumps in the second or higher order derivatives of f can be treated similarly. The major statistical tool used here is the jump regression analysis (cf., Qiu (2005)). Under that framework, model (2.1) is regarded as a 2-dimensional (2-D) regression model, f is a jump regression surface, and various edges correspond to jumps in f and in its derivatives of different orders. Local smoothing procedures are proposed under that framework for detecting step and roof/valley edges and for estimating f by removing both point-wise noise and spatial blur. Our proposed approach is a three-stage procedure. In the first stage, candidate edge pixels (i.e., step edge pixels and roof/valley edge pixels) are detected using a edge detection criterion. In the second stage, a local principal component line is fitted through detected edge pixels in a neighborhood of a given pixel. If there is a single edge curve in the neighborhood, then this principal component line provides a first-order approximation to the edge curve in that neighborhood. In the third stage, observations on the same side of the line as the given pixel are combined using a weighted average procedure to estimate the intensity at the given pixel. One major feature of this approach is that it does not require restrictive assumptions on either h or f. It even allows h to change over location.

The remaining part of this chapter is organized as follows. In the next section, our proposed methodology is described in detail. Some of its statistical properties are presented in Section 3.3. Its numerical performance is investigated in Section 3.4. Some discussions are presented in Section 3.5. Some technical details are provided in Section 3.6.

3.2 Methodology

Our method is described in three parts. Detection of step and roof/valley edges is discussed in Subsection 3.2.1. Estimation of f is discussed in Subsection 3.2.2. Selection of procedure parameters is discussed in Subsection 3.2.3.

3.2.1 Edge Detection

Detection of step edges

Since step edge detection in noisy blurred images has been carefully discussed in the previous chapter, we are going to use step edge detector (2.14) to serve the purpose here without much explanation, although any of the four proposed detectors can be used in principle. For convenience of description, slightly different notations are adopted in this chapter. As did in Chapter 1, Assume that $\{(x_i, y_j, Z_{ij}), i = 1, 2, \ldots, n, j = 1, 2, \ldots, n\}$ follow model (2.2). Namely,

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

where ε_{ij} are i.i.d. errors with mean 0 and unknown variance σ^2. For a given design point $(x, y) \in [k_1/n, 1 - k_1/n]^2$, where $k_1 < n$ is a positive integer and k_1/n called bandwidth parameter for step edge detection, we consider its circular neighborhood:

$$O'_n(x, y) = \left\{(u, v) : (u, v) \in \Omega \text{ and } \sqrt{(u - x)^2 + (v - y)^2} \leq k_1/n\right\}.$$

In this neighborhood, let us consider the following LLK smoothing procedure:

$$\min_{a,b,c} \sum_{i^2+j^2 \leq k_1^2} \left\{ Z\left(x + \frac{i}{n}, y + \frac{j}{n}\right) - \left[a + b\frac{i}{n} + c\frac{j}{n}\right] \right\}^2 K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right), \tag{3.2}$$

where $Z\left(x + \frac{i}{n}, y + \frac{j}{n}\right)$ denotes the observed value of the intensity function f at design point $\left(x + \frac{i}{n}, y + \frac{j}{n}\right)$, 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 (3.2) be denoted as $\{\widehat{a}(x,y), \widehat{b}(x,y), \widehat{c}(x,y)\}$. Then, $(\widehat{b}(x,y), \widehat{c}(x,y))$ 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 $(\widehat{b}(x,y), \widehat{c}(x,y))$. Then the step edge detector defined in (2.14), denoted as $\mathrm{LL2K}'_n(x,y)$, is as follows.

$$\mathrm{LL2K}'_n(x,y) = \frac{\left|\widehat{f}_{LL2K,+}(x,y) - \widehat{f}_{LL2K,-}(x,y)\right|}{\sqrt{\frac{\sum_{(x_i,y_j)\in U'_n(x,y)} b_{ij}(x,y)^2}{\left[\sum_{(x_i,y_j)\in U'_n(x,y)} b_{ij}(x,y)\right]^2} + \frac{\sum_{(x_i,y_j)\in V'_n(x,y)} b'_{ij}(x,y)^2}{\left[\sum_{(x_i,y_j)\in V'_n(x,y)} b'_{ij}(x,y)\right]^2}}}, \tag{3.3}$$

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:

$$\min_{a_0,a_1,a_2} \sum_{U'_n(x,y)} \left\{ Z(x+i/n, y+j/n) - \left[a_0 + a_1\frac{i}{n} + a_2\frac{j}{n}\right]\right\}^2 K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), \tag{3.4}$$

$$\min_{a_0,a_1,a_2} \sum_{V'_n(x,y)} \left\{ Z(x+i/n, y+j/n) - \left[a_0 + a_1\frac{i}{n} + a_2\frac{j}{n}\right]\right\}^2 K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), \tag{3.5}$$

$$b_{ij}(x,y) = \left[B_1(x,y) + B_2(x,y)\frac{i}{n} + B_3(x,y)\frac{j}{n}\right] K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), ,$$

$$\begin{aligned} B_1(x,y) &= t_{20}(x,y)t_{02}(x,y) - t_{11}(x,y)t_{11}(x,y), \\ B_2(x,y) &= t_{01}(x,y)t_{11}(x,y) - t_{10}(x,y)t_{02}(x,y), \end{aligned}$$

$$\begin{aligned} B_3(x,y) &= t_{10}(x,y)t_{11}(x,y) - t_{01}(x,y)t_{20}(x,y), \\ t_{s_1,s_2}(x,y) &= \sum_{U'_n(x,y)} (i/n)^{s_1}(j/n)^{s_2} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), \text{ for } s_1, s_2 = 0, 1, 2, \end{aligned}$$

L^* is a univariate increasing density kernel function with support $[0,1]$, 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 $V'_n(x,y)$, and $b'_{ij}(x,y)$ are defined in the way as $b_{ij}(x,y)$ except $U'_n(x,y)$ should be replaced by $V'_n(x,y)$. The design point (x,y) is flagged as a detected step edge pixel if

$$\text{LL2K}'_n(x,y) > u_n\sigma, \tag{3.6}$$

where u_n is a threshold value. As pointed out by Qiu and Yandell (1997), detected edge pixels by (3.6) usually contain two types of deceptive ones: those scattered in the whole design space due to randomness and those around true edge curves due to thresholding. They can be deleted reasonably well by the two modification procedures proposed in that paper. 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 residual mean squares of the jump-preserving surface estimation procedure suggested by Qiu (2004).

Detection of roof/valley edges

Next, we describe our proposed roof/valley edge detection procedure. The methodology is similar to the step edge detection procedure described above. For any given design point $(x,y) \in [k_2/n, 1-k_2/n] \times [k_2/n, 1-k_2/n]$, where $k_2/n \in (0, 0.5)$ is the bandwidth parameter for roof/valley edge detection, let us consider a circular neighborhood $O''_n(x,y) = \{(u,v) : (u,v) \in \Omega \text{ and } \sqrt{(u-x)^2 + (v-y)^2} \le k_2/n\}$ and the

following local quadratic kernel (LQK) smoothing procedure:

$$\min_{c_0,c_1,c_2,c_3,c_4,c_5\in R} \sum_{i^2+j^2\le k_2^2} \left\{ Z\left(x+i/n, y+j/n\right) - \left[c_0 + c_1\frac{i}{n} + c_2\frac{j}{n} + \frac{c_3}{2}\left(\frac{i}{n}\right)^2 + \frac{c_4}{2}\left(\frac{j}{n}\right)^2 + c_5\frac{ij}{n^2}\right]\right\}^2 K^*\left(\frac{i}{k_2}, \frac{j}{k_2}\right). \tag{3.7}$$

Let$(\widehat{c}_0(x,y), \widehat{c}_1(x,y), \widehat{c}_2(x,y), \widehat{c}_3(x,y), \widehat{c}_4(x,y), \widehat{c}_5(x,y))$ denote the solution to $(c_0, c_1, c_2, c_3, c_4, c_5)$ of (3.7). Then, $\widehat{c}_3(x,y)$ and $\widehat{c}_4(x,y)$ are LQK estimators of $f''_{xx}(x,y)$ and $f''_{yy}(x,y)$. Similar to step edge detection, 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 $(\widehat{c}_3(x,y), \widehat{c}_4(x,y))$. We should point out that, in this paper, only jumps in f'_x along the x-direction and jumps in f'_y along the y-direction are considered in roof/valley edge detection because they include most roof/valley edges as their special cases. To detect jumps in f'_x, we define

$$\mathrm{M}^{(2)}_{1,n}(x,y) = \frac{\left|\widehat{b}_+(x,y) - \widehat{b}_-(x,y)\right|}{\sqrt{\frac{\sum_{U''_n(x,y)} g_{ij}(x,y)^2}{\left[\sum_{U''_n(x,y)} g_{ij}(x,y)\right]^2} + \frac{\sum_{V''_n(x,y)} g'_{ij}(x,y)^2}{\left[\sum_{V''_n(x,y)} g'_{ij}(x,y)\right]^2}}}, \tag{3.8}$$

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

$$\min_{a,b,c} \sum_{U''_n(x,y)} \left[Z(x+i/n, y+j/n) - \left(a + b\frac{i}{n} + c\frac{j}{n}\right)\right]^2 K^*\left(\frac{i}{k_2}, \frac{j}{k_2}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), \tag{3.9}$$

$$\min_{a,b,c} \sum_{V''_n(x,y)} \left[Z(x+i/n, y+j/n) - \left(a + b\frac{i}{n} + c\frac{j}{n}\right)\right]^2 K^*\left(\frac{i}{k_2}, \frac{j}{k_2}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), \tag{3.10}$$

$$
\begin{aligned}
g_{ij}(x,y) &= \left[G_1(x,y) + G_2(x,y)\frac{i}{n} + G_3(x,y)\frac{j}{n}\right] K^*\left(\frac{i}{k_2},\frac{j}{k_2}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), \\
G_1(x,y) &= u_{11}(x,y)u_{01}(x,y) - u_{10}(x,y)u_{02}(x,y), \\
G_2(x,y) &= u_{00}(x,y)u_{02}(x,y) - u_{01}(x,y)u_{01}(x,y), \\
G_3(x,y) &= u_{01}(x,y)u_{10}(x,y) - u_{00}(x,y)u_{11}(x,y), \\
u_{s_1,s_2}(x,y) &= \sum_{U''_n(x,y)} (i/n)^{s_1}(j/n)^{s_2} K^*\left(\frac{i}{k_2},\frac{j}{k_2}\right) L^*\left(\frac{d'_{ij}}{k_1/n}\right), \text{ for } s_1, s_2 = 0,1,2,
\end{aligned}
$$

$g'_{ij}(x,y)$ is defined in the same way as $g_{ij}(x,y)$, except that $U''_n(x,y)$ in the definition of $u_{s_1,s_2}(x,y)$ should be replaced by $V''_n(x,y)$, and d''_{ij} is the Euclidean distance from the design point (x_i, y_j) to the line separating $U''_n(x,y)$ from $V''_n(x,y)$. Similarly, to detect jumps in f'_y, we define

$$
\mathrm{M}^{(2)}_{2,n}(x,y) = \frac{|\widehat{c}_+(x,y) - \widehat{c}_-(x,y)|}{\sqrt{\frac{\sum_{U''_n(x,y)} h_{ij}(x,y)^2}{\left[\sum_{U''_n(x,y)} h_{ij}(x,y)\right]^2} + \frac{\sum_{V''_n(x,y)} h'_{ij}(x,y)^2}{\left[\sum_{V''_n(x,y)} h'_{ij}(x,y)\right]^2}}}, \tag{3.11}
$$

where $\widehat{c}_+(x,y)$ and $\widehat{c}_-(x,y)$ are respectively the solutions to c of (3.9) and (3.10),

$$
\begin{aligned}
h_{ij}(x,y) &= \left[H_1(x,y) + H_2(x,y)\frac{i}{n} + H_3(x,y)\frac{j}{n}\right] K^*\left(\frac{i}{k_2},\frac{j}{k_2}\right) L^*(d''_{ij}/(k_2/n)), \\
H_1(x,y) &= u_{10}(x,y)u_{11}(x,y) - u_{01}(x,y)u_{20}(x,y), \\
H_2(x,y) &= u_{01}(x,y)u_{10}(x,y) - u_{00}(x,y)u_{11}(x,y), \\
H_3(x,y) &= u_{00}(x,y)u_{20}(x,y) - u_{10}(x,y)u_{10}(x,y),
\end{aligned}
$$

and $h'_{ij}(x,y)$ is defined in the same way as $h_{ij}(x,y)$, except that $U''_n(x,y)$ in the definition of $u_{s_1,s_2}(x,y)$ should be replaced by $V''_n(x,y)$. Then, the design point (x,y) is flagged as a roof/valley edge pixel if

$$
\mathrm{M}^{(2)}_n(x,y) = \max\left\{\mathrm{M}^{(2)}_{1,n}(x,y), \mathrm{M}^{(2)}_{2,n}(x,y)\right\} > v_n\sigma,
$$

where v_n is a threshold value. However, this criterion could be large around step edges too, due to the zero-crossing properties of the second-order derivatives around step edges (cf., Qiu 2005, Figure 6.2). To overcome this difficulty, we propose flagging (x, y) as a roof/valley edge pixel if

$$I_n(x, y) = 0, \text{ and } \mathrm{M}_n^{(2)}(x, y) > v_n\sigma, \tag{3.12}$$

where $I_n(x, y)$ denotes the number of detected step edge pixels in $O_n''(x, y)$. The two modification procedures in Qiu and Yandell (1997) can also be used here to remove the two types of deceptive roof/valley edge pixels detected by (3.12). Again, σ should be replaced by $\widehat{\sigma}$ in practice.

3.2.2 Blind image deblurring

As described in Section 1, our proposed BID procedure pays special attention to regions around the detected step and roof/valley edges when deblurring the observed image Z (or, estimating the true image f from Z). To estimate f at a given design point (x, y), let us consider a circular neighborhood

$$O_n(x, y) = \{(u, v) : \sqrt{(u - x)^2 + (v - y)^2} \leq k/n\},$$

where k/n is a bandwidth that could be different from the bandwidths k_1/n or k_2/n used in edge detection. Let $\{(w_l, v_l), l = 1, 2, \ldots, m\}$ be detected step edge points in $O_n(x, y)$, $\bar{w}, \bar{v}, \sigma_{ww}$ and σ_{vv} be the sample means and sample variances of $\{w_l, l = 1, 2, \ldots, m\}$ and $\{v_l, l = 1, 2, \ldots, m\}$, σ_{wv} be their sample covariance, and (W, V) be a vector variable taking values over $\{(w_l, v_l), l = 1, 2, \cdots, m\}$. To estimate the underlying step edge segment in $O_n(x, y)$ from the detected step edge points, we consider using the principal component (PC) line of the points $\{(w_l, v_l), l = 1, 2, \ldots, m\}$, which goes through the center of these points along the direction that they have the biggest

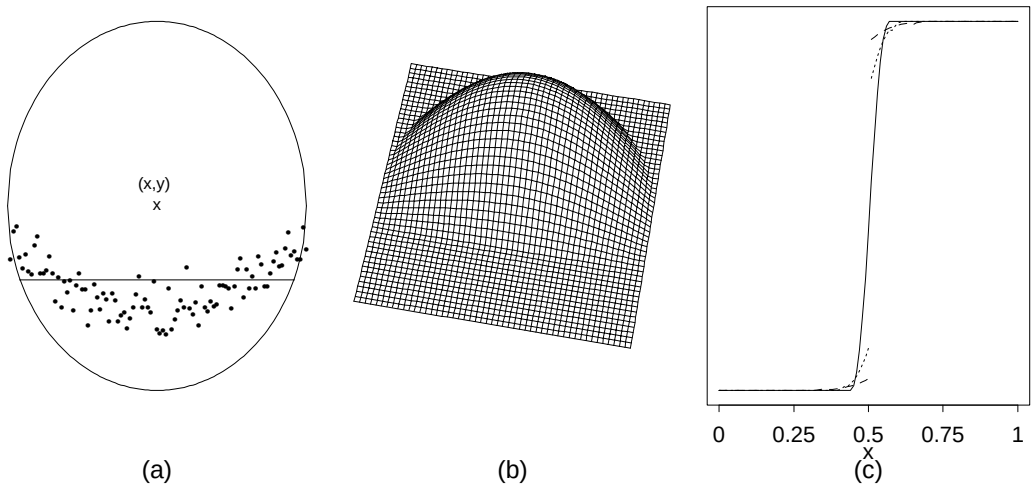

Figure 3.1: (a): In neighborhood $O_n(x, y)$, the PC line (solid line) goes through the center of the detected step edge points (small dots) along the direction that they have the biggest dispersion. (b): In $O_n(x, y)$, a typical weighting function used in (3.14) is shown by the surface. (c): A cross section of a blurred image intensity surface around a step edge (solid line) and the deblurred versions by (3.15) when the bandwidth k/n is relatively small (dotted line) and relatively large (dashed line).

dispersion. See Figure 1(a) for a demonstration. The PC line has the expression

$$\sigma_{wv}(W - \bar{w}) + (\lambda_1 - \sigma_{ww})(V - \bar{v}) = 0, \tag{3.13}$$

where

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

is the smaller eigenvalue of the sample covariance matrix of $\{(w_l, v_l), l = 1, 2, \ldots, m\}$. In cases when no blurring is involved in the observed image Z, Qiu (1998) has shown that this PC line provides a good approximation to the underlying step edge segment if the step edge segment has unique tangent line at (x, y). In Section 3, we will show that this is still true in cases when Z contains spatial blur.

Intuitively, if Z contains no blur, then $f(x, y)$ can be estimated by a weighted average of the observations located on the same side of the PC line as (x, y), as did

in the image denoising literature (cf., Qiu 1998). In cases when blurring is present, if a design point in $O_n(x, y)$ is closer to the PC line, then it is more likely that the corresponding observed image intensity has blurring involved, as discussed in Section 2. Thus, it should receive a smaller weight in the weighted average. To address this issue, similar to edge detection, besides a 2-D kernel function used in conventional local smoothing to assign more weights to design points closer to (x, y), a univariate kernel function L is used to assign less weights to design points closer to the PC line. Then, the deblurred image $\widehat{f}(x, y)$ is defined by the solution to a_0 of the following local constant kernel (LCK) smoothing procedure:

$$\min_{a_0 \in R} \sum_{U_n(x,y)} \{Z(x + i/n, y + j/n) - a_0\}^2 K\left(\frac{i}{k}, \frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n + d(x, y)}\right), \quad (3.14)$$

where K is a circularly symmetric 2-D kernel density function that could be different from the kernel function K^* used in edge detection, L is a univariate increasing density kernel function with support $[0, 1]$ that could also be different from the kernel function L^*, $d(x, y)$ is the Euclidean distance from point (x, y) to the PC line, d_{ij} is the Euclidean distance from point (x_i, y_j) to the PC line, and $U_n(x, y)$ denotes the set of design points in $O_n(x, y)$ that are on the same side of the PC line as (x, y). That is,

$$U_n(x, y) := \begin{cases} \{(x_i, y_j) : \sigma_{wv}(x_i - \bar{w}) + (\lambda_1 - \sigma_{ww})(y_j - \bar{v}) \geq 0, \ \ (x_i, y_j) \in O_n(x, y)\}, \\ \qquad\qquad \text{if } \sigma_{wv}(x - \bar{w}) + (\lambda_1 - \sigma_{ww})(y - \bar{v}) \geq 0; \\ \{(x_i, y_j) : \sigma_{wv}(x_i - \bar{w}) + (\lambda_1 - \sigma_{ww})(y_j - \bar{v}) < 0, \ \ (x_i, y_j) \in O_n(x, y)\}, \\ \qquad\qquad \text{otherwise.} \end{cases}$$

By certain routine algebraic manipulations, we have

$$\widehat{f}(x, y) = \frac{\sum_{(x_i,y_j) \in U_n(x,y)} \widetilde{w}_{ij}(x, y) Z_{ij}}{\sum_{(x_i,y_j) \in U_n(x,y)} \widetilde{w}_{ij}(x, y)}, \quad (3.15)$$

where

$$\widetilde{w}_{ij}(x,y) = K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n + d(x,y)}\right).$$

By (3.14) and (3.15), we actually fit a constant in $O_n(x,y)$, using a weighted average scheme. The weights are controlled by $K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n + d(x,y)}\right)$. When K and L are chosen to be the ones used in Section 4 and when the detected step edge points are those shown in Figure 1(a), the weights used in (3.14) when estimating $f(x,y)$ are demonstrated by the surface shown in Figure 1(b). From the plot, we can see that (i) only those design points that are on the same side of the PC line as (x,y) would receive positive weights, and (ii) a given design point would receive more weight if it is closer to (x,y) and farther away from the PC line. Because of this weighting scheme, the fitted plane in $O_n(x,y)$ is mainly determined by observations whose design points are certain distance away from the PC line. Consequently, $\widehat{f}(x,y)$ would not be affected much by the blurring around the underlying step edge segment in $O_n(x,y)$, especially when the blurring extent $r_n(x,y)$ (see its definition in Section 3) is relatively small, compared to k. In Figure 1(c), a cross section of a blurred image intensity function around a step edge is shown by the solid line. The dotted and dashed lines denote the deblurred image intensity functions by (3.15) when $k = r_n(x,y)$ and $k = 2r_n(x,y)$, respectively. It can be seen that (i) the BID procedure (3.15) does have the ability to deblur the image around step edges, and (ii) it would deblur the image better if the ratio r_n/k is smaller. The second conclusion implies that procedure (3.15) would perform better if the blurring extent is smaller. If the blurring extent is relatively large, then the deblurred image by (3.15) may still contain certain blur, although the blur is mostly eliminated, because (3.15) cannot use a very large bandwidth in order to avoid large bias in local smoothing. More theoretical and numerical justifications are given in the next two sections about the proposed BID procedure (3.15).

Formula (3.15) is for image deblurring around the detected step edges. In $O_n(x,y)$, if the number of detected step edge points, denoted as $I'_n(x,y)$, is so small that it is unlikely to have a step edge segment in $O_n(x,y)$ (e.g., $I'_n(x,y) < k$,), then we check to see whether $O_n(x,y)$ would possibly contain a roof/valley edge segment. In the case when the number of detected roof/valley edge points, denoted as $I''_n(x,y)$, is quite large (e.g., $I''_n(x,y) \geq k$) and it is possible to have a roof/valley edge segment in $O_n(x,y)$, $\widehat{f}(x,y)$ can still be defined by (3.15), except that the detected step edge points $\{(w_l, v_l), l = 1, 2, \ldots, m\}$ should be replaced by the detected roof/valley edge points in its definition. The corresponding estimator $\widehat{f}(x,y)$ should deblur the image well around the detected roof/valley edge points, as explained above about image deblurring around the detected step edge points. In cases when both $I'_n(x,y)$ and $I''_n(x,y)$ are so small that $O_n(x,y)$ is unlikely to contain any step or roof/valley edge segments, we suggest estimating $f(x,y)$ by the conventional LLK estimator, which is the solution to a_0 of the LLK procedure

$$\min_{a_0,a_1,a_2 \in R} \sum_{i^2+j^2 \leq k^2} \left[Z(x+i/n, y+j/n) - \left(a_0 + a_1 \frac{i}{n} + a_2 \frac{j}{n} \right) \right]^2 K\left(\frac{i}{k}, \frac{j}{k} \right). \quad (3.16)$$

In such cases, $\widehat{f}(x,y)$ has the expression

$$\widehat{f}(x,y) = \frac{\sum_{i^2+j^2 \leq k^2} w_{ij}(x,y) Z_{ij}}{\sum_{i^2+j^2 \leq k^2} w_{ij}(x,y)}, \quad (3.17)$$

where

$$w_{ij}(x,y) = \left[A_1(x,y) + A_2(x,y)\frac{i}{n} + A_3(x,y)\frac{j}{n} \right] K\left(\frac{i}{k}, \frac{j}{k} \right),$$

$$\begin{aligned} A_1(x,y) &= r_{20}(x,y) r_{02}(x,y) - r_{11}(x,y) r_{11}(x,y), \\ A_2(x,y) &= r_{01}(x,y) r_{11}(x,y) - r_{10}(x,y) r_{02}(x,y), \end{aligned}$$

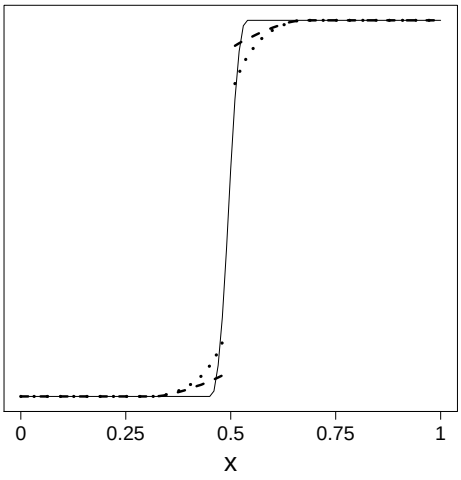

Figure 3.2: A cross section of a blurred image intensity surface around a step edge (solid line), the deblurred version by (3.15) with local constant kernel estimation (dashed line), and the deblurred version by (3.15) with local linear kernel estimation (dotted line).

$$\begin{aligned} A_3(x,y) &= r_{10}(x,y)r_{11}(x,y) - r_{01}(x,y)r_{20}(x,y), \\ r_{s_1s_2}(x,y) &= \sum_{i^2+j^2\le k^2} \left(\frac{i}{n}\right)^{s_1} \left(\frac{j}{n}\right)^{s_2} K\left(\frac{i}{k}, \frac{j}{k}\right), \text{ for } s_1, s_2 = 0, 1, 2. \end{aligned}$$

By comparing (3.17) with (3.15), it can be seen that, when estimating $f(x, y)$, an LCK estimator is used in (3.15) in cases when (x, y) is close to a step or roof/valley edge segment, while an LLK estimator is used in (3.17) in cases when (x, y) is far away from any step or roof/valley edge. That is because the LCK estimator is more robust to spatial blur around edges, as demonstrated by Figure Figure 3.2, and the LLK estimator is less biased in continuity regions (cf., Qiu (2005), Chapter 2). In practice, a regular image usually contains some regions where no step or roof/valley edge segments are present. So, at a given pixel (x, y), before estimating a PC line from the detected step or roof/valley edge pixels in $O_n(x, y)$, we suggest making a judgment to insure that a step or roof/valley edge segment is possible in $O_n(x, y)$. One major benefit to do so is that the image estimator in (3.17) would be more efficient than the

one in (3.15) at places without any step or roof/valley edge segments, because of the fact that the former estimator is constructed from all observations in $O_n(x, y)$ while the latter estimator uses only part observations in $O_n(x, y)$. After taking all these considerations into account, our proposed BID procedure is summarized below.

Proposed Blind Image Deblurring Procedure

1. Detect step and roof/valley edge points by (3.6) and (3.12).

2. At a given point (x, y), if the number of detected step edge points $I'_n(x, y) \geq k$, then estimate $f(x, y)$ by (3.13)–(3.15).

3. If $I'_n(x, y) < k$ but $I''_n(x, y) \geq k$, then still estimate $f(x, y)$ by (3.13)–(3.15), after the detected step edge points are replaced by the detected roof/valley edge points in $O_n(x, y)$ when computing the estimator.

4. If $I'_n(x, y) < k$ and $I''_n(x, y) < k$, then estimate $f(x, y)$ by (3.16) and (3.17).

In cases when $I'_n(x, y) \geq k$ and $I''_n(x, y) \geq k$, the neighborhood $O_n(x, y)$ likely contains both step and roof/valley edge segments. For instance, on two different sides of a step edge segment in $O_n(x, y)$, the image intensity surface often has different slopes. So, the step edge segment is also a roof/valley edge segment. In such cases, the above BID procedure focuses on the detected step edge points only, because step edges would dominate the roof/valley edges in terms of human visual perception about the image. When we make the judgment whether there are step and/or roof/valley edge segments in $O_n(x, y)$, the number $\gamma_n = k$ is used as a threshold in the above procedure, which is roughly half of the number of pixels on a line that passes (x, y) and is parallel to the x- or y-axis in $O_n(x, y)$. If there is a step (or, roof/valley) edge segment in $O_n(x, y)$ and the threshold values u_n and v_n in (3.6) and (3.12) are properly chosen, then the number of detected step edge points (or, detected roof/valley edge points) should

generally be larger than k. Our numerical studies show that other numbers around k (e.g., numbers in $[0.75k, 1.25k]$) can also be chosen as γ_n; but such choices can hardly improve the estimated image. For this reason, we recommend using $\gamma_n = k$, instead of choosing γ_n separately, to save some computation.

3.2.3 Selection of procedure parameters

In the proposed BID procedure, there are several parameters to choose. In the image processing practice, people often choose parameter values to be the ones giving the best visual impression. In this part, we provide an alternative approach, which chooses parameters by data-driven procedures.

Our BID procedure consists of two sequential steps for edge detection and for image estimation. Parameters in these two steps can also be chosen sequentially. In the edge detection procedures (3.6) and (3.12), there are four parameters k_1, u_n, k_2 and v_n to choose. Since detection of step edges is more important to our proposed BID procedure, compared to detection of roof/valley edges, we suggest choosing k_1 and u_n before choosing k_2 and v_n. To this end, we need a performance measure for the set of detected step edges, denoted as $\widehat{S}_n$, in estimating the set of true step edges, denoted as S. In cases when there is no blurring in the observed image Z, Qiu (2002) suggested the following measure:

$$d_Q(\widehat{S}_n, S; k_1, u_n) = w\frac{|\widehat{S}_n \backslash S|}{|\Omega \backslash S|} + (1-w)\frac{|S \backslash \widehat{S}_n|}{|S|},$$

where $0 \le w \le 1$ is a weighting parameter, and $|A|$ denotes the number of design points in the point set A. In practice, D can be replaced by $D^* = \{(x_i, y_j) : d_E((x_i, y_j), D) \le 1/\sqrt{2n}\}$ for the purpose of calculating d_Q, where d_E is the Euclidean distance. Obviously, $d_Q(\widehat{S}_n, S; k_1, u_n)$ is a weighed average of the proportion of false step edge points detected by (3.6) and the proportion of true step edge points missed by (3.6). The weight w represents the relative importance of the first proportion com-

pared to the second proportion, and it should be determined beforehand. In cases when we do not have any prior information about the relative importance of the two proportions, we can simply choose $w = 0.5$.

In simulations, the point set S is usually known. So, k_1 and u_n can be chosen by minimizing $d_Q(\widehat{S}_n, S; k_1, u_n)$. In practice, however, S is often unknown. In such cases, we propose the following bootstrap procedure. Let

$$\{\widehat{\varepsilon}_{ij} = Z_{ij} - \widehat{a}_0(x_i, y_j), \;\; i, j = 1, 2, \ldots, n\}$$

be the set of residuals obtained from the LLK procedure (3.16) with the bandwidth k selected separately via the conventional cross validation procedure. Since the mean function of $Z(x, y)$ is $H\{f\}(x, y)$, which is the blurred version of f and which is a continuous function, these residuals should be reasonable estimates of the random errors $\{\varepsilon_{ij}, i, j = 1, 2, \ldots, n\}$ in model (2.2). Then, we draw n^2 residuals from the above residual set with replacement, and the selected residuals are denoted as $\{\widetilde{\varepsilon}_{ij}^{(1)}, i, j = 1, 2, \ldots, n\}$. The first bootstrap sample is defined by

$$\widetilde{\mathbf{Z}}^{(1)} = \left\{ \widetilde{Z}_{ij}^{(1)} = \widehat{a}_0(x_i, y_j) + \widetilde{\varepsilon}_{ij}^{(1)}, \;\; i, j = 1, 2, \ldots, n \right\}.$$

After repeating this process B times, we get B bootstrap samples $\widetilde{\mathbf{Z}}^{(1)}, \widetilde{\mathbf{Z}}^{(2)}, \ldots, \widetilde{\mathbf{Z}}^{(B)}$, where B is the bootstrap sample size. Assume that the detected sets of step edge points from these bootstrap samples are $\widetilde{S}_n^{(1)}, \widetilde{S}_n^{(2)}, \ldots, \widetilde{S}_n^{(B)}$, respectively. Then, k_1 and u_n can be chosen to be the solution of

$$\min_{k_1, u_n} \frac{1}{B} \sum_{l=1}^{B} d_Q \left(\widetilde{S}_n^{(l)}, \widehat{S}_n; k_1, u_n \right). \tag{3.18}$$

The parameters k_2 and v_n can be chosen similarly. Let RV, $\widehat{RV}_n$, and $\{\widetilde{RV}_n^{(l)}, l = 1, 2, \ldots, B\}$ be the true set of roof/valley edge points, its estimate from the original data, and its estimates from the bootstrap samples, respectively. Then, in cases when

RV is known, k_2 and v_n can be chosen by minimizing $d_Q(\widehat{RV}_n, RV; k_2, v_n)$, which is defined similarly to $d_Q(\widehat{S}_n, S; k_1, u_n)$. In cases when RV is unknown, k_2 and v_n can be chosen by minimizing $\frac{1}{B}\sum_{l=1}^{B} d_Q(\widetilde{RV}_n^{(l)}, \widehat{RV}_n; k_2, v_n)$.

After edge detection, we need to choose the bandwidth k used in the BID procedure (3.14)-(3.17) for defining the deblurred image $\widehat{f}$. In simulation, the true image f is often known, then k can be selected by minimizing $\mathrm{MSE}(f, \widehat{f}; k) = \frac{1}{n^2}\sum_{i,j=1}^{n}(f(x_i, y_j) - \widehat{f}(x_i, y_j))^2$. In practice, however, f is never known. To this end, cross-validation procedures are natural to consider. But, in the BID model (2.1), the mean response is the blurred image $H\{f\}(x, y)$, instead of the true image f. So, the cross-validation idea is appropriate only at places where f is continuous because $H\{f\}(x, y)$ and $f(x, y)$ are close if the f is straight around (x, y). On the other hand, observations at those design points that are around edges are affected much by the blur; thus, choosing k by minimizing the distance between individual observations Z_{ij} and the leave-one-out estimates of $f(x_i, y_j)$ may not be appropriate at those places. For this reason, we suggest using the following bootstrap procedure instead. Let $\{\widehat{f}^{(l)}(x, y), l = 1, 2, \ldots, B\}$ be BID estimates of f constructed from the B bootstrap samples $\{\widetilde{\mathbf{Z}}^{(l)}, l = 1, 2, \ldots, B\}$ defined above. Then, k is approximated by the minimizer of

$$\min_k \frac{1}{B}\sum_{l=1}^{B} \left\{ \frac{\widetilde{w}}{|\widehat{S}_n \cup \widehat{RV}_n|} \sum_{(x_i,y_j)\in \widehat{S}_n \cup \widehat{RV}_n} \left[\widehat{f}^{(l)}(x_i, y_j) - \widehat{f}(x_i, y_j)\right]^2 + \frac{(1-\widetilde{w})}{|\Omega \backslash (\widehat{S}_n \cup \widehat{RV}_n)|} \sum_{(x_i,y_j)\notin \widehat{S}_n \cup \widehat{RV}_n} \left[\widehat{f}^{(l)}_{-(x_i,y_j)}(x_i, y_j) - \widetilde{Z}^{(l)}_{ij}\right]^2 \right\}, \quad (3.19)$$

where $\widehat{f}^{(l)}_{-(x_i,y_j)}(x_i, y_j)$ denotes the leave-one-out estimate at the design point (x_i, y_j) constructed from the bootstrap sample $\widetilde{\mathbf{Z}}^{(l)}$, and $\widetilde{w} \in (0, 1)$ indicates the relative importance of the two quantities. In (3.19), the first quantity reflects the effectiveness of

deblurring around edges while the second one indicates the data fidelity in continuity regions.

When defining $\widehat{f}^{(l)}(x,y)$ from the bootstrap sample $\widetilde{\mathbf{Z}}^{(l)}$, we suggest using the detected step and roof/valley edge points from the original data (i.e., using $\widehat{S}_n$ and $\widehat{RV}_n$), instead of using detected step and roof/valley edge points from $\widetilde{\mathbf{Z}}^{(l)}$, which will save much computation without losing much efficacy of the selected parameter value. Also, we have used the same bootstrap samples as those used in (3.18) for simplicity.

3.3 Statistical Properties

We discuss some statistical properties of the edge detection procedures (3.6) and (3.12) and the BID procedure (3.14)–(3.17) in this section. In the literature on image deblurring, the psf $h(u,v;x,y)$ in model (2.1) is usually 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 would not change the image mass (cf., Bates and McDonnell (1986). This conventional assumption is also adopted here. More specifically, we assume that, for any $(x,y) \in \Omega$, (i) $h(u,v;x,y) \geq 0$, for all $(u,v) \in R^2$, (ii) $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(u,v;x,y)\,dudv = 1$, and (iii) $h(u,v;x,y) = 0$ if $\sqrt{u^2+v^2} > r_n(x,y)/n$, where $r_n(x,y)$ is a positive integer indicating the number of pixels that are affected by the blur at (x,y), i.e. $r_n(x,y)$ is the blurring extent at (x,y). First, we have the following results regarding the detected step and roof/valley edge points by procedures (3.6) and (3.12).

Theorem 3.1

Assume that the true image intensity function f has piece-wisely continuous second-order derivatives in each closed subset of $[0,1]\times[0,1]$ where f and its first-order derivatives are continuous; at the boundary curves of the pieces, f has uniformly

bounded, directional second-order derivatives from any direction in a single piece; f also has uniformly bounded, directional first-order derivatives at any point in RV from any direction passing through a region that the first-order derivatives are continuous; $E(\varepsilon_{11}^3) < \infty$; the kernel function K^* is a Lipschitz-1 continuous, circularly symmetric, density function; the kernel function L^* is a Lipschitz-1 continuous increasing density function supported on $[0,1]$; the psf $h(u,v;x,y)$ is bounded uniformly with respect to (u,v) and (x,y); the blurring extent $R_n = \sup_{i,j} r_n(x_i,y_j)$, the bandwidths k_1, k_2, and the sample size n satisfy the conditions that $k_1/n = o(1)$, $k_2/n = o(1)$, $1/k_1 = o(1)$, $1/k_2 = o(1)$, $R_n/k_1 = o(1)$, $nR_n/k_2^2 = o(1)$, $n^2\log(n)/k_1^3 = o(1)$ and $n^2\log(n)/k_2^3 = o(1)$; u_n and v_n satisfy the conditions that $k_1^2/(nu_n) = o(1)$, $u_n/k_1 = o(1)$, $k_1^2/(nv_n) = o(1)$, and $v_n/k_1 = o(1)$. Then, we have

(i) $d_H\left(\widehat{S}_n \bigcap (\Omega_{k_1,n} \bigcap \overline{J}_{S,k_1,n}),\ S \bigcap (\Omega_{k_1,n} \bigcap \overline{J}_{S,k_1,n})\right) = O(k_1/n)$, *a.s., and*

(ii) $d_H\left(\widehat{RV}_n \bigcap (\Omega_{k_2,n} \bigcap \overline{S}_{k_2,n} \bigcap \overline{J}_{RV,k_2,n}),\ RV \bigcap (\Omega_{k_2,n} \bigcap \overline{S}_{k_2,n} \bigcap \overline{J}_{RV,k_2,n})\right) = O\left(\frac{k_2}{n}\right)$, *a.s.,*

where $\Omega_{k_1,n} = \{(x_i,y_j) : (x_i,y_j) \in [k_1/n, 1-k_1/n] \times [k_1/n, 1-k_1/n]\}$, J_S includes all singular points in S, defined to be crossing points of step edge segments, points on a single step edge segment at which there does not exist a unique tangent line of the edge segment, or points on a single step edge segment at which the jump sizes in f are 0, $J_{S,k_1,n} = \{(x,y) : d_E((x,y),(x',y')) \le k_1/n,$ for any $(x',y') \in J_S\}$, $\overline{J}_{S,k_1,n} = \Omega \backslash J_{k_1,n}$, $S_{k_2,n} = \{(x,y) : d_E((x,y),(x',y')) \le k_2/n,$ for any $(x',y') \in S\}$, $\overline{S}_{k_2,n} = \Omega \backslash S_{k_2,n}$, $\Omega_{k_2,n}$ is defined similarly to $\Omega_{k_1,n}$, J_{RV} includes all singular points on RV defined similarly to J_S, $\overline{J}_{RV,k_2,n}$ is defined similarly to $\overline{J}_{S,k_1,n}$, and

$$d_H(A,B) = \max\left\{\sup_{\mathbf{s}_1 \in A} \inf_{\mathbf{s}_2 \in B} d_E(\mathbf{s}_1,\mathbf{s}_2),\ \sup_{\mathbf{s}_1 \in B} \inf_{\mathbf{s}_2 \in A} d_E(\mathbf{s}_1,\mathbf{s}_2)\right\}$$

is the Hausdorff distance between two point sets A and B. □

By Theorem 3.1, the detected step edge point set $\widehat{S}_n$ converges almost surely to the true step edge point set S in Hausdorff distance, after some small regions around the singular points of S and the border of the design space are excluded. Regarding the detected roof/valley edge point set $\widehat{RV}_n$, Theorem 3.1 says that it converges almost surely to the true roof/valley edge point set RV in Hausdorff distance, after some small regions around the step edge point set S, the singular points of RV and the border of the design space are excluded. Note that in the case when $r_n(x, y)$ does not depend on n, i.e., the number of blurred pixels around (x, y) is a fixed positive integer for any design point (x, y), the conditions on R_n in the theorem are clearly satisfied. The current conditions on R_n in Theorem 3.1 are actually quite flexible in the sense that R_n is allowed to increase to infinity at a certain rate as n increases. The next theorem gives some properties of the deblurred image by the proposed BID procedure (3.14)–(3.17). Proofs of the two theorems are given in Section 3.6.

Theorem 3.2

Under the conditions stated in Theorem 3.1, if we further assume that $k_1/k = o(1)$ *and* $k_2/k = o(1)$*, then we have*

(i) for any given point $(x, y) \in \Omega_{k,n} \backslash (S \bigcup RV)$*,*

$$\widehat{f}(x, y) = f(x, y) + O(k/n) + O\left(\frac{\log(n)}{k}\right), \ a.s.,$$

(ii) for any given point $(x, y) \in \Omega_{k,n} \bigcap (S \setminus J_S)$*,*

$$\widehat{f}(x, y) = f(x, y) + O(k/n) + O\left(k_1/k\right) + O\left(\frac{\log(n)}{k}\right), \ a.s.,$$

(iii) for any given point $(x, y) \in \Omega_{k,n} \bigcap (RV \setminus J_{RV})$*,*

$$\widehat{f}(x, y) = f(x, y) + O(k/n) + O\left(k_2/k\right) + O\left(\frac{\log(n)}{k}\right), \ a.s.,$$

where $\Omega_{k,n}$ *is defined similarly to* $\Omega_{k_1,n}$ *used in Theorem 3.1.* □

3.4 Numerical Examples

In this section, we present some numerical examples concerning the numerical performance of the proposed BID procedure (3.14)–(3.17). Throughout this section, if there is no further specification, the kernel functions K^* and K used in (3.2), (3.7) and (3.14) are both chosen to be $\frac{1}{2\pi-3\pi\exp(-0.5)}[\exp(-(x^2+y^2)/2)-\exp(-0.5)]I_{x^2+y^2\le 1}$, the kernel functions L^* and L used in (3.4), (3.5), (3.9), (3.10) and (3.14) are both chosen to be $\frac{1}{1.194958}\exp(x^2/2)I_{0\le x\le 1}$, w in (3.18) and $\widetilde{w}$ in (3.19) are both fixed at 0.5, and B in (3.18) and (3.19) is chosen to be 100. Degraded images are generated from model (2.2), in which the psf is chosen to be

$$h(u,v;x,y)=\frac{3}{\pi}\left(1-\frac{\sqrt{u^2+v^2}}{r_n(x,y)/n}\right)I_{u^2+v^2\le(r_n(x,y)/n)^2},$$

and the additive random errors ε_{ij} follow the distribution $N(0,\sigma^2)$. The above psf is circularly symmetric with the blurring extent $r_n(x,y)$ which may depend on (x,y). Let $\rho_n(x,y)=r_n(x,y)/n$ denote the blurring-extent-to-sample-size ratio (BSR) at (x,y). We first consider the following true image intensity functions:

$$f_1(x,y) \;=\; \begin{cases} -(x-0.5)^2-(y-0.5)^2+1, & \text{if } (x-0.5)^2+(y-0.5)^2\le 0.25^2,\\ -(x-0.5)^2-(y-0.5)^2, & \text{otherwise.}\end{cases}$$

It is shown in Figure 3.3 by a 3-D plot, from which it can be seen that f_1 has one circular step edge and it does not have any roof/valley edges. Figure 3.4(a) shows the original true image of f_1, and Figure 3.4(b) shows an observed image when $\sigma=0.1$, $\rho_n^{(1)}(x,y)=0.03(1-(x-0.5)^2-(y-0.5)^2)+0.02$, and $n=100$. The detected step edges are shown in Figure 3.4(c). Finally, the deblurred image is displayed in Figure 3.4(d). The parameters k_1, u_n, and k are selected to be $8, 4.9$ and 4, respectively, in this example. It can be seen from the figure that, in the deblurred image, the noise has been mostly removed and the spatial blur has also been significantly reduced, which

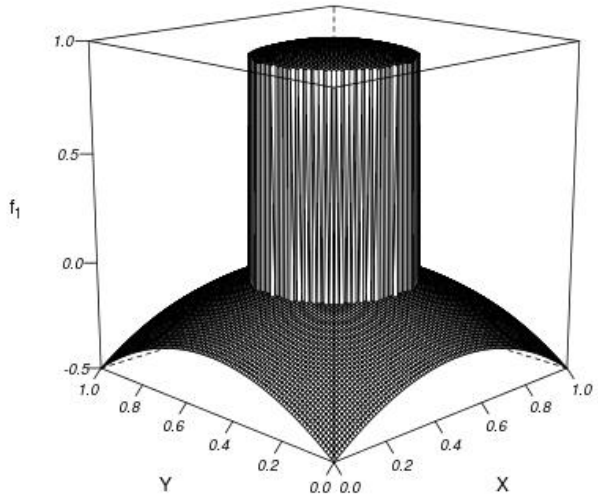

Figure 3.3: A 3-D plot of $f_1(x, y)$.

confirms our theoretical justification discussed in the previous section.

Next, the performance of our proposed method is measured quantitatively. We consider two BSR functions $\rho_n^{(1)}(x, y) = 0.03(1 - (x - 0.5)^2 - (y - 0.5)^2) + 0.02$ and $\rho_n^{(2)}(x, y) = 0.05x$, two sample sizes $n = 100$ and 200, and three σ values 0.05, 0.1 and 0.2. Simulation results based on 100 replications are presented in Table 3.1. In the simulation, for each combination of $\rho_n(x, y)$, n, and σ, the parameters (k_1, u_n) and k are chosen sequentially by minimizing the averaged values of $d_Q(\widehat{S}_n, S; k_1, u_n)$ and $\text{MSE}(\widehat{f}, f) = \frac{1}{n^2}\sum_{i=1}^n \sum_{i=1}^n (\widehat{f}(x_i, y_j) - f(x_i, y_j))^2$, respectively. Such parameter values are called optimal ones hereafter. Parameters chosen by our proposed bootstrap procedures (3.18) and (3.19) are also presented in the table. From the table, it can be seen that (i) $\text{MSE}(\widehat{f}, f)$ increases as σ increases, and decreases as n increases, and (ii) parameters chosen via our proposed bootstrap procedure are quite close to their optimal ones, and this is true especially when the sample size gets large, with the optimal values of k_1 and u_n slightly smaller than their values chosen by the bootstrap

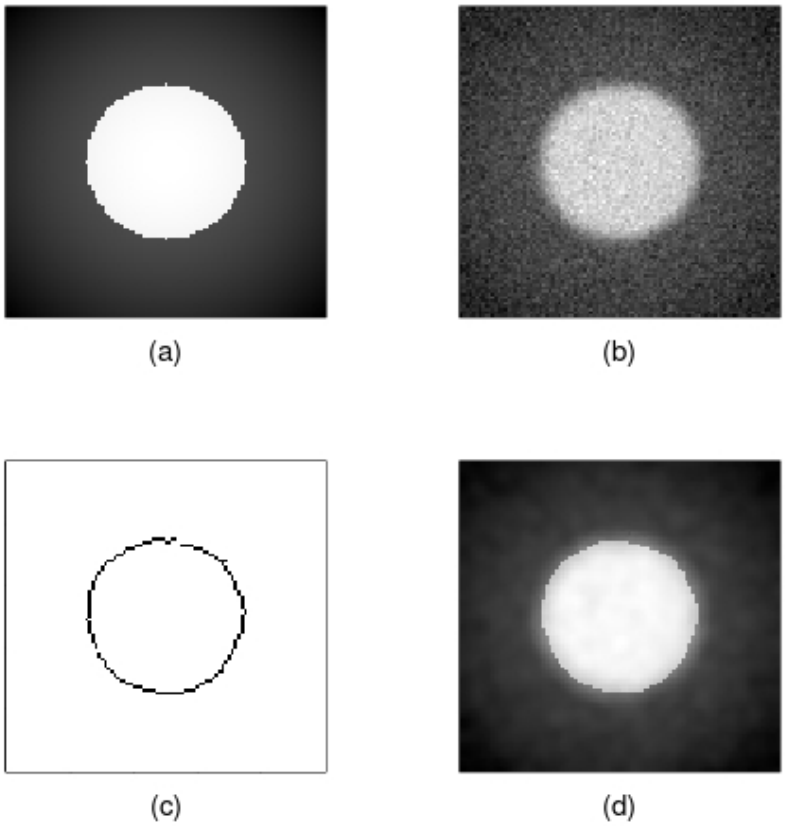

Figure 3.4: (a): True image; (b): Observed image; (c): Detected step edges;(d): Deblurred image.

Table 3.1: Simulation results of the BID procedure (3.14)–(3.17) in the example of Figure 3.4 based on 100 replications. In each entry, the first line presents the optimal values of k_1/n, u_n and k/n, the second line presents their values chosen by the proposed bootstrap procedure with $B = 100$, the third line presents the value of $d_Q(\widehat{S}_n, S; k_1, u_n)$, and the fourth line presents the value of $\mathrm{MSE}(\widehat{f}, f)$.

		$\rho_n^{(1)}(x,y)$	
n	$\sigma = .05$	$\sigma = .1$	$\sigma = .2$
100	(0.07, 6.2, 0.04)	(0.08, 4.9, 0.04)	(0.09, 3.7, 0.05)
	(0.08, 7.8, 0.02)	(0.10, 8.6, 0.03)	(0.12, 4.8, 0.08)
	1.07×10^{-2}	1.37×10^{-2}	1.62×10^{-2}
	5.44×10^{-3}	6.23×10^{-3}	8.44×10^{-3}
200	(0.06, 8.6, 0.04)	(0.08, 9.5, 0.035)	(0.105, 10.0, 0.035)
	(0.06, 8.5, 0.02)	(0.09, 9.8, 0.02)	(0.11, 10.2, 0.03)
	0.785×10^{-2}	0.980×10^{-2}	1.12×10^{-2}
	5.33×10^{-3}	6.02×10^{-3}	7.54×10^{-3}
		$\rho_n^{(2)}(x,y)$	
n	$\sigma = .05$	$\sigma = .1$	$\sigma = .2$
100	(0.08, 13.1, 0.02)	(0.08, 6.5, 0.03)	(0.09, 4.7, 0.04)
	(0.10, 16.1, 0.07)	(0.10, 8.0, 0.08)	(0.11, 5.7, 0.08)
	1.15×10^{-2}	1.03×10^{-2}	1.15×10^{-2}
	3.99×10^{-3}	4.72×10^{-3}	6.05×10^{-3}
200	(0.065, 19.5, 0.03)	(0.065, 9.6, 0.03)	(0.07, 5.9, 0.03)
	(0.065, 20.7, 0.03)	(0.075, 9.2, 0.055)	(0.08, 5.1, 0.06)
	0.73×10^{-2}	0.84×10^{-2}	0.88×10^{-2}
	3.32×10^{-3}	3.56×10^{-3}	4.14×10^{-3}

procedure.

Next, we consider another example with the following true image intensity function:

$$f_2(x,y) = \begin{cases} 0, & \text{if } x \le 0.5 \text{ and } y > 0.5 \\ 1, & \text{if } x > 0.5 \text{ and } y > 0.5 \\ 3, & \text{if } x > 0.5 \text{ and } y \le 0.5 \\ -1, & \text{if } x \le 0.25 \text{ and } y \le 0.5 \\ 16(x - 0.25) - 1, & \text{if } 0.25 < x \le 0.5 \text{ and } y \le 0.5. \end{cases}$$

The surface of f_2 is shown in Figure 3.5(a), from which it can be seen that f_2 has several step edge segments and two roof/valley edge segments at $(x = 0.25, y \leq 0.5)$ and $(x = 0.5, y \leq 0.5)$. Figure 3.5(b) shows a 3-D plot of an observed surface in the case when $\rho_n^{(2)}(x, y) = 0.05x$, $\sigma = 0.2$, and $n = 100$. Figure 3.5(c) shows the deblurred surface by our BID procedure, using the corresponding parameter values presented in Table 3.2. The observed surface of f_2 is then shown as an image in Figure 3.5(d). The detected edge segments are shown in Figure 3.5(e), in which step edges are shown in black and roof/valley edges are shown in gray. Finally, the deblurred image by our proposed BID procedure is shown in Figure 3.5(f). From the figure, it can be seen that (i) spatial blur gets severer as x gets larger in the observed image, (ii) the pointwise noise and the spatial blur are well removed in the deblurred image by the BID procedure (3.14)–(3.17), (iii) both step edges and roof/valley edges have been detected successfully except at places around certain singular points (cf., Section Section 3.3 for their definition).

Some numerical results in a similar setup to that of Table 3.1 are presented in Table 3.2. From the table, it can be seen that (i) values of $d_Q(\widehat{S}_n, S)$, $d_Q(\widehat{RV}_n, RV)$ and $\mathrm{MSE}(\widehat{f}, f)$ increase as the noisy level σ increases, and decrease as the sample size n increases, (ii) the value of the bandwidth k/n chosen by the bootstrap procedure (3.19) is close to its optimal value, and (iii) as the observed image gets noisier (i.e., σ is larger), the bandwidths k_1, k_2, k and k^B should generally be chosen larger, which is intuitively reasonable because more observations should be used in local smoothing to remove noise in such cases.

At the end of this section, we consider the test image of peppers with 256×256 pixels shown in Figure 4.5, and compare our proposed BID procedure with three representatives of the existing deblurring methods. The first existing method considered here is the one accomplished by the Matlab's blind deconvolution routine *deconvblind*, which is based on the method discussed by Biggs and Andrews (1997) and Jansson

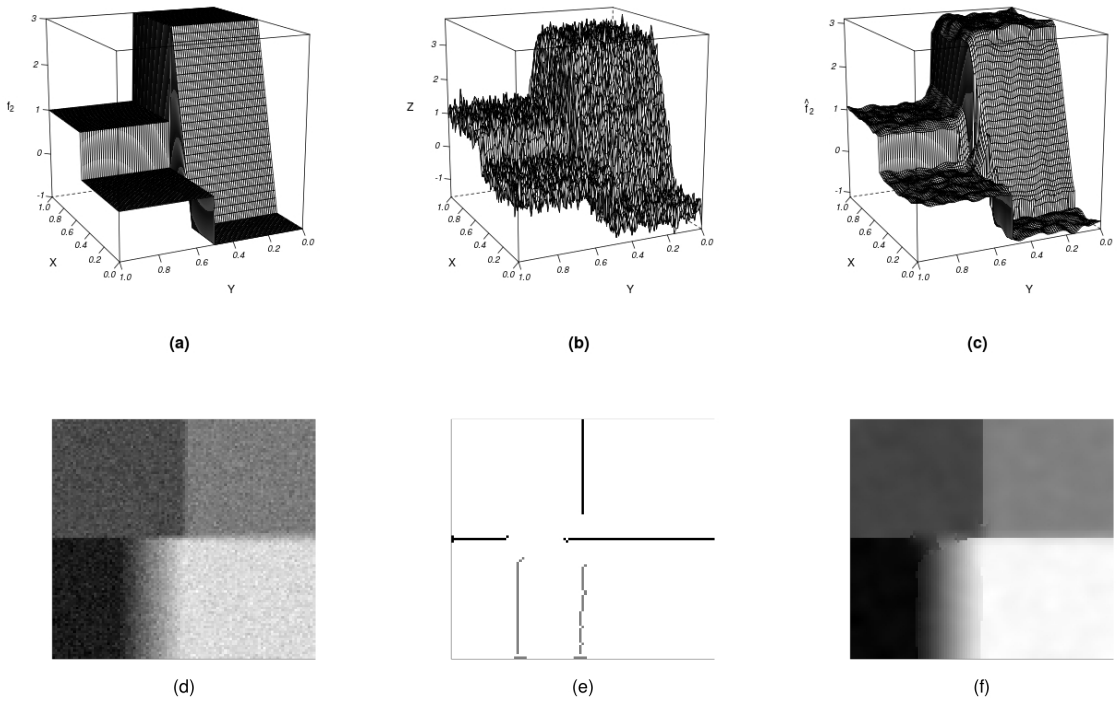

Figure 3.5: (a)-(c): 3-D plots of the true surface, observed surface, and deblurred surface of $f_2(x, y)$; (d)-(f): Observed image, detected step edge segments (black lines) and detected roof/valley edge segments (gray lines), and deblurred image.

Table 3.2: Simulation results of the BID procedure (3.14)–(3.17) in the example of Figure 3.5 based on 100 replications. In each entry, the first line presents the optimal values of k_1/n and u_n, the second line presents the d_Q value of the detected step edges, the third line presents the optimal values of k_2/n and v_n, the fourth line present the d_Q value of the detected roof/valley edges, the fifth line presents the values of k/n and k^B/n, and the sixth line presents the value of MSE$(\widehat{f}, f)$.

	$\rho_n^{(1)}(x,y)$		
n	$\sigma = .1$	$\sigma = .2$	$\sigma = .3$
100	(0.13, 11.1)	(0.18, 13.1)	(0.18, 9.0)
	5.085×10^{-3}	8.103×10^{-3}	9.591×10^{-3}
	(0.09, 4.3)	(0.10, 2.8)	(0.11, 2.9)
	4.935×10^{-3}	8.282×10^{-3}	9.998×10^{-3}
	(0.04, 0.04)	(0.05, 0.06)	(0.05, 0.06)
	9.199×10^{-3}	11.21×10^{-3}	13.01×10^{-3}
200	(0.11, 18.0)	(0.13, 10.7)	(0.17, 16.0)
	3.124×10^{-3}	4.357×10^{-3}	4.546×10^{-3}
	(0.085, 7.8)	(0.105, 7.5)	(0.105, 5.0)
	2.668×10^{-3}	5.958×10^{-3}	6.815×10^{-3}
	(0.045, 0.025)	(0.035, 0.035)	(0.05, 0.045)
	9.181×10^{-3}	9.349×10^{-3}	10.53×10^{-3}

	$\rho_n^{(2)}(x,y)$		
n	$\sigma = .1$	$\sigma = .2$	$\sigma = .3$
100	(0.11, 15.2)	(0.12, 9.0)	(0.12, 6.1)
	3.143×10^{-3}	4.340×10^{-3}	6.026×10^{-3}
	(0.10, 5.9)	(0.10, 3.0)	(0.10, 2.0)
	3.581×10^{-3}	6.815×10^{-3}	9.193×10^{-3}
	(0.04, 0.04)	(0.05, 0.06)	(0.05, 0.08)
	4.852×10^{-3}	6.689×10^{-3}	7.989×10^{-3}
200	(0.085, 23.0)	(0.115, 16.2)	(0.115, 10.9)
	1.589×10^{-3}	2.375×10^{-3}	2.913×10^{-3}
	(0.095, 11.1)	(0.105, 8.0)	(0.11, 6.0)
	1.713×10^{-3}	3.453×10^{-3}	4.234×10^{-3}
	(0.045, 0.045)	(0.05, 0.055)	(0.05, 0.055)
	4.255×10^{-3}	5.721×10^{-3}	6.118×10^{-3}

Figure 3.6: Original test image of peppers.

(1997) in the framework of the Richardson-Lucy algorithm. The second method is the total-variation-based image deblurring method proposed by Oliveira et al. (2009). The third method is the blind image deconvolution procedure developed under the Bayesian framework by Fergus et al. (2006). These methods are denoted as RL, TV and Bayes, respectively. It should be pointed out that both RL and Bayes are blind image deblurring schemes, but TV requires the specification of a parametric form for the psf h.

The example is set up as follows. The original test image is first blurred by the psf h used in the previous examples with three different BSR functions: $\widetilde{\rho}_n^{(1)}(x,y) = 0.03(1-(x-0.5)^2-(y-0.5)^2)$, $\widetilde{\rho}_n^{(2)}(x,y) = 0.03x$, and $\widetilde{\rho}^{(3)}(x,y) = 0.02$. Then, additive random noise generated from the $N(0,\sigma^2)$ distribution is added to the blurred test image, where σ is fixed at 5 or 10. Figure 3.7(a) presents the degraded test image in the case with $\widetilde{\rho}_n^{(2)}$ and $\sigma = 10$. The deblurred image by our BID method with the bandwidth k/n chosen to be $5/256$ is shown in Figure 3.7(b). The deblurred image by RL is shown in Figure 3.7(c). Note that the RL algorithm is designed to handle location-invariant blur only, and the blurring extent r_n needs to be specified beforehand. In our simulation, r_n is selected by minimizing the mean squared error of

the deblurred image so that it can have its best performance in the example. The TV procedure requires the specification of the form of h beforehand. In this example, we have tried two different ways for that purpose. One is that the true expression of h is used, and the second way is that h is assumed to be the psf of a motion blur along the x-axis (i.e., $h(u,v;x,y) = \frac{2n}{r_n} I_{\{|u| \le r_n/n\}} \delta_0(v)$, where δ_0 is a point mass at 0). In both cases, the blurring extent r_n is chosen by minimizing the mean square errors of the deblurred image. The two different versions of the TV procedure are denoted as TV1 and TV2 hereafter, and their deblurred images are shown in Figure 3.7(d)-(e). To implement the Bayes method, a sub-region of the image with a known edge structure needs to be chosen before. Based on our visual impression of the test image, we choose the sub-region $[25/256, 75/256] \times [25/256, 75/256]$, and the deblurred image is shown in Figure 3.7(f). From the figure, it can be seen that (i) our proposed method can remove both noise and blur reasonably well, (ii) the deblurred image by RL contains much noise and some artifacts, (iii) the TV1 method, in which the form of the psf h is correctly specified, cannot handle spatially-variant blur well, in that much artifacts are generated at the boundaries of the image while its deblurred image in the central part looks reasonably good, (iv) the TV2 method, in which a wrong psf is specified, performs poorly, and (v) the Bayes method performs poorly as well due to the facts that the observed test image contains much noise and that the method is developed mainly for handling motion blur which is not the case in the current example. We have tried several different choices of the sub-region when implementing the Bayes method. No significant improvement in the results can be obtained.

Figure 3.8 presents the corresponding results in cases when $\widetilde{\rho}_n^{(1)}(x,y) = 0.02$ and $\sigma = 10$. In such cases, the spatial blur is location invariant. From the figure, it can be seen that (i) our proposed method sharpens the observed test image much, (ii) the deblurred image by RL contains much noise and many artifacts as before, (iii) the TV method performs well when the true psf h is fully specified, (iv) the TV method

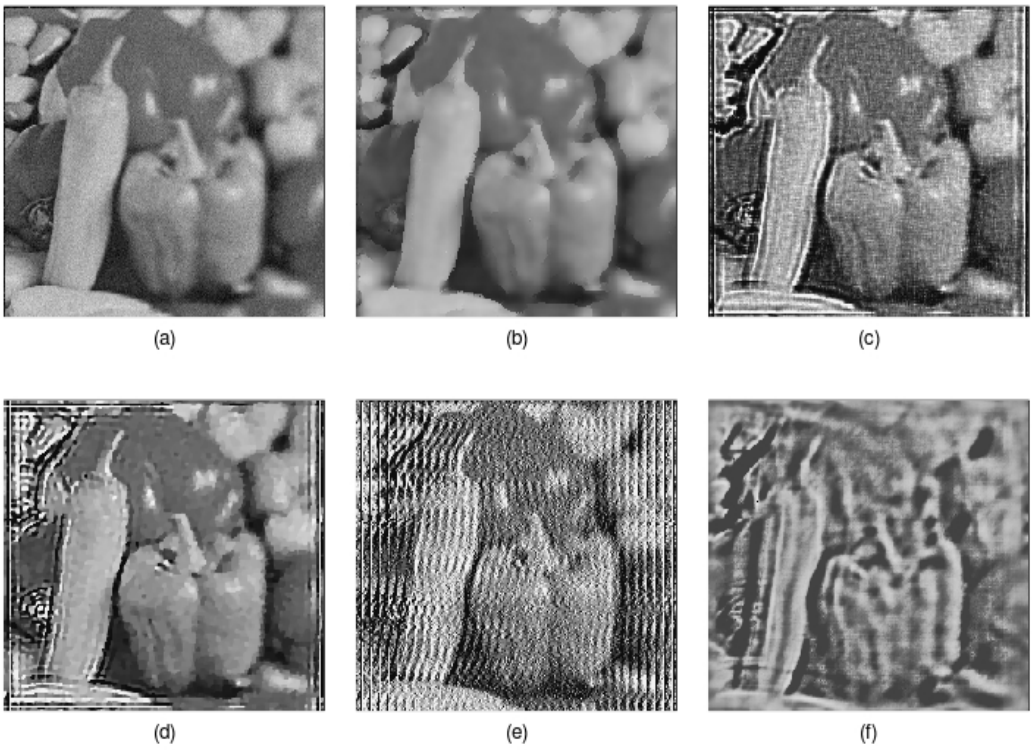

Figure 3.7: (a): Observed test image of peppers in the case when $\rho_n(x, y) = 0.03x$ and $\sigma = 10$; (b): Deblurred image by our proposed method; (c): Deblurred image by RL; (d): Deblurred image by TV when the true psf h is specified correctly; (e): Deblurred image by TV when the psf h is specified incorrectly; (f): Deblurred image by Bayes.

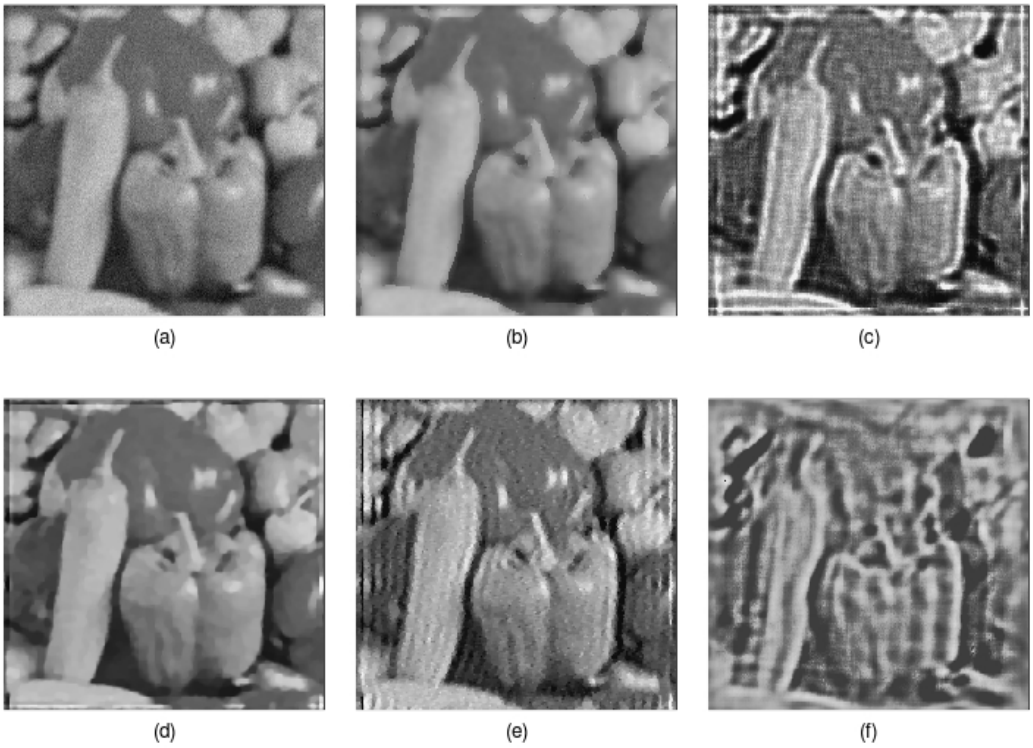

Figure 3.8: (a): Observed test image of peppers in the case when $\rho_n(x, y) = 0.02$ and $\sigma = 10$; (b): Deblurred image by our proposed method; (c): Deblurred image by RL; (d): Deblurred image by TV when the true psf h is specified correctly; (e): Deblurred image by TV when the psf h is specified incorrectly; (f): Deblurred image by Bayes.

does not perform well when h is misspecified, and (v) the Bayes method performs poorly once again.

Table 3.3 presents the root mean squared errors (RMSE) of the four deblurring methods, based on 100 replicated simulations. From the table, it can be seen that (i) the proposed method outperforms the other methods in most cases considered, and it can handle a wide variety of blurs since it does not require any restrictive conditions on the psf h, (ii) the TV method performs the best in cases when the true psf is location-invariant and fully specified (i.e., the case when $\rho_n(x, y) = \widetilde{\rho}_n^{(3)}(x, y)$ for TV1), (iii) the TV method could perform poorly when h is misspecified (i.e., TV2), and (iv) both the RL method and the Bayes method cannot deblur the image well in

Table 3.3: RMSE values of the four image deblurring methods in the example of the test image of peppers.

Methods	$\widetilde{\rho}_n^{(1)}(x,y)$		$\widetilde{\rho}_n^{(2)}(x,y)$		$\widetilde{\rho}_n^{(3)}(x,y)$	
	$\sigma = 5$	$\sigma = 10$	$\sigma = 5$	$\sigma = 10$	$\sigma = 5$	$\sigma = 10$
Proposed method	19.18	19.37	18.14	19.34	18.07	18.40
RL	30.32	33.66	49.18	52.58	37.90	40.14
TV1	19.40	19.64	64.79	64.59	17.66	17.75
TV2	27.71	36.84	76.27	103.29	20.73	29.36
Bayes	33.78	45.06	42.44	52.81	46.48	48.99

this example.

3.5 Discussions

We have described our proposed method for blind image deblurring in the previous sections. Our method differs from most existing methods by imposing little restriction on the psf h and the true image intensity function. It even allows the psf h to be spatially variant. Our method makes use of the hierarchical structure of blurred images by paying special attention to regions around the detected step and valley/roof edges. Also, a data-driven parameter selection scheme based on bootstrap has been suggested. Both theoretical justifications and numerical studies show that our method can remove pointwise noise and spatial blur well in various cases.

The proposed method still has much room for further improvements. For instance, the current version of the method uses constant bandwidth and threshold parameters in step edge and roof/valley edge detection and in image deblurring as well. The idea of multilevel smoothing with location-variant bandwidths and threshold values can be incorporated into the proposed method. In such cases, one price to pay is the expensive computation. Furthermore, our current method cannot provide an estimate for the psf h. It requires much future research to modify it properly such that h can

be estimated accurately at the same time when the observed image is deblurred.

3.6 Technical Details

Lemma 3.1

Under the conditions of Theorem Theorem 3.1, the estimated gradient $(\widehat{b}(x,y), \widehat{c}(x,y))$ *obtained from the local linear kernel smoothing procedure (3.2) has the following properties:*

(i) If f *has continuous first order derivatives at* (x,y)*, then*

$$(\widehat{b}(x,y), \widehat{c}(x,y)) \to (f'_x(x,y), f'_y(x,y)), \ \ a.s, \qquad as\ n \to \infty. \tag{3.20}$$

(ii) If (x,y) *is a non-singular point on a roof/valley edge, i.e.,* f *is continuous at* (x,y) *and has finite first order directional derivatives but the limits of* f'_x *or* f'_y *from the two parts separated by the roof/valley edge are not the same, then*

$$\begin{aligned}(\widehat{b}(x,y), \widehat{c}(x,y)) \ \ \to \ \ & \left(\frac{f'_{x+}(x,y) + f'_{x-}(x,y)}{2}, \frac{f'_{y+}(x,y) + f'_{y-}(x,y)}{2} \right), \\ & a.s, \qquad as\ n \to \infty,\end{aligned} \tag{3.21}$$

where $f'_{x+}(x,y)$*,* $f'_{x-}(x,y)$*,* $f'_{y+}(x,y)$*, and* $f'_{y-}(x,y)$ *denote the limits of the first order derivatives of* $f(u,v)$ *as* (u,v) *approaches* (x,y) *from the two parts separated by the roof/valley edge.*

(iii) If (x,y) *is a non-singular point on a step edge which has a tangent line at* (x,y)*, then*

$$\frac{(\widehat{b}(x,y), \widehat{c}(x,y))}{\sqrt{\widehat{b}(x,y)^2 + \widehat{c}(x,y)^2}} \to (-\sin\theta, \cos\theta), \ \ a.s. \ , \qquad as\ n \to \infty, \tag{3.22}$$

where θ is the angle formed by the tangent line of the JLC at (x, y) and the x-axis. □

Proof 3.1 (Lemma 3.1)

Recall that $(x_i, y_j) = (i/n, j/n)$, for $i, j = 1, 2, \ldots, n$, and $R_n = \sup_{i,j} r_n(i/n, j/n)$, where $r_n(i/n, j/n)$ is the blurring extent at $(i/n, j/n)$. Then, it is not difficult to verify that the solution of (3.2) has the expressions

$$\widehat{b}(x,y) = \frac{1}{r^*_{20}} \sum_{i^2+j^2 \le k_1^2} \frac{i}{n} Z\left(x + \frac{i}{n}, y + \frac{j}{n}\right) K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right), \tag{3.23}$$

$$\widehat{c}(x,y) = \frac{1}{r^*_{02}} \sum_{i^2+j^2 \le k_1^2} \frac{j}{n} Z\left(x + \frac{i}{n}, y + \frac{j}{n}\right) K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right), \tag{3.24}$$

where $r^*_{s_1 s_2} = \sum_{i^2+j^2 \le k_1^2} (\frac{i}{n})^{s_1} (\frac{j}{n})^{s_2} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right)$, for $s_1, s_2 = 0, 1, 2$. To prove the result (3.20), we notice that if (x, y) is a continuity point, then

$$\mathrm{E}(\widehat{b}(x,y)) = \frac{1}{r^*_{20}} \sum_{i^2+j^2 \le k_1^2} H\{f\}\left(x + \frac{i}{n}, y + \frac{j}{n}\right) \frac{i}{n} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right), \tag{3.25}$$

where

$$\begin{aligned} H\{f\}(\xi_i, \gamma_j) &= \int\int_{u^2+v^2 \le (\frac{r_n(\xi_i,\gamma_j)}{n})^2} h(u, v; \xi_i, \gamma_j) f(\xi_i - u, \gamma_j - v)\, du dv \\ &= \int\int_{u^2+v^2 \le (\frac{r_n(\xi_i,\gamma_j)}{n})^2} h(u, v; \xi_i, \gamma_j) \left[f(\xi_i, \gamma_j) - f'_x(\xi_i, \gamma_j) u \right. \\ &- \left. f'_y(\xi_i, \gamma_j) v + o(\frac{r_n(\xi_i, \gamma_j)}{n}) \right] du dv \\ &= f(\xi_i, \gamma_j) + O(R_n/n), \end{aligned} \tag{3.26}$$

and $(\xi_i, \gamma_j) = \left(x + \frac{i}{n}, y + \frac{j}{n}\right)$. By (3.25) and (3.26), we have

$$\mathrm{E}(\widehat{b}(x,y))$$

$$
\begin{aligned}
&= \frac{1}{r_{20}^*} \sum_{i^2+j^2\le k_1^2} [f(x+i/n, y+j/n) + O(R_n/n)] \frac{i}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) \\
&= \frac{1}{r_{20}^*} \sum_{i^2+j^2\le k_1^2} [f(x,y) + f'_x(x,y)i/n + f'_y(x,y)j/n + O(k_1^2/n^2)] \frac{i}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) \\
&\quad + \frac{1}{r_{20}^*} \sum_{i^2+j^2\le k_1^2} O(R_n/n) \frac{i}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) \\
&= \frac{f(x,y)}{r_{20}^*} \sum_{i^2+j^2\le k_1^2} \frac{i}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) + f'_x(x,y) + \\
&\quad \frac{f'_y(x,y)}{r_{20}^*} \sum_{i^2+j^2\le k_1^2} \frac{ij}{n^2} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) + \frac{O(R_n/n) + O(k_1^2/n^2)}{k_1/n} \\
&= f'_x(x,y) + O(R_n/k_1) + O(k_1/n). \qquad (3.27)
\end{aligned}
$$

In the last equation of the above expression, we have used the results that $\sum_{i^2+j^2\le k_1^2} \frac{i}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) = 0$, $\sum_{i^2+j^2\le k_1^2} \frac{ij}{n^2} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) = 0$, by the circular symmetry of K^*. We have also used the result that $r_{20}^* = O(k_1^4/n^2)$, which can be proved similarly to expression (23) in Proposition 2 of Qiu (2009). Then by (24) in Proposition 2 of Qiu (2009), we have

$$
\frac{1}{k_1^2} \sum_{i^2+j^2\le k_1^2} \varepsilon_{ij} \phi \left(\frac{i}{k_1}, \frac{j}{k_1}\right) K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) = O\left(\frac{\log n}{k_1}\right), \ a.s., \qquad (3.28)
$$

where $\phi(u,v)$ is any Lipschitz-1 continuous function defined in the region $\{(u,v) : u^2+v^2 \le 1\}$. By (3.25) and the fact that $r_{20}^* = O(k_1^4/n^2)$, we have

$$
\widehat{b}(x,y) - \mathrm{E}(\widehat{b}(x,y)) = \frac{1}{r_{20}^*} \sum_{i^2+j^2\le k_1^2} \varepsilon_{ij} \frac{i}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1}\right) = O\left(\frac{n \log(n)}{k'^2}\right), \ a.s. \quad (3.29)
$$

Similarly, we have

$$\widehat{c}(x,y) - \mathrm{E}(\widehat{c}(x,y)) = \frac{1}{r_{02}^*} \sum_{i^2+j^2 \le k_1^2} \varepsilon_{ij} \frac{j}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1} \right) = O\left(\frac{n \log(n)}{k'^2} \right), \ a.s. \quad (3.30)$$

After combining (3.29) and (3.30), (3.20) is proved.

Now, assume that (x, y) is a non-singular point on a roof/valley edge segment. Since f has bounded directional first-order derivatives, we can find a positive constant C such that for any two points $(x_i, y_j)$and $(x_i - u, y_j - v)$ in Ω, we have

$$f(x_i, y_j) - C\sqrt{u^2+v^2} \le f(x_i - u, y_j - v) \le f(x_i, y_j) + C\sqrt{u^2+v^2}.$$

Consequently, $f(x_i, y_j) - C\frac{r_n(x_i,y_j)}{n} \le \int\int_{u^2+v^2 \le (\frac{r(x_i,y_j)}{n})^2} h(u, v; x_i, y_j) f(x_i - u, y_j - v)\, dudv \le f(x_i, y_j) + C\frac{r_n(x_i,y_j)}{n}$. So,

$$H\{f\}(x_i, y_j) = f(x_i, y_j) + O(r_n(x_i, y_j)/n). \quad (3.31)$$

Because (x, y) is a non-singular point, the related roof/valley edge segment has a unique tangent line at (x, y). Without loss of generality, let us assume that (i) the roof/valley edge segment in $O'_n(x, y)$ is a straight line, which forms an angle θ with the x-axis, and (ii) the roof/valley edge segment divides $O'_n(x, y)$ into two parts $O'_{1n}(x, y)$ and $O'_{2n}(x, y)$, where $O'_{1n}(x, y)$ contains the lower-left quarter of $O'_n(x, y)$ and $O'_{2n}(x, y)$ contains the upper-right quarter of $O'_n(x, y)$. The first assumption is reasonable because the difference between the roof/valley edge segment and the tangent line at (x, y) is negligible in $O'_n(x, y)$ when n is sufficiently large. Then, we have

$$\begin{aligned} & \mathrm{E}(\widehat{b}(x,y)) \\ = \ & \frac{1}{r_{20}^*} \left(\sum_{O'_{1n}(x,y)} + \sum_{O'_{2n}(x,y)} \right) [f(x + i/n, y + j/n) + O(R_n/n)] \frac{i}{n} K^* \left(\frac{i}{k_1}, \frac{j}{k_1} \right) \end{aligned}$$

$$
\begin{aligned}
= & \frac{1}{r_{20}^*}\sum_{O'_{1n}(x,y)}\left[f(x,y)+f'_{x+}(x,y)\frac{i}{n}+f'_{y+}(x,y)\frac{j}{n}\right]\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
& \frac{1}{r_{20}^*}\sum_{O'_{2n}(x,y)}\left[f(x,y)+\ f'_{x-}(x,y)\frac{i}{n}+f'_{y-}(x,y)\frac{j}{n}\right]\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
& \frac{1}{r_{20}^*}\sum_{O'_{n}(x,y)}\left[O(R_n/n)+O(k_1^2/n^2)\right]\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)\\
= & \frac{f(x,y)}{r_{20}^*}\sum_{O'_{n}(x,y)}\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\frac{f'_{x+}(x,y)}{r_{20}^*}\sum_{O'_{1n}(x,y)}\left(\frac{i}{n}\right)^2K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
& \frac{f'_{x-}(x,y)}{r_{20}^*}\sum_{O'_{2n}(x,y)}\left(\frac{i}{n}\right)^2K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\frac{f'_{y+}(x,y)}{r_{20}^*}\sum_{O'_{1n}(x,y)}\frac{i}{n}\frac{j}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
& \frac{f'_{y-}(x,y)}{r_{20}^*}\sum_{O'_{2n}(x,y)}\frac{i}{n}\frac{j}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+O(R_n/k_1)+O(k_1/n)\\
= & \left(\frac{f'_{x+}(x,y)}{r_{20}^*}\sum_{O'_{1n}(x,y)}+\frac{f'_{x-}(x,y)}{r_{20}^*}\sum_{O'_{2n}(x,y)}\right)\left(\frac{i}{n}\right)^2K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
& \left(\frac{f'_{y+}(x,y)}{r_{20}^*}\sum_{O'_{1n}(x,y)}+\frac{f'_{y-}(x,y)}{r_{20}^*}\sum_{O'_{2n}(x,y)}\right)\frac{i}{n}\frac{j}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
& O(R_n/k_1)+O(k_1/n),
\end{aligned}
$$

where we have used the result that $r_{10}^*=0$ due to the circular symmetry of K^*. Also observe the following facts:

$$
\begin{aligned}
\frac{\sum_{O'_{1n}(x,y)}(i/n)^2K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)}{r_{20}^*} &= \frac{\int_\theta^{\theta+\pi}\,d\varphi\int_0^1 r^3\cos^2\varphi\widetilde{K}^*(r)\,dr}{\int_0^{2\pi}\,d\varphi\int_0^1 r^3\cos^2\varphi\widetilde{K}^*(r)\,dr}+O(1/k_1)\\
&= \frac{1}{2}+O(1/k_1),\\
\frac{\sum_{O'_{2n}(x,y)}(i/n)^2K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)}{r_{20}^*} &= \frac{\int_{\theta+\pi}^{\theta+2\pi}\,d\varphi\int_0^1 r^3\cos^2\varphi\widetilde{K}^*(r)\,dr}{\int_0^{2\pi}\,d\varphi\int_0^1 r^3\cos^2\varphi\widetilde{K}^*(r)\,dr}+O(1/k_1)\\
&= \frac{1}{2}+O(1/k_1),
\end{aligned}
$$

$$\begin{aligned}
\frac{\sum_{O'_{1n}(x,y)}(i/n)(j/n)K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)}{r^*_{20}} &= \frac{\int_\theta^{\theta+\pi}\, d\varphi \int_0^1 r^3\cos\varphi\sin\varphi\widetilde{K}^*(r)\, dr}{\int_0^{2\pi}\, d\varphi \int_0^1 r^3\cos^2\varphi\widetilde{K}^*(r)\, dr} + O(1/k_1) \\
&= 0 + O(1/k_1), \\
\frac{\sum_{O'_{2n}(x,y)}(i/n)(j/n)K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)}{r^*_{20}} &= \frac{\int_{\theta+\pi}^{\theta+2\pi}\, d\varphi \int_0^1 r^3\cos\varphi\sin\varphi\widetilde{K}^*(r)\, dr}{\int_0^{2\pi}\, d\varphi \int_0^1 r^3\cos^2\varphi\widetilde{K}^*(r)\, dr} + O(1/k_1) \\
&= 0 + O(1/k_1),
\end{aligned}$$

where $\widetilde{K}^*(r) = K^*(r\cos\varphi, r\sin\varphi)$. Then we have

$$\mathrm{E}(\widehat{b}(x,y)) = \frac{f'_{x-}(x,y) + f'_{x+}(x,y)}{2} + O(1/k_1) + O(R_n/k_1) + O(k_1/n). \tag{3.32}$$

By (3.29) and (3.32), we have

$$\widehat{b}(x,y) = \frac{f'_{x-}(x,y) + f'_{x+}(x,y)}{2} + O(1/k_1) + O(R_n/k_1) + O(k_1/n) + O\left(\frac{n\log(n)}{k_1^2}\right).$$

Similarly, we have

$$\widehat{c}(x,y) = \frac{f'_{y-}(x,y) + f'_{y+}(x,y)}{2} + O(1/k_1) + O(R_n/k_1) + O(k_1/n) + O\left(\frac{n\log(n)}{k_1^2}\right).$$

Then, (3.21) is proved.

Now, if (x,y) is a non-singular point on a step edge segment, then $O'_n(x,y)$ consists of the following three disjoint parts $O'_{n,l}(x,y)$, $O'_{n,c}(x,y)$, and $O'_{n,r}(x,y)$, where $O'_{n,c}(x,y)$ is a band of width $2R_n/n$ containing a step edge segment in its middle, and $O'_{n,l}(x,y)$ and $O'_{n,r}(x,y)$ are on its two different sides. Since (x,y) is non-singular, the step edge segment has a unique tangent line at (x,y). Also, without loss of generality, we can assume that the step edge segment is a straight line in $O'_n(x,y)$ and it forms an angle θ with the x-axis. Then, we have

$$\begin{aligned}
&\mathrm{E}(\widehat{b}(x,y)) \\
&= \frac{1}{r^*_{20}}\left(\sum_{O'_{n,l}(x,y)} + \sum_{O'_{n,c}(x,y)} + \sum_{O'_{n,r}(x,y)}\right) H\{f\}(x+i/n, y+j/n)\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{r_{20}^*}\sum_{O'_{n,l}(x,y)}[f(x+i/n,y+j/n)+O(R_n/n)]\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
&\quad \frac{1}{r_{20}^*}\sum_{O'_{n,c}(x,y)}H\{f\}(x+i/n,y+j/n)\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
&\quad \frac{1}{r_{20}^*}\sum_{O'_{n,r}(x,y)}[f(x+i/n,y+j/n)+O(R_n/n)]\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)\\
&= \frac{1}{r_{20}^*}\sum_{O'_{n,l}(x,y)}[f_-(x,y)+O(k_1/n)+O(R_n/n)]\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+O\left(\frac{nR_n}{k_1^2}\right)+\\
&\quad \frac{1}{r_{20}^*}\sum_{O'_{n,r}(x,y)}[f_+(x,y)+O(k_1/n)+O(R_n/n)]\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)\\
&= \frac{1}{r_{20}^*}f_-(x,y)\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)-\frac{1}{r_{20}^*}f_-(x,y)\sum_{O'_{n,r}(x,y)}\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)-\\
&\quad \frac{1}{r_{20}^*}f_-(x,y)\sum_{O'_{n,c}(x,y)}\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\frac{1}{r_{20}^*}f_+(x,y)\sum_{O'_{n,r}(x,y)}\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+\\
&\quad O(R_n/k_1)+O(1)+O\left(\frac{nR_n}{k_1^2}\right)\\
&= \frac{f_+(x,y)-f_-(x,y)}{r_{20}^*}\sum_{O'_{n,r}(x,y)}\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+O(1)+O\left(\frac{nR_n}{k_1^2}\right). \qquad (3.33)
\end{aligned}
$$

In the third equation of (3.33), we have used the results that $r_{20}^* = O(n^2/(k_1)^4)$, $H\{f\}(x_i, y_j)$ are uniformly bounded when $(x_i, y_j) \in O'_{n,c}(x, y)$, and the fact that the ratio of the area of $O'_{n,c}(x, y)$ to the area of $O'_n(x, y)$ is of order $O(R_n/k_1)$. In the fourth equation, we have used the results that $\sum_{O'_{n,r}(x,y)} \frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right) = O(k_1^3/n)$, $\sum_{O'_{n,l}(x,y)} \frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right) = O(k_1^3/n)$, and $r_{20}^* = O(k_1^4/n^2)$. In the last equation, we have used the result $r_{10}^* = 0$ and $\frac{1}{r_{20}^*}\sum_{O'_{n,c}(x,y)} \frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right) = O\left(\frac{nR_n}{k_1^2}\right)$. By (3.29), we have

$$
\widehat{b}(x,y) = \frac{f_+(x,y)-f_-(x,y)}{r_{20}^*}\sum_{O'_{n,r}(x,y)}\frac{i}{n}K^*\left(\frac{i}{k_1},\frac{j}{k_1}\right)+
$$

$$O(1) + O\left(\frac{nR_n}{(k_1)^2}\right) + O\left(\frac{n\log(n)}{k_1^2}\right),\ a.s. \tag{3.34}$$

Similarly, we have

$$\begin{aligned}\widehat{c}(x,y) &= \frac{f_+(x,y) - f_-(x,y)}{r_{02}^*} \sum_{O'_{n,r}(x,y)} \frac{j}{n} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) + \\ & O(1) + O\left(\frac{nR_n}{k_1^2}\right) + O\left(\frac{n\log(n)}{k_1^2}\right),\ a.s. \end{aligned} \tag{3.35}$$

By using the following two facts:

$$\frac{k_1/n}{r_{20}^*} \sum_{O'_{n,r}(x,y)} \frac{i}{n} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) \to \frac{\int_\theta^{\theta+\pi}\ d\varphi \int_0^1 r^2 \cos\varphi K^*(r)\ dr}{\int_0^{2\pi}\ d\varphi \int_0^1 r^3 \cos^2\varphi K^*(r)\ dr} = \frac{-2\int_0^1 r^2 K^*(r)\ dr}{\pi \int_0^1 r^3 K^*(r)\ dr} \sin\theta, \tag{3.36}$$

$$\frac{k_1/n}{r_{02}^*} \sum_{O'_{n,r}(x,y)} \frac{j}{n} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) \to \frac{\int_\theta^{\theta+\pi}\ d\varphi \int_0^1 r^2 \sin\varphi K^*(r)\ dr}{\int_0^{2\pi}\ d\varphi \int_0^1 r^3 \sin^2\varphi K^*(r)\ dr} = \frac{2\int_0^1 r^2 K^*(r)\ dr}{\pi \int_0^1 r^3 K^*(r)\ dr} \cos\theta, \tag{3.37}$$

we have,

$$\begin{aligned}\frac{(\widehat{b}(x,y), \widehat{c}(x,y))}{\sqrt{\widehat{b}(x,y)^2 + \widehat{c}(x,y)^2}} &= \frac{((k_1/n)\widehat{b}(x,y), (k_1/n)\widehat{c}(x,y))}{\sqrt{(k_1/n)^2\widehat{b}(x,y)^2 + (k_1/n)^2\widehat{c}(x,y)^2}} \\ &\to (-\sin\theta, \cos\theta),\ a.s, \end{aligned}$$

which completes the proof of (3.22). □

Proof 3.2 (Theorem 3.1)

By some routine algebraic manipulations, the solution to a_0 in the local linear kernel

smoothing problem (3.4) has the expression

$$\begin{aligned}\widehat{f}_{LL2K,+}(x,y) &= \frac{\sum_{U'_n(x,y)} b_{ij}(x,y)Z_{ij}}{\sum_{U'_n(x,y)} b_{ij}(x,y)} \\ &= \frac{\sum_{U'_n(x,y)} b_{ij}(x,y)H\{f\}(x_i,y_j)}{\sum_{U'_n(x,y)} b_{ij}(x,y)} + \frac{\sum_{U'_n(x,y)} b_{ij}(x,y)\varepsilon_{ij}}{\sum_{U'_n(x,y)} b_{ij}(x,y)} \\ &=: I_1(x,y) + I_2(x,y). \end{aligned} \tag{3.38}$$

Let $\widetilde{U}'_n(x,y)$ denote the half of $O'_n(x,y)$ separated by a line passing (x,y) in the direction perpendicular to the asymptotic direction of $(\widehat{b}(x,y),\widehat{c}(x,y))$, which is given in Lemma 3.1, and $\widetilde{d}'_{ij}$ denote the Euclidean distance from (x_i,y_j) to that dividing line. For a function ϕ satisfying the condition that $\sup_{u^2+v^2\le 1}|\phi(u,v)| \le b_\phi < \infty$, we have

$$\begin{aligned} & \left| \sum_{U'_n(x,y)} \phi(\frac{i}{k_1},\frac{j}{k_1})K^*(\frac{i}{k_1},\frac{j}{k_1})L^*(nd'_{ij}/k_1)\frac{1}{k_1^2} - \right. \\ & \left. \sum_{\widetilde{U}'_n(x,y)} \phi(\frac{i}{k_1},\frac{j}{k_1})K^*(\frac{i}{k_1},\frac{j}{k_1})L^*(n\widetilde{d}'_{ij}/k_1)\frac{1}{k_1^2} \right| \\ \le\ & \frac{1}{k_1^2}\left| \sum_{U'_n(x,y)} \phi(\frac{i}{k_1},\frac{j}{k_1})K^*(\frac{i}{k_1},\frac{j}{k_1})L^*(n\widetilde{d}'_{ij}/k_1) - \right. \\ & \left. \sum_{\widetilde{U}'_n(x,y)} \phi(\frac{i}{k_1},\frac{j}{k_1})K^*(\frac{i}{k_1},\frac{j}{k_1})L^*(n\widetilde{d}'_{ij}/k_1) \right| + O\left(\frac{|d'_{ij}-\widetilde{d}'_{ij}|}{k_1/n}\right) \\ \le\ & b_\phi \|K\|_\infty \|L\|_\infty \left| \frac{1}{k_1^2} \sum_{U'_n(x,y)\triangle\widetilde{U}'_n(x,y)} 1 \right| + O\left(\frac{|d'_{ij}-\widetilde{d}'_{ij}|}{k_1/n}\right) \\ =\ & O(\theta_n) = o(1),\ a.s. \end{aligned} \tag{3.39}$$

where θ_n denotes the acute angle between $(\widehat{b}(x,y),\widehat{c}(x,y))$ and its asymptotic direction and $U'_n(x,y) \triangle \widetilde{U}'_n(x,y) = (U'_n(x,y) \setminus \widetilde{U}'_n(x,y)) \cup (\widetilde{U}'_n(x,y) \setminus U'_n(x,y))$. In the

first inequality of (3.39), we have used the Lipschitz-1 continuity of L^*. In the last equation, Lemma 3.1 has been applied. Now, let

$$\begin{aligned}
\widetilde{b}_{i,j}(x,y) &= \left[\widetilde{B}_1(x,y) + \widetilde{B}_2(x,y)(x_i - x) + \widetilde{B}_3(x,y)(y_j - y)\right] \cdot \\
&\quad K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*(n\widetilde{d}'_{ij}/k_1), \\
\widetilde{B}_1(x,y) &= \widetilde{t}_{20}(x,y)\widetilde{t}_{02}(x,y) - \widetilde{t}_{11}(x,y)\widetilde{t}_{11}(x,y), \\
\widetilde{B}_2(x,y) &= \widetilde{t}_{01}(x,y)\widetilde{t}_{11}(x,y) - \widetilde{t}_{10}(x,y)\widetilde{t}_{02}(x,y), \\
\widetilde{B}_3(x,y) &= \widetilde{t}_{10}(x,y)\widetilde{t}_{11}(x,y) - \widetilde{t}_{01}(x,y)\widetilde{t}_{20}(x,y), \\
\widetilde{t}_{s_1,s_2}(x,y) &= \sum_{\widetilde{U}'_n(x,y)} (i/n)^{s_1}(j/n)^{s_2} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*(n\widetilde{d}'_{ij}/k_1).
\end{aligned}$$

Then, by using similar arguments to those in (3.39), we can check that

$$I_1(x,y) = \frac{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y) H\{f\}(x_i, y_j)}{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y)} + O(\theta_n), \ a.s. \tag{3.40}$$

By using (3.39) and some similar arguments to those in the proof of Proposition 2 in Qiu (2009), we can prove that

$$\begin{aligned}
I_2(x,y) &= \sum_{U'_n(x,y)} \frac{b_{ij}(x,y)\frac{n^4}{k_1^8}}{\frac{n^4}{k_1^{10}}\sum_{U'_n(x,y)} b_{ij}(x,y)} \frac{1}{k_1^2}\varepsilon_{ij} \\
&= \sum_{U'_n(x,y)} \frac{\frac{n^4}{k_1^8}\widetilde{b}_{ij}(x,y) + O(\theta_n)}{\frac{n^4}{k_1^{10}}\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y) + O(\theta_n)} \frac{1}{k_1^2}\varepsilon_{ij} \\
&= \sum_{U'_n(x,y)} \left(\frac{\frac{n^4}{k_1^8}\widetilde{b}_{ij}(x,y)}{\frac{n^4}{k_1^{10}}\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y)} + O(\theta_n)\right) \frac{1}{k_1^2}\varepsilon_{ij} \\
&= \sum_{U'_n(x,y)} \frac{\widetilde{b}_{ij}(x,y)}{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y)}\varepsilon_{ij} + \frac{1}{k_1^2}\sum_{U'_n(x,y)} O(\theta_n)\varepsilon_{ij} \\
&= O\left(\frac{\log(n)}{k_1}\right) + O(\theta_n)
\end{aligned}$$

$$= O\left(\frac{\log(n)}{k_1}\right) + O(\theta_n),\ a.s. \tag{3.41}$$

In the second equation of (3.41), we have used (3.39) and the results that $\widetilde{B}_1(x,y) = O(k_1^8/n^4)$, $\widetilde{B}_2(x,y) = O(k_1^7/n^3)$, $\widetilde{B}_3(x,y) = O(k_1^7/n^3)$, and $\widetilde{t}_{s_1,s_2}(x,y) = O\left(\frac{k_1^{s_1+s_2+2}}{n^{s_1+s_2}}\right)$, for $s_1, s_2 = 0, 1$. The fifth equation is a direct conclusion of Proposition 2 in Qiu (2009), since $\widetilde{b}_{ij}(x,y)$ is deterministic. Now, for any given point (x,y) such that $d_E((x,y), S) > k_1/n$, $O'_n(x,y)$ does not contain any step edge. By (3.26) and (3.31), we have

$$\begin{aligned}
& \frac{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y) H\{f\}(x_i, y_j)}{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y)} \\
= & \frac{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y)(f(x+i/n, y+j/n) + O(R_n/n))}{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y)} \\
= & \frac{\widetilde{B}_1(x,y)}{|\widetilde{\triangle}|} \sum_{\widetilde{U}'_n(x,y)} (f(x,y) + O(k_1/n) + O(R_n/n)) K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*(n\widetilde{d}'_{ij}/k_1) + \\
& \frac{\widetilde{B}_2(x,y)}{|\widetilde{\triangle}|} \sum_{\widetilde{U}'_n(x,y)} (f(x,y) + O(k_1/n) + O(R_n/n)) \frac{i}{n} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*(n\widetilde{d}'_{ij}/k_1) + \\
& \frac{\widetilde{B}_3(x,y)}{|\widetilde{\triangle}|} \sum_{\widetilde{U}'_n(x,y)} (f(x,y) + O(k_1/n) + O(R_n/n)) \frac{j}{n} K^*\left(\frac{i}{k_1}, \frac{j}{k_1}\right) L^*(n\widetilde{d}'_{ij}/k_1) \\
= & f(x,y) + O(k_1/n) + O(R_n/n),
\end{aligned} \tag{3.42}$$

where $|\widetilde{\triangle}| = \widetilde{t}_{00}(x,y)\widetilde{t}_{20}(x,y)\widetilde{t}_{02}(x,y) + 2\widetilde{t}_{10}(x,y)\widetilde{t}_{01}(x,y)\widetilde{t}_{11}(x,y) - \widetilde{t}_{01}(x,y)^2\widetilde{t}_{20}(x,y) - \widetilde{t}_{11}(x,y)^2\widetilde{t}_{00}(x,y) - \widetilde{t}_{10}(x,y)^2\widetilde{t}_{02}(x,y)$. In the second equation of (3.42), we have used (3.31) and the fact f has uniformly bounded directional derivatives. In the last equation, we have used the results that $|\widetilde{\triangle}| = \widetilde{B}_1(x,y)\widetilde{t}_{00}(x,y) + \widetilde{B}_2(x,y)\widetilde{t}_{10}(x,y) + \widetilde{B}_3(x,y)\widetilde{t}_{01}(x,y)$ and that $\widetilde{B}_1(x,y) = O(k_1^8/n^4)$, $\widetilde{B}_2(x,y) = O(k_1^7/n^3)$, $\widetilde{B}_3(x,y) = O(k_1^7/n^3)$, $|\widetilde{\triangle}| = O(k_1^{10}/n^4)$, $\widetilde{t}_{s_1,s_2}(x,y) = O\left(\frac{k_1^{s_1+s_2+2}}{n^{s_1+s_2}}\right)$, for $s_1, s_2 = 0, 1$. All these results can be proved similarly to the result (23) in Proposition 2 of Qiu (2009).

After combining (3.38), (3.40), (3.41) and (3.42), we have

$$\widehat{f}_{LL2K,+}(x,y) = f(x,y) + O(k_1/n) + O(R_n/n) + O\left(\frac{\log(n)}{k_1}\right) + O(\theta_n),\ a.s. \quad (3.43)$$

Similarly, we have

$$\widehat{f}_{LL2K,-}(x,y) = f(x,y) + O(k_1/n) + O(R_n/n) + O\left(\frac{\log(n)}{k_1}\right) + O(\theta_n),\ a.s. \quad (3.44)$$

From the proof of (3.20) and (3.21) in Lemma 3.1 , we know that

$$\theta_n = O(R_n/k_1) + O(k_1/n) + O\left(\frac{n\log(n)}{k_1^2}\right),\ a.s. \quad (3.45)$$

Thus,

$$\widehat{f}_{LL2K,+}(x,y) - \widehat{f}_{LL2K,-}(x,y) = O(R_n/k_1) + O(k_1/n) + O\left(\frac{n\log(n)}{k_1^2}\right),\ a.s. \quad (3.46)$$

Also, by using similar arguments to those in (3.39) and the fact that $\widetilde{b}_{ij}(x,y) = O(k_1^{16}/n^8)$, we have

$$\begin{aligned}
\frac{\sum_{U_n'(x,y)} b_{ij}^2(x,y)}{[\sum_{U_n'(x,y)} b_{ij}(x,y)]^2} &= \frac{\frac{k_1^{18}}{n^8}\frac{n^8}{k_1^{18}}\sum_{U_n'(x,y)} b_{ij}^2(x,y)}{\left[\frac{k_1^{10}}{n^4}\frac{n^4}{k_1^{10}}\sum_{U_n'(x,y)} b_{ij}(x,y)\right]^2} \\
&= \frac{\frac{k_1^{18}}{n^8}\left(\frac{n^8}{k_1^{18}}\sum_{\widetilde{U}_n'(x,y)} \widetilde{b}_{ij}^2(x,y) + O(\theta_n)\right)}{\left[\frac{k_1^{10}}{n^4}\left(\frac{n^4}{k_1^{10}}\sum_{\widetilde{U}_n'(x,y)} \widetilde{b}_{ij}(x,y) + O(\theta_n)\right)\right]^2} \\
&= \frac{1}{k_1^2}\frac{\frac{n^8}{k_1^{18}}\sum_{\widetilde{U}_n'(x,y)} \widetilde{b}_{ij}^2(x,y) + O(\theta_n)}{\left[\frac{n^4}{k_1^{10}}\sum_{\widetilde{U}_n'(x,y)} \widetilde{b}_{ij}(x,y) + O(\theta_n)\right]^2} \\
&= \frac{1}{k_1^2}\left\{\frac{\frac{n^8}{k_1^{18}}\sum_{\widetilde{U}_n'(x,y)} \widetilde{b}_{ij}^2(x,y)}{\left[\frac{n^4}{k_1^{10}}\sum_{\widetilde{U}_n'(x,y)} \widetilde{b}_{ij}(x,y)\right]^2} + O(\theta_n)\right\},\ a.s.
\end{aligned}$$

Then, it follows that

$$\sqrt{\frac{\sum_{U'_n(x,y)} b_{ij}^2(x,y)}{[\sum_{U'_n(x,y)} b_{ij}(x,y)]^2} + \frac{\sum_{V'_n(x,y)} b_{ij}^2(x,y)}{[\sum_{V'_n(x,y)} b_{ij}(x,y)]^2}} = O\left(\frac{1}{k_1}\right), \ a.s. \tag{3.47}$$

Hence, by (3.46) and (3.47), we have

$$\frac{\mathrm{M}_n^{(1)}(x,y)}{u_n} = O(R_n/u_n) + O\left(\frac{k_1^2}{u_n n}\right) + O\left(\frac{n\log(n)}{k' u_n}\right), \ a.s. \tag{3.48}$$

Now, consider any given non-singular point (x, y) on a step edge. Then, the related step edge has a unique tangent line at (x, y). Also, without loss of generality, we can assume that the step edge is a straight line in $O'_n(x, y)$. Suppose the step edge separates $O'_n(x, y)$ into two halves, denoted by $O'_{n,1}(x, y)$ and $O'_{n,2}(x, y)$, respectively. Then, it follows from (3.22) in Lemma 3.1 that $\widetilde{U}'_n(x, y) = O'_{n,1}(x, y)$. By the same arguments as those in (3.42), we have

$$\frac{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y) H\{f\}(x_i, y_j)}{\sum_{\widetilde{U}'_n(x,y)} \widetilde{b}_{ij}(x,y)} = f_+(x,y) + O(k_1/n) + O(R_n/n), \tag{3.49}$$

where $f_+(x, y)$ denotes the limit of $f(u, v)$ as (u, v) approaching (x, y) from $\widetilde{U}'_n(x, y)$. Similar to (3.43) and (3.44), we have

$$\widehat{f}_{LL2K,+}(x,y) = f_+(x,y) + O(R_n/k_1) + O(k_1/n) + O\left(\frac{n\log(n)}{k_1^2}\right), \ a.s. \tag{3.50}$$

$$\widehat{f}_{LL2K,+}(x,y) = f_-(x,y) + O(R_n/k_1) + O(k_1/n) + O\left(\frac{n\log(n)}{k_1^2}\right), \ a.s. \tag{3.51}$$

Hence,

$$\frac{\mathrm{M}_n^{(1)}(x,y)}{u_n} \ = \ O\left(\frac{k_1(f_+(x,y) - f_-(x,y))}{u_n}\right) + O(R_n/u_n)$$

$$+ \quad O\left(\frac{k_1^2}{u_n n}\right) + O\left(\frac{n\log(n)}{k' u_n}\right), \ a.s. \tag{3.52}$$

where $f_-(x,y)$ is defined similarly to $f_+(x,y)$. It follows from (3.48) and (3.52) that the proposed step edge detection procedure (3.6) could detect all points in $S \cap \Omega_{k_1,n} \cap \overline{J}_{S,k_1,n}$, and all points whose Euclidean distances to S are greater than k_1/n would not be detected. So, when n is sufficiently large, $S \cap \Omega_{k_1,n} \cap \overline{J}_{S,k,n}$ is included in $\widehat{S}_n$, and $\widehat{S}_n$ is included in the band of S with width k_1/n. Thus, the result (i) in Theorem 3.1 is proved. For roof/valley edge detection, results parallel to Lemma 3.1, (3.48) and (3.52) can be derived in a similar way. Therefore, the result (ii) in Theorem 3.1 is also valid. □

Proof 3.3 (Theorem 3.2)

For a point $(x,y) \in \Omega_{k,n} \backslash (S \bigcup RV)$, by Theorem 3.1, we know that, when n is large enough, there would be no detected step or roof/valley edge points in $O_n(x,y)$. Therefore, $\widehat{f}(x,y)$ is defined by (3.17) in such cases, and

$$\begin{aligned}
\mathrm{E}(\widehat{f}(x,y)) &= \frac{\sum_{i^2+j^2\le k^2} w_{ij}(x,y) H\{f\}(x+i/n, y+j/n)}{\sum_{i^2+j^2\le k^2} w_{ij}(x,y)} \\
&= \frac{\sum_{i^2+j^2\le k^2} w_{ij}(x,y)[f(x+i/n, y+j/n) + O(R_n/n)]}{\sum_{i^2+j^2\le k^2} w_{ij}(x,y)} \\
&= \frac{\sum_{i^2+j^2\le k^2} w_{ij}(x,y)[f(x,y) + O(k/n) + O(R_n/n)]}{\sum_{i^2+j^2\le k^2} w_{ij}(x,y)} \\
&= f(x,y) + O(k/n) + O(R_n/n).
\end{aligned} \tag{3.53}$$

In the second equation of (3.53), the result (3.26) has been used. In the third equation, we have used the property that f has continuous first-order derivatives in $O_n(x,y)$. Therefore, for any $(x_i, y_j) \in O_n(x,y)$, there is a constant $C_1 > 0$ that $|f(x_i,y_j) - f(x,y)| \le C_1 k/n$. On the other hand, by similar results to (3.28) and by the fact

that $r_{s_1 s_2} = O(k^{s_1+s_2+2}/n^{s_1+s_2})$, for $s_1, s_2 = 0, 1, 2$, we have

$$\begin{aligned}
& \widehat{f}(x,y) - \mathrm{E}(\widehat{f}(x,y)) \\
= & \frac{\sum_{i^2+j^2\le k^2} w_{ij}(x,y)\varepsilon_{ij}}{\sum_{i^2+j^2\le k^2} w_{ij}(x,y)} \\
= & \frac{A_1(x,y)\sum_{i^2+j^2\le k^2}\varepsilon_{ij}K\left(\frac{i}{k},\frac{j}{k}\right) + A_2(x,y)\sum_{i^2+j^2\le k^2}\varepsilon_{ij}(i/n)K\left(\frac{i}{k},\frac{j}{k}\right)}{A_1(x,y)r_{00} + A_2(x,y)r_{10} + A_3(x,y)r_{01}} + \\
& \frac{A_3(x,y)\sum_{i^2+j^2\le k^2}\varepsilon_{ij}(j/n)K\left(\frac{i}{k},\frac{j}{k}\right)}{A_1(x,y)r_{00} + A_2(x,y)r_{10} + A_3(x,y)r_{01}} \\
= & O\left(\frac{\log(n)}{k}\right), \ a.s. \qquad (3.54)
\end{aligned}$$

In the last equation, we have used the facts that $A_1(x,y) = r_{20}(x,y)r_{02}(x,y) - r_{11}(x,y)r_{11}(x,y) = O(k^8/n^4)$, $A_2(x,y) = O(k^7/n^3)$, and $A_3(x,y) = O(k^7/n^3)$. Then, the result (i) of the theorem follows from (3.53) and (3.54).

Now, let us consider a point $(x,y) \in S\backslash J_S$. In such cases, from Theorem 3.1, we know that all design points in $S \bigcap O_n(x,y)$ would be detected as step edge points by the procedure (3.3) and all design points outside $S_{k_1,n}(x,y) = S_{k_1,n} \bigcap O_n(x,y)$ would not be detected. Because (x,y) is not a singular point, S has a tangent line at (x,y). Without loss of generality, we assume that S is a straight line with slope $\gamma \neq \infty$. Then, all detected step edge points in $O_n(x,y)$ (i.e., $\{(w_l, v_l), l = 1, 2, \ldots, m\}$) have the expression

$$v_l - y = \gamma(w_l - x) + O(k_1/n), a.s., \text{ for } l = 1, 2, \ldots, m. \qquad (3.55)$$

So, by the results (B.1) and (B.2) in Qiu (1998), it is easy to check that $\sigma_{wv} = \gamma\sigma_{ww} + O(k_1/n)$, $\sigma_{vv} = \gamma^2\sigma_{ww} + O(k_1/n)$, and the slope of the fitted PC line is $\sigma_{wv}/(\sigma_{ww} + \lambda_1) = \gamma + O(k_1/n), a.s.$. By (3.55), we also know that $(\bar{w}, \bar{v})$ converges to (x,y) almost surely with the rate $O(k_1/n)$. Let $\widetilde{U}_n(x,y)$ be the part of $O_n(x,y)$ that is separated by S and contains (x,y), then $U_n(x,y) \setminus S_{k_1,n} = \widetilde{U}_n(x,y) \setminus S_{k_1,n}$ since

the fitted PC line would be asymptotically contained in $S_{k_1,n}$. And, without loss of generality, we can assume that $f(x,y)$ equals the limit of $f(u,v)$ as (u,v) approaches (x,y) from $\widetilde{U}_n(x,y)$. Let $\widetilde{d}_{ij}$ be the Euclidean distance from (x_i,y_j) to S. Then, by (3.15), we have

$$
\begin{aligned}
& \mathrm{E}(\widehat{f}(x,y)) \\
= & \frac{\sum_{U_n(x,y)} \widetilde{w}_{ij}(x,y) H\{f\}(x+i/n, y+j/n)}{\sum_{U_n(x,y)} \widetilde{w}_{ij}(x,y)} \\
= & \frac{\left(\sum_{U_n(x,y)\setminus S_{k_1,n}(x,y)} + \sum_{U_n(x,y)\bigcap S_{k_1,n}(x,y)}\right) \widetilde{w}_{ij}(x,y) H\{f\}(x+i/n, y+j/n)}{\left(\sum_{U_n(x,y)\setminus S_{k_1,n}(x,y)} + \sum_{U_n(x,y)\bigcap S_{k_1,n}(x,y)}\right) \widetilde{w}_{ij}(x,y)} \\
= & \frac{\sum_{U_n(x,y)\setminus S_{k_1,n}(x,y)} \widetilde{w}_{ij}(x,y) H\{f\}(x+i/n, y+j/n)}{\sum_{U_n(x,y)\setminus S_{k_1,n}(x,y)} \widetilde{w}_{ij}(x,y)} + O\left(\frac{k_1}{k}\right) \\
= & \frac{\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} \widetilde{w}_{ij}(x,y) H\{f\}(x+i/n, y+j/n)}{\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} \widetilde{w}_{ij}(x,y)} + O\left(\frac{k_1}{k}\right) \\
= & \frac{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right)\left(L\left(\frac{\widetilde{d}_{ij}}{k/n}\right)+O\left(\frac{k_1}{k}\right)\right) H\{f\}(x+i/n, y+j/n)}{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right)\left(L\left(\frac{\widetilde{d}_{ij}}{k/n}\right)+O\left(\frac{k_1}{k}\right)\right)} + \\
& O\left(\frac{k_1}{k}\right) \\
= & \frac{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right) H\{f\}(x+i/n, y+j/n)}{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right)} + O\left(\frac{k_1}{k}\right) \\
= & \frac{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right) [f(x,y)+(k/n)+O(R_n/n)]}{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right)} + O\left(\frac{k_1}{k}\right) \\
= & f(x,y) + O(k/n) + O(R_n/n) + O\left(\frac{k_1}{k}\right) \\
= & f(x,y) + O(k/n) + O\left(\frac{k_1}{k}\right). \qquad (3.56)
\end{aligned}
$$

In the third equation of (3.56), we have used the results that

$$\mathcal{T}(U_n(x,y)\bigcap S_{k_1,n}(x,y))/\mathcal{T}(O_n(x,y)) = O(k_1/k) \qquad (3.57)$$

and that $H\{f\}(x_i, y_j)$ is uniformly bounded, where $\mathcal{T}(O_n(x,y))$ denotes the area of $O_n(x,y)$. In the fifth equation of (3.56), we have used the Lipschitz-1 continuity of L and the fact that both S and the fitted PC line are contained in $S_{k_1,n}$. In the seventh equation, results (3.26) and (3.31) have been used. Now, by similar arguments to those in (3.41), we can check that

$$
\begin{aligned}
& \widehat{f}(x,y) - \mathrm{E}(\widehat{f}(x,y)) \\
= & \frac{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)\varepsilon_{ij}}{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} \\
= & \frac{\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)\varepsilon_{ij}}{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} + \frac{\sum_{U_n(x,y)\bigcap S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)\varepsilon_{ij}}{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} \\
= & \frac{\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right)\varepsilon_{ij}}{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} + \frac{\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) O\left(\frac{k_1}{k}\right)\varepsilon_{ij}}{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} \\
+ & \frac{\sum_{U_n(x,y)\bigcap S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)\varepsilon_{ij}}{\sum_{U_n(x,y)} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} \\
= & \frac{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right)\varepsilon_{ij}}{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right) + O\left(\frac{k_1}{k}\right) + \frac{1}{k^2}\sum_{U_n(x,y)\bigcap S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} \\
+ & \frac{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) O\left(\frac{k_1}{k}\right)\varepsilon_{ij}}{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right) + O\left(\frac{k_1}{k}\right) + \frac{1}{k^2}\sum_{U_n(x,y)\bigcap S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} \\
+ & \frac{\frac{1}{k^2}\sum_{U_n(x,y)\bigcap S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)\varepsilon_{ij}}{\frac{1}{k^2}\sum_{\widetilde{U}_n(x,y)\setminus S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{\widetilde{d}_{ij}}{k/n}\right) + O\left(\frac{k_1}{k}\right) + \frac{1}{k^2}\sum_{U_n(x,y)\bigcap S_{k_1,n}} K\left(\frac{i}{k},\frac{j}{k}\right) L\left(\frac{d_{ij}}{k/n}\right)} \\
= & O\left(\frac{\log(n)}{k}\right) + O\left(\frac{k_1}{k}\right) + O\left(\frac{k_1}{k}\right). \; a.s. \qquad (3.58)
\end{aligned}
$$

In the second equation of (3.58), we have used the fact that $U_n(x,y) \setminus S_{k_1,n} = \widetilde{U}_n(x,y) \setminus S_{k_1,n}$. In the third and fourth equations, the Lipschitz-1 continuity of L has been used. In the fifth equation, we have used (3.57) and the Proposition 2 in Qiu

(2009).

Finally, we consider the case when $(x, y) \in RV \backslash J_{RV}$. From the above arguments, we can check that (3.56) and (3.58) still hold after we replace the detected step edge points by the detected roof/valley edge points. Therefore, the results (ii) and (iii) of the theorem are valid. □

Chapter 4

Efficient Blind Image Deblurring Using Nonparametric Regression and Local Pixel Clustering

4.1 Introduction

The BID procedure based on edge detection introduced in the previous chapter aims to remove blur around step and roof/valley edges by detecting the possible edge pixels first and then approximating the edge curves with local PC lines. Although this procedure is theoretically appealing, it has two major limitations. First, it is relatively complicated to use because the deblurred image can only be obtained after the possible step and roof/valley edges are detected and the local PC lines are fitted, which are quite technical for general users. Second, the local PC lines may not approximate the edge curves well when the local edge structure is complicated.

In this chapter, we propose an efficient BID approach. This approach uses local pixel clustering to accomplish the challenging task of restoring the complicated edge structure tapered by blur without detecting the possible edge pixels explicitly. Furthermore, similar to the edge-detection-based BID procedure described in Chapter 3, it does not require restrictive assumptions on either the psf or the true image, and

it allows the psf to be location variant. The rest part of this chapter is organized as follows. The efficient BID procedure is described in detail in Section 4.2. Some numerical examples are presented in Section 4.3. Some discussions conclude the chapter in Section 4.4.

4.2 Methodology

We describe our proposed BID method in three parts. In Subsection 4.2.1, our new procedure is described in detail. In Subsection 4.2.2, a modification to address misclassification in local pixel clustering is discussed. In Subsection 4.2.3, selection of procedure parameters is discussed.

4.2.1 Proposed BID Method

Assume that an observed image follows the model

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

where $\{(x_i, y_j), i, j = 1, 2, \ldots, n\}$ are equally spaced pixels in the design space $\Omega = [0, 1] \times [0, 1]$, $\{Z_{ij}, i, j = 1, 2, \ldots, n\}$ are observed image intensities, and $\{\varepsilon_{ij}, i, j = 1, 2, \ldots, n\}$ are independent and identically distributed (i.i.d.) random errors with mean 0 and unknown variance σ^2. It is further assumed that f is continuous in Ω except on some edge curves (see Qiu (1998) for a mathematical definition).

For a given pixel $(x, y) \in [h_n, 1 - h_n] \times [h_n, 1 - h_n]$ where $h_n \in (0, 1/2)$ is a bandwidth parameter, let us consider its circular neighborhood

$$O(x, y; h_n) = \left\{ (x_i, y_j) \in \Omega : \sqrt{(x_i - x)^2 + (y_j - y)^2} \leq h_n^2 \right\}.$$

In this neighborhood, a local plane is fitted by the following local linear kernel (LLK)

smoothing procedure (cf., Fan and Gijbels (1996)):

$$\min_{a,b,c}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n}\left[Z_{ij}-a-b(x_i-x)-c(y_j-y)\right]^2 K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)\right\}, \tag{4.2}$$

where K is a circularly symmetric bivariate density kernel function with its support on the unit disk. The above LLK smoothing procedure approximates the image intensity surface locally by a plane and uses the kernel function K to control the weights in the weighted least squares procedure (4.2). Usually, K is chosen such that pixels closer to (x,y) receive more weights, which is intuitively reasonable because pixels closer to (x,y) should provide more information about the image intensity at (x,y). Let $(\widehat{a}(x,y),\widehat{b}(x,y),\widehat{c}(x,y))$ denote the solution to the minimization problem (4.2). They have the following expression:

$$\widehat{a}(x,y) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^{(1)}(x,y)Z_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^{(1)}(x,y)}, \tag{4.3}$$

$$\widehat{b}(x,y) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^{(2)}(x,y)Z_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^{(2)}(x,y)}, \tag{4.4}$$

$$\widehat{c}(x,y) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^{(3)}(x,y)Z_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^{(3)}(x,y)}, \tag{4.5}$$

where

$$\begin{aligned}
w_{ij}^{(1)}(x,y) &= [A_{11}(x,y)+A_{12}(x,y)(x_i-x)+A_{13}(x,y)(y_j-y)]K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right),\\
w_{ij}^{(2)}(x,y) &= [A_{21}(x,y)+A_{22}(x,y)(x_i-x)+A_{23}(x,y)(y_j-y)]K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right),\\
w_{ij}^{(3)}(x,y) &= [A_{31}(x,y)+A_{32}(x,y)(x_i-x)+A_{33}(x,y)(y_j-y)]K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right),
\end{aligned}$$

$$A_{11}(x,y) = r_{20}(x,y)r_{02}(x,y)-r_{11}(x,y)r_{11}(x,y),$$

$$
\begin{aligned}
A_{12}(x,y) &= r_{01}(x,y)r_{11}(x,y) - r_{10}(x,y)r_{02}(x,y),\\
A_{13}(x,y) &= r_{10}(x,y)r_{11}(x,y) - r_{01}(x,y)r_{20}(x,y),\\
A_{21}(x,y) &= r_{01}(x,y)r_{11}(x,y) - r_{10}(x,y)r_{02}(x,y),\\
A_{22}(x,y) &= r_{00}(x,y)r_{02}(x,y) - r_{01}(x,y)r_{01}(x,y),\\
A_{23}(x,y) &= r_{01}(x,y)r_{10}(x,y) - r_{00}(x,y)r_{11}(x,y),\\
A_{31}(x,y) &= r_{10}(x,y)r_{11}(x,y) - r_{20}(x,y)r_{01}(x,y),\\
A_{32}(x,y) &= r_{01}(x,y)r_{10}(x,y) - r_{00}(x,y)r_{11}(x,y),\\
A_{33}(x,y) &= r_{00}(x,y)r_{20}(x,y) - r_{10}(x,y)r_{10}(x,y),\\
r_{s_1,s_2}(x,y) &= \sum_{i=1}^{n}\sum_{j=1}^{n}(x_i-x)^{s_1}(y_j-y)^{s_2}K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right),\ \text{for } s_1,s_2=0,1,2,
\end{aligned}
$$

and $\widehat{a}(x,y)$ in (4.3) is called the LLK estimator of $f(x,y)$. Its weighted residual mean square (WRMS) is defined by

$$
e(x,y)=\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left[Z_{ij}-\widehat{a}(x,y)-\widehat{b}(x,y)(x_i-x)-\widehat{c}(x,y)(y_j-y)\right]^2 K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)}{\sum_{i=1}^{n}\sum_{j=1}^{n}K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)}. \tag{4.6}
$$

If (x,y) is in the continuity region of f, then the image structure within $O(x,y;h_n)$ should be approximated well by the local plane described by $(\widehat{a}(x,y),\widehat{b}(x,y),\widehat{c}(x,y))$. Thus, $e(x,y)$ should be relatively small. On the other hand, if $O(x,y;h_n)$ contains edge curves, then the fitted local plane cannot well describe the image structure within $O(x,y;h_n)$. Consequently, the value of $e(x,y)$ would be relatively large. Therefore, $e(x,y)$ can be used to judge whether the neighborhood $O(x,y;h_n)$ contains any edge curves. More specifically, if

$$
e(x,y) > u_n, \tag{4.7}
$$

then we can conclude that there are edge curves in $O(x, y; h_n)$, where u_n is a threshold value. In such a case, we can cluster the pixels in $O(x, y; h_n)$ into two groups based on their observed image intensities. Intuitively, pixels on the same side of an edge curve have similar intensity values. So, they can be put in the same group. Pixels on different sides of the edge curve have quite different intensity values, and they should be put in different groups. An informative pixel clustering procedure should generate groups such that pixels within a group are similar in their intensity values and pixels in different groups have quite different intensity values. Thus, pixel clustering within $O(x, y; h_n)$ should reflect the local edge structure well without requiring restrictive conditions on the smoothness or shape of the edge curve. In this paper, we suggest using a cut-off constant c to define the two clusters in $O(x, y; h_n)$. More specifically, the two clusters are defined to be

$$\begin{aligned} O_1(x, y; h_n, c) &= \{(x_i, y_j) \in O(x, y; h_n) : Z_{ij} \leq c\}, \\ O_2(x, y; h_n, c) &= \{(x_i, y_j) \in O(x, y; h_n) : Z_{ij} > c\}, \end{aligned}$$

where $c \in R(x, y; h_n)$, and $R(x, y; h_n)$ is the range of the image intensity values in $O(x, y; h_n)$. Namely,

$$R(x, y; h_n) = \left(\min_{(x_i, y_j) \in O(x, y; h_n)} Z_{ij}, \max_{(x_i, y_j) \in O(x, y; h_n)} Z_{ij} \right).$$

So, it is obvious that both $O_1(x, y; h_n, c)$ and $O_2(x, y; h_n, c)$ are non-empty sets for any constant $c \in R(x, y; h_n)$, $O(x, y; h_n, c) = O_1(x, y; h_n, c) \cup O_2(x, y; h_n, c)$, and $O_1(x, y; h_n, c) \cap O_2(x, y; h_n, c) = \emptyset$. Let c_0 be the maximizer to the following maximization problem:

$$\max_{c \in R(x, y; h_n)} \frac{|O_1(x, y; h_n, c)|(\overline{\xi}_1 - \overline{\xi})^2 + |O_2(x, y; h_n, c)|(\overline{\xi}_2 - \overline{\xi})^2}{\sum_{(x_i, y_j) \in O_1(x, y; h_n, c)} (Z_{ij} - \overline{\xi}_1)^2 + \sum_{(x_i, y_j) \in O_2(x, y; h_n, c)} (Z_{ij} - \overline{\xi}_2)^2}, \tag{4.8}$$

where $|A|$ denotes the number of elements in the pointset A, $\overline{\xi}_s$ denotes the sample mean of the intensity values of the pixels in $O_s(x, y; h_n, c)$, for $s = 1, 2$, and $\overline{\xi}$ denotes the sample mean of the intensity values of all pixels in $O(x, y; h_n, c)$. In (4.8), the numerator measures the dissimilarity between the two groups, and the denominator measures the dissimilarity of intensity values within each of the two groups. Thus, it is reasonable to cluster the pixels in $O(x, y; h_n, c)$ by maximizing their ratio. It can be checked that (4.8) is actually the one dimensional version of the clustering criterion proposed by Friedman and Rubin (1967).

Without loss of generality, assume that $(x, y) \in O_1(x, y; h_n, c_0)$. Then, a weighted average of observations in $O_1(x, y; h_n, c_0)$ should provide a good estimate for $f(x, y)$ when there is no blurring involved, as discussed in the image denoising literature (cf., Qiu 1998). In cases when the observed image contains blur, if the intensity of a pixel is closer to the cut-off constant c_0, then it should receive less weight in the weighted average since it is more likely for that pixel to have blur involved. To address this issue about image blur, besides a bivariate kernel function used in the conventional kernel smoothing procedure to assign more weights to pixels closer to (x, y), a univariate kernel function is used to assign less weights to pixels whose intensity values are closer to c_0. Then, the proposed BID estimator $\widehat{f}(x, y)$ is defined to be the solution to a_0 in the following local constant kernel (LCK) smoothing procedure:

$$\min_{a_0 \in \mathbf{R}} \sum_{(x_i, y_j) \in O_1(x, y; h_n, c_0)} (Z_{ij} - a_0)^2 \, K\left(\frac{x_i - x}{h_n}, \frac{y_j - y}{h_n}\right) L\left(\frac{|Z_{ij} - c_0|}{|Z^{(1)}_{min} - c_0|}\right), \qquad (4.9)$$

where L is a univariate increasing density kernel function with support $[0, 1]$, and $Z^{(1)}_{min}$ denotes the minimum intensity in $O_1(x, y; h_n, c_0)$. It is easy to check that $\widehat{f}(x, y)$ has

the following expression:

$$\widehat{f}(x,y) = \frac{\sum_{(x_i,y_j)\in O_1(x,y;h_n,c_0)} Z_{ij} K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) L\left(\frac{|Z_{ij}-c_0|}{|Z^{(1)}_{min}-c_0|}\right)}{\sum_{(x_i,y_j)\in O_1(x,y;h_n,c_0)} K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) L\left(\frac{|Z_{ij}-c_0|}{|Z^{(1)}_{min}-c_0|}\right)}. \tag{4.10}$$

In cases when $(x,y) \in O_2(x,y;h_n,c_0)$, $\widehat{f}(x,y)$ can be defined in the same way except that $O_1(x,y;\ h_n,c_0)$ and $Z^{(1)}_{min}$ in (4.10) should be replaced by $O_2(x,y;h_n,c_0)$ and $Z^{(2)}_{max}$, respectively, where $Z^{(2)}_{max}$ denotes the maximum intensity in $O_2(x,y;h_n,c_0)$.

To demonstrate the efficacy of the image deblurring procedure (4.10), a cross section of an image around a step edge, a blurred version, a blurred-and- noisy version and the deblurred version by (4.10) when K and L are chosen to be the ones used in Section 3 are shown in plots (a)-(d) of Figure 4.1, respectively. From plot (d), it can be seen that (4.10) can restore the blurred edge structure to some extent while removing the noise at the same time.

In cases when (4.7) is not satisfied, it is likely that the pixel (x,y) is in a continuity region of f. In such cases, the spatial blur would not alter the image much, as discussed in Section 1. So, we suggest estimating $f(x,y)$ by the conventional LLK estimator $\widehat{a}(x,y)$ in (4.3). There are two benefits of doing this. First, it has been well demonstrated in the literature that the LLK estimator has less bias compared to the LCK estimator in continuity regions of f (cf., Fan and Gijbels (1996)). Second, since $\widehat{a}(x,y)$ is already computed as a by-product in (4.7) before we compute $\widehat{f}(x,y)$, it saves much computation. By taking into account all these considerations, our proposed BID procedure is summarized below.

Proposed Blind Image Deblurring Procedure

1. For a given pixel (x,y), solve the minimization problem (4.2) by (4.3)-(4.5).

2. Compute the WRMS in (4.6).

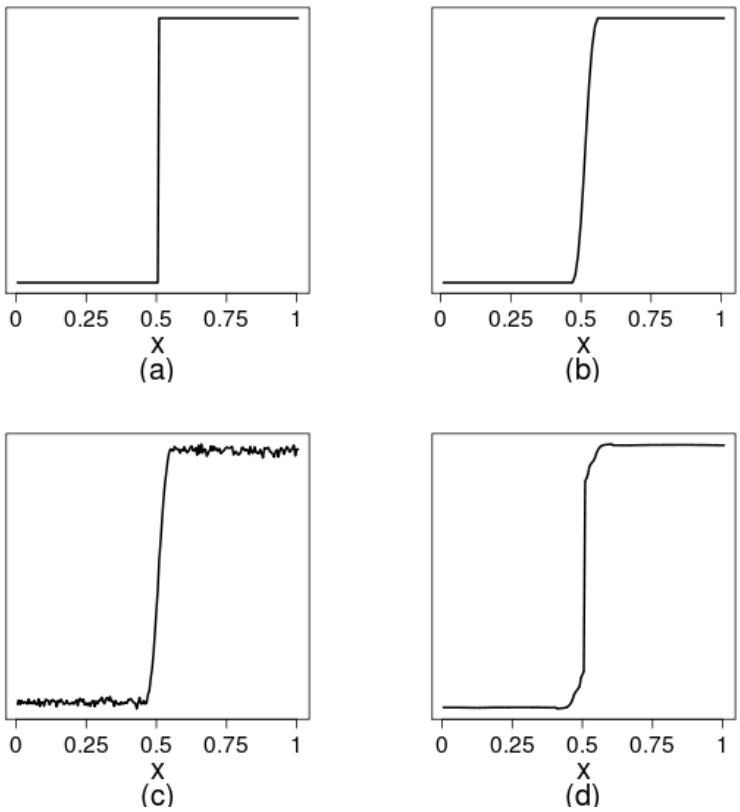

Figure 4.1: (a): A cross section of an image around a step edge; (b): A blurred version of (a); (c): A blurred-and-noisy version of (a); (d): The deblurred version from (c) by the BID procedure (4.10).

3. If (4.7) holds, then do local clustering by maximizing (4.8) and estimate $f(x, y)$ by (4.10). Otherwise, estimate $f(x, y)$ by (4.3).

4.2.2 A Modification

When the noise level of the observed image is quite low, the clustering procedure introduced in the previous subsection works well. However, its misclassification rate would get higher as the observed image gets noisier. As a brief demonstration, let us consider a toy example about an image with a single step edge. A blurred version and a blurred-and-noisy version of this image are shown in Figure 4.2(a), (b), respectively. The deblurred image by our proposed BID procedure (4.8)– (4.10) is presented in Figure 4.2(c). It can be seen that the step edge structure is restored well and the noise is mostly removed as well. But, the true image intensities of certain pixels around the step edge are not estimated well, which is mainly caused by the misclassification of the clustering procedure. To address this issue, we suggest a modification procedure described below. Assume that (4.7) holds at a given pixel (x, y). Then, local clustering needs to be performed in $O(x, y; h_n)$ by the clustering procedure discussed in Subsection 2.1 and all the pixels in that neighborhood are grouped into two clusters. If more than four of the eight immediately neighboring pixels of (x, y) are grouped into the same cluster as (x, y), then we say that (x, y) has been correctly classified. Otherwise, we say that (x, y) has been misclassified because (x, y) should belong to the same cluster as most of its immediately neighboring pixels (cf., Figure 4.3 for a demonstration). The deblurred image of the one presented in Figure 4.2(b) by the BID procedure (4.8)–(4.10) after the modification is shown in Figure 4.2(d). Compared to Figure 4.2(c), it can be seen that the misclassification rate has been greatly reduced by the suggested modification.

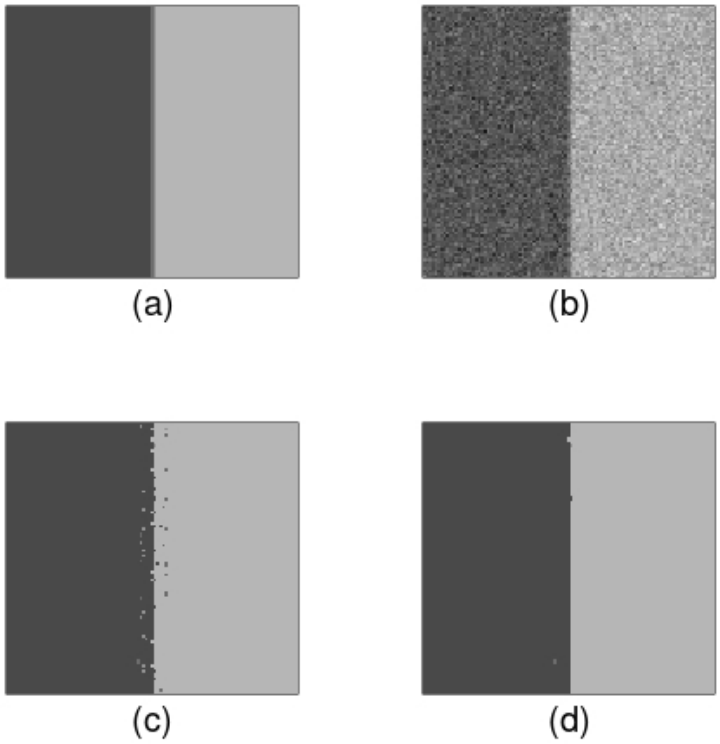

Figure 4.2: (a): A blurred version of an image with a step edge; (b): A blurred-and-noisy version; (c): The deblurred image by the BID procedure (4.8)–(4.10) without the modification; (d): The deblurred image by the BID procedure (4.8)–(4.10) with the modification.

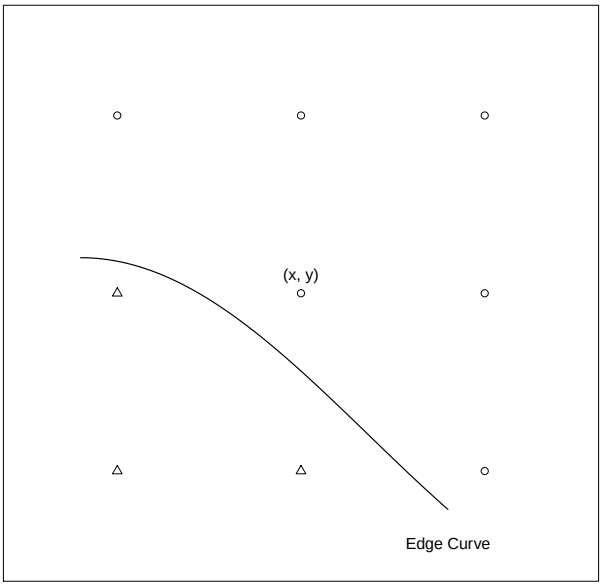

Figure 4.3: Pixels separated by an edge curve belong to different clusters, which are represented by two different symbols: circles and triangles.

4.2.3 Parameter Selection

In the proposed BID procedure (4.8)–(4.10), there are two parameters to choose, including the threshold value, u_n in (4.7) and the bandwidth h_n in (4.2). To choose a reasonable threshold value u_n, we need to derive the asymptotic distribution of $e(x, y)$ defined in (4.6). Based on (4.6), we have

$$
\begin{aligned}
& e(x,y) \\
= \; & \frac{1}{\sum_{(x_i,y_j)\in O(x,y;h_n)} K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)} \sum_{O(x,y;h_n)} [\varepsilon_{ij} + H\{f\}(x_i,y_j) - \widehat{a}(x,y) - \\
& \widehat{b}(x,y)(x_i-x) - \widehat{c}(x,y)(y_j-y)\Big]^2 K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right) \\
= \; & \frac{\sum_{O(x,y;h_n)} \varepsilon_{ij}^2 K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)}{\sum_{(x_i,y_j)\in O(x,y;h_n)} K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)} + \frac{2}{\sum_{(x_i,y_j)\in O(x,y;h_n)} K\left(\frac{x_i-x}{h_n}, \frac{y_j-y}{h_n}\right)} \cdot \\
& \sum_{O(x,y;h_n)} \Big\{\varepsilon_{ij}\Big[H\{f\}(x_i,y_j) - \widehat{a}(x,y) - \widehat{b}(x,y)(x_i-x) - \widehat{c}(x,y)(y_j-y)\Big] \cdot
\end{aligned}
$$

$$K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)\Big\}+\frac{1}{\sum_{O(x,y;h_n)}K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)}\cdot\sum_{O(x,y;h_n)}[H\{f\}(x_i,y_j)-$$
$$\widehat{a}(x,y)-\widehat{b}(x,y)(x_i-x)-\widehat{c}(x,y)(y_j-y)\Big]^2K\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)$$
$$=: I_1(x,y)+I_2(x,y)+I_3(x,y).$$

Note that $H\{f\}$ is actually a continuous function. By some arguments similar to those in the proofs of Theorem 2 in Qiu (2009) and Lemma A.1 in Sun and Qiu (2007), it can be checked that both $I_2(x,y)$ and $I_3(x,y)$ converge to zero fast. Also, we notice that $I_1(x,y)$ is a weighted average of i.i.d. random variables $\{\varepsilon_{ij}\}$. By the Central Limit Theorem, $e(x,y)$ has an asymptotic normal distribution with mean μ_e and variance σ_e^2, where

$$\mu_e=\sigma^2,\qquad \sigma_e^2=\frac{(E(\varepsilon_{11}^4)-\sigma^2)\sum_{i=1}^n\sum_{j=1}^nK\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)^2}{\left[\sum_{i=1}^n\sum_{j=1}^nK\left(\frac{x_i-x}{h_n},\frac{y_j-y}{h_n}\right)\right]^2}. \tag{4.11}$$

Therefore, a natural choice of u_n is $\mu_e+\sigma_eZ_{1-\alpha_n/2}$, where $Z_{1-\alpha_n/2}$ is a $(1-\alpha_n/2)^{\text{th}}$ quantile of the standard normal distribution in (4.11), the significance level α_n needs to be specified beforehand, both $E(\varepsilon_{11}^4)$ and σ^2 are usually unknown in practice. But they can be reasonably estimated using the conventional local kernel smoothing method.

In numerical simulations, the true image is often known. In such cases, h_n can be chosen by minimizing

$$MSE(f,\widehat{f};h_n)=\frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n\Big[f(x_i,y_j)-\widehat{f}(x_i,y_j)\Big]^2, \tag{4.12}$$

where $\widehat{f}$ is the deblurred image. In practice, f is usually unknown. In such cases, the cross validation (CV) approach is natural to consider (cf., Qiu (2005), Chapter 2).

In the image deblurring problem, however, the mean response is $H\{f\}$, instead of f. In such cases, the CV approach is inappropriate to use because the chosen parameter is for approximating $H\{f\}$. So we suggest choosing h_n by visual perception of the deblurred image. Note that the choice of h_n is related to the image resolution (i.e., n), the noise level, and the blurring extent (see its definition in Section 3). Based on our numerical experience, when deblurring a typical image with 512×512 pixels by the proposed BID method, it often gives a reasonably good result by choosing h_n between 6/512 and 10/512.

4.3 Numerical Examples

In this section, we discuss several numerical examples concerning the performance of the proposed BID procedure, denoted as NEW, in comparison with three representative state-of-the-art existing deblurring methods in the literature. The first existing method considered here is the one accomplished by the MATLAB blind deconvolution routine *deconvblind*, which is based on the method discussed by Biggs and Andrews (1997) and Jansson (1997) under the framework of Richardson-Lucy (RL) algorithm. The second existing method is the total variation (TV) image deblurring method proposed by Oliveira et al. (2009). The third existing method is the blind image deconvolution procedure developed under the Bayesian framework by Fergus et al. (2006). These three existing methods are denoted as RL, TV and Bayes, respectively. It should be pointed out that both RL and Bayes are blind image deblurring schemes, but TV is designed for non-blind image deblurring. Two versions of TV, denoted as TV_1 and TV_2, distinguished by how the psf h is specified, are considered in each numerical example. The specific meanings of TV_1 and TV_2 will be clarified in the context of each example. Throughout this section, the bandwidth h_n used in (4.2) is chosen by minimizing (4.12) except in the example of brain image where h_n is

chosen by visual perception of the deblurred image, the significance level α_n is fixed at 0.001, the two dimensional kernel function K used in (4.2) and (4.10) is chosen to be $(2/\pi)(1-x^2-y^2)I(x^2+y^2\le 1)$, and the one dimensional kernel function L used in (4.10) is chosen to be $(1/1.194958)\exp(x^2/2)I(0\le x\le 1)$. We choose these two kernel functions because the former is the Epanechnikov kernel function, which is a standard choice in the Statistics literature, and the latter is a truncated Gaussian kernel function, which is commonly used in the Computer Science literature.

In the first example, we consider the Lena test image with 512×512 pixels. The following two psf's are used:

$$h_1(u,v;x,y) = \begin{cases} \frac{1}{C_1(x,y)}\exp\{-\frac{u^2+v^2}{2}\}I(u^2+v^2\le 0.1^2) & \text{if } y>0.5,\\ \delta_0(u)\delta_0(v) & \text{otherwise;}\end{cases}$$

$$h_2(u,v;x,y) = \begin{cases} \frac{1}{C_2(x,y)}I(|u|\le 0.1)\delta_0(v) & \text{if } |x-0.5|\le 0.3 \text{ and } |y-0.5|\le 0.3,\\ \frac{1}{C_2(x,y)}\delta_0(u)I(|v|\le 0.1) & \text{otherwise,}\end{cases}$$

where $C_j(x,y)$ is the standardization constant such that $\int\int_{\mathbf{R}^2} h_j(u,v;x,y)\,dudv=1$, for any $(x,y)\in\Omega$ and $j=1,2$, and $\delta_0(\cdot)$ is the delta function with the point mass at 0. The random noise is generated from the normal distribution $N(0,\sigma^2)$, and two different noise levels, $\sigma=5$ and 10 are considered. From the above expression, we can see that h_1 is a truncated Gaussian blur for the upper half of the image and there is no blur for the lower half; g_2 is a horizontal motion blur for the central part of the image and is a vertical motion blur for the rest part of the image. Figure 4.4(a)-(c) present the original Lena image, its blurred version with h_2, and its blurred-and-noisy version with h_2 and $\sigma=10$, respectively. Figure 4.4(d)–(h) present the deblurred images by NEW, RL, TV_1, TV_2 and Bayes, respectively, where TV_1 and TV_2 denotes the TV method when the psf is specified as the horizontal motion blur of h_2 and the vertical motion blur of h_2, respectively. It should be pointed out that the support of the

Figure 4.4: (a)–(c): Original Lena image, its blurred version and its blurred-and-noisy version, respectively. (d)–(h): Deblurred images by NEW, RL, TV_1, TV_2 and Bayes, respectively.

psf needs to be specified when using RL and the true support of h_2 is used in this example to show its best performance, and a subregion defined by the coordinates $[86/512, 214/512] \times [293/512, 421/512]$ is prespecified for Bayes, as suggested in Fergus et al. (2006) that their algorithm performs better and runs faster if a smaller patch, rich in edge structure, is manually selected. From Figure 4.4, it can be seen that (i) NEW removes noise and the blur well, (ii) there are many artifacts in the deblurred image of RL and the noise has not been reduced much, and (iii) TV generates many artifacts at places where the psf is misspecified.

Next, we compare the five methods quantitatively. Table 4.1 presents the values of the root mean squared error (RMSE) of the five methods for each case considered based on 100 replicated simulations, where RMSE is the square root of MSE defined in (4.12). The number in each parenthesis represents the standard error of the corresponding RMSE. In cases when the psf is g_1, TV_1 and TV_2 denotes the TV method when the psf is specified as the Gaussian blur of g_1 and a horizontal motion

blur, respectively. From Table 4.1, it can be seen that NEW outperforms all the other four methods. Interestingly, all deblurring methods except the two TV-based methods perform better as the noise level decreases. But, it turns out that the two TV-based methods perform counter-intuitively the other way. We checked the source code and it seemed that the TV method presented in Oliveira et al. (2009) is adaptive in the sense that it does not require users to specify the value of the regularization parameter. In our specific cases when $\sigma = 5$, the TV method automatically chooses the parameter value that does not give a good performance.

Table 4.1: Estimated values of RMSE of the five image deblurring methods in the Lena image example based on 100 replicated simulations. The numbers in the parentheses are the standard errors of RMSE.

Methods	h_1		h_2	
	$\sigma = 5$	$\sigma = 10$	$\sigma = 5$	$\sigma = 10$
New	9.13 (0.01)	9.54 (0.02)	12.09 (0.02)	12.99 (0.03)
RL	19.80 (0.03)	30.73 (0.05)	26.61 (0.04)	29.67 (0.08)
TV_1	295.65 (9.66)	17.11 (0.49)	79.9 (2.02)	26.34 (2.79)
TV_2	185.30 (1.56)	92.24 (4.90)	22.73 (0.22)	14.64 (0.64)
Bayes	26.43 (1.54)	38.11 (1.08)	27.93 (1.11)	40.91 (1.95)

Next, we discuss the second numerical example, in which the test image of peppers with 256×256 pixels is used. The psf g considered has the expression:

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

where $r(x, y) > 0$ may change over location and it is the radius of the circular support of g. In this papers, $r(x, y)$ is called the *blur extent* function. Three blur extent functions, $r_1(x, y) = 0.03(1-(x-0.5)^2-(y-0.5)^2)$, $r_2(x, y) = 0.03x$, $r_3(x, y) = 0.02$, and two noise levels, $\sigma = 5$, $\sigma = 10$, are considered. Clearly, $r_1(x, y)$ and $r_2(x, y)$, are location variant., and $r_3(x, y)$ is location invariant. In the case with $r_3(x, y)$, the blur described by $g(u, v; x, y)$ is homogeneous across the entire image, which is the case

discussed by most references. As in the previous example, the noise is generated from the distribution $N(0, \sigma^2)$. Regarding the five image deblurring methods, we would like to make the following remarks. (i) RL requires the blur extent function to be constant (i.e., location invariant) and completely specified. So, in this example, we searched the value of r to achieve the minimum RMSE such that RL performs the best. (ii) TV requires the psf g to be completely specified and the blur extent function needs to be constant as well. In this example the value of r is searched to achieve the minimum RMSE values for TV_1 and TV_2 as well, where TV_1 and TV_2 denote the TV method when the parametric form of g is correctly specified and when it is misspecified as a horizontal motion blur, respectively. (iii) The prespecified subregion for Bayes is chosen to be $[78/256, 206/256] \times [42/256, 170/256]$.

The results in the same setup as Figure 4.4 are shown in Figure 4.5, where the blur extent function $r_2(x, y)$ and $\sigma = 10$ are considered. From the figure, it can be seen that (i) the blur gets more severe when moving from the left side of the image to the right side (cf., plot(b)), (ii) NEW deblurs the image and removes the noise well, (iii) RL performs poorly, (iv) the middle part of the deblurred image by TV_1 looks good but the places near the boundary contain many artifacts because TV cannot handle location variant blur, (v) TV does not work well when the blurring mechanism is misspecified (cf., plot(g)), and (vi) Bayes performs poorly in this example.

In cases when $r_3(x, y)$ (i.e., blur is location invariant) and $\sigma = 10$ are considered, the results are shown in Figure 4.6. In this case, TV is denoted by TV_1 when the psf is correctly specified and by TV_2 when the psf is wrongly specified as a horizontal motion blur. From Figure 4.6, we can see that (i) both RL and Bayes perform poorly, (ii) TV is highly sensitive to the misspecification of the psf, (iii) TV_1 performs well in this case because the psf is location invariant and completely specified, and (iv) NEW still gives a comparable performance to TV_1 despite it uses much less prior information.

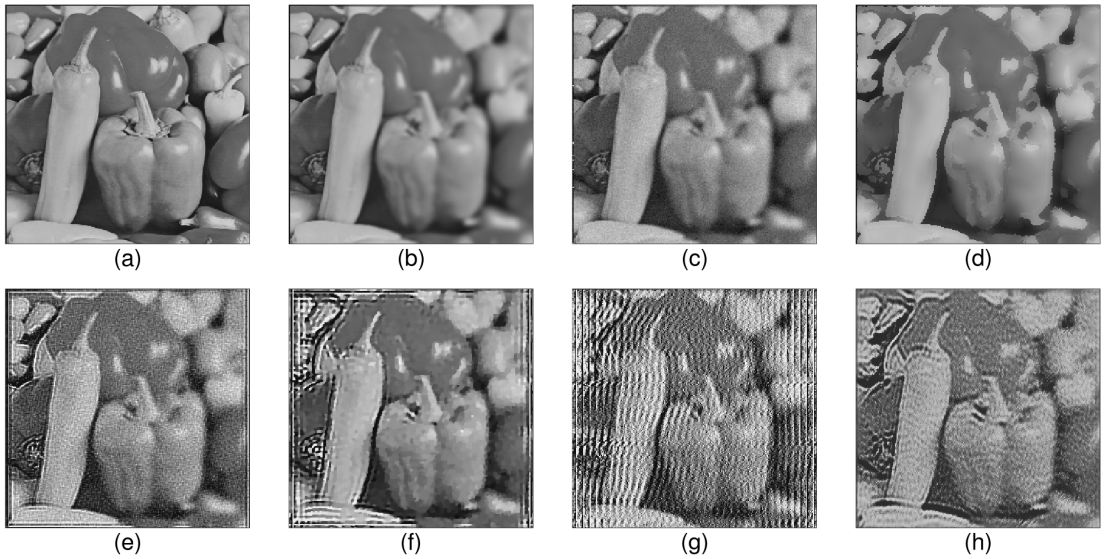

Figure 4.5: (a)–(c): Original Peppers image, its blurred version and its blurred-and-noisy version, respectively. (d)–(h): Deblurred images by NEW, RL, TV_1, TV_2 and Bayes, respectively.

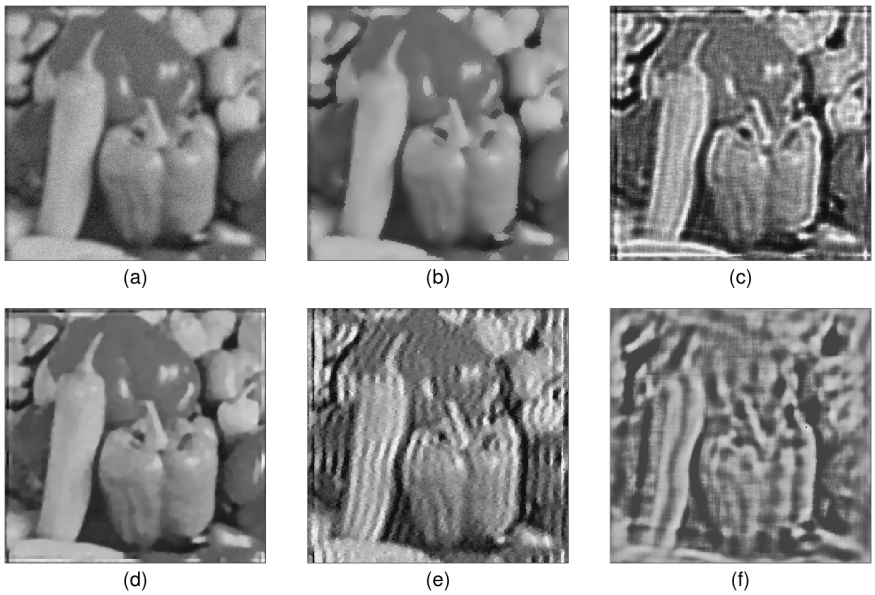

Figure 4.6: (a) Blurred-and-noisy Peppers image in the case when the blur extent function is $r_3(x, y)$ and $\sigma = 10$. (b)–(f): Deblurred images by NEW, RL, TV_1, TV_2 and Bayes, respectively.

The quantitative performance measures of the five methods in the same setup as that of Table 4.1 are presented in Table 4.2. It can be seen from Table 4.2 that (i) NEW outperforms all the other four methods in most cases considered and it works stably as the blur extent function and noise level change, (ii) TV_1 , which requires the parametric form of the psf is correctly specified, works slightly better than NEW in a few cases , and (iii) RL, TV_2 and Bayes all perform poorly.

Table 4.2: Estimated values of RMSE of the five image deblurring methods in the the Peppers image example based on 100 replicated simulations. The numbers in the parentheses are the standard errors of RMSE.

Methods	$r_1(x,y)$		$r_2(x,y)$		$r_3(x,y)$	
	$\sigma = 5$	$\sigma = 10$	$\sigma = 5$	$\sigma = 10$	$\sigma = 5$	$\sigma = 10$
New	18.92	19.78	15.44	16.91	17.27	18.12
	(0.04)	(0.06)	(0.05)	(0.07)	(0.04)	(0.06)
RL	27.03	34.73	42.79	47.16	37.91	40.15
	(0.06)	(0.12)	(1.46)	(0.69)	(0.10)	(0.18)
TV_1	22.94	19.64	65.37	63.95	26.84	17.76
	(0.09)	(0.08)	(0.93)	(5.98)	(0.39)	(0.09)
TV_2	27.71	38.60	76.25	103.17	92.82	62.01
	(0.26)	(12.02)	(0.66)	(1.56)	(2.13)	(26.69)
Bayes	29.19	45.05	34.80	43.48	28.28	42.89
	(6.75)	(3.89)	(9.00)	(7.07)	(10.00)	(7.68)

Finally, we consider an example with a brain test image. Figure 4.7(a) shows an observed brain image with 217 $\times$ 217 pixels which seems to have some blur involved. Its noisy version is shown in Figure 4.7(b), where the noise is generated from $N(0, 7^2)$. Figure 4.7(c)–(f) present the deblurred images by NEW, RL, TV and Bayes, respectively. The bandwidth in NEW is chosen to be 4/217. The support of the psf for RL is chosen to give its best visual impression. For TV, the psf is specified as a horizontal motion blur and the blur extent is chosen to give the best visual impression. We also tried several other forms of psf for TV but they did not provide significant improvements. The prespecified subregion required by Bayes is chosen to

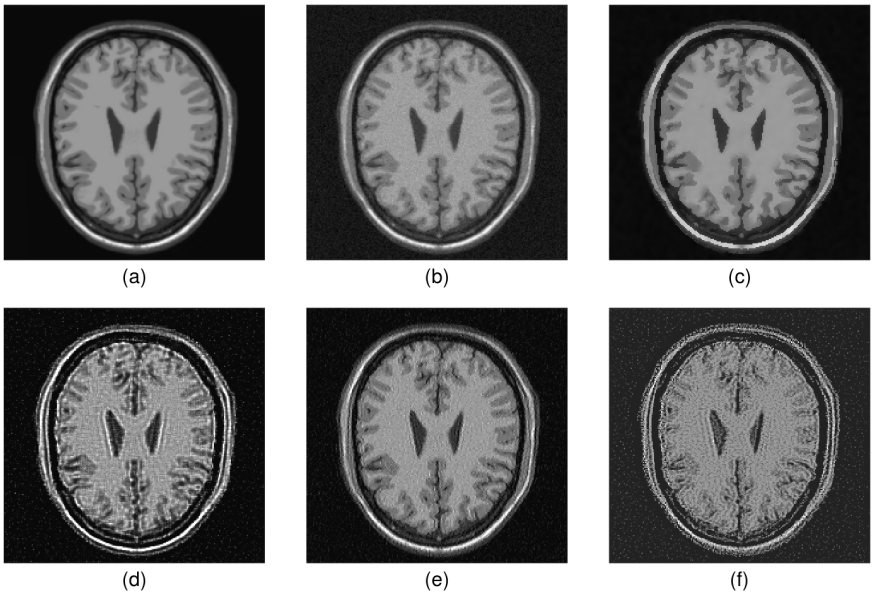

Figure 4.7: (a): A brain image with some blurring involved. (b): A noisy version of (a). (c)–(f): Deblurred images by NEW, RL, TV and Bayes, respectively.

be $[84/217, 138/217] \times [22/217, 76/217]$. It can be seen from Figure 4.7 that (i) NEW sharpens the image and removes the noise efficiently, (ii) both RL and Bayes generate many artifacts in their deblurred images around edges, and (iii) the deblurred image by TV does not seem to improve much of the observed image.

4.4 Discussions

We have proposed an efficient blind image deblurring method which simultaneously removes spatial blur and pointwise noise from an observed image without imposing restrictive assumptions on the blurring mechanism. This method is based on our observation that spatial blur alters the image structure significantly around step edges, but does not change image structure much in continuity regions of the image intensity surface. The challenging task of restoring complicated edge structures tapered by blurring is accomplished by a local clustering procedure and by a weighted lo-

cal smoothing. Numerical comparison with some state-of-the-art image deblurring methods shows that our proposed procedure can do a better job in a wide variety of different blurs and different noise levels.

Our proposed methodology can be generalized in several directions. First, the current method focuses on removing blur around step edges because those places dominate human visual perception. A natural improvement is to properly deblur the observed image around roof/valley edges as well. Moreover, features other than step and roof/valley edges (e.g., peaks, corners, etc.) should also be preserved during image deblurring and denoising. Second, we used a single bandwidth for local smoothing in the current method. The idea of multilevel smoothing that uses variable bandwidths can be incorporated into the proposed method. Third, our method is not fully automatic in the sense that it requires users to choose the bandwidth parameter via visual perception. Automatic bandwidth selection is notoriously difficult in the blind deconvolution problem and it requires much future research.

References

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

Biggs, D. and Andrews, M. (1997). Acceleration of iterative image restoration algorithms. *Applied Optics*, 36 (8):1766–1775.

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

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

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

Chu, C., Siao, J., Wang, L., and Deng, W. (2012). Estimation of 2d jump location curve and 3d jump location surface in nonparametric regression. *Statistics and Computing*, 22:17–31.

Clark, J. (1989). Authenticating edges produced by zero-crossing algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11:43–57.

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

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. (2006). Removing camera shake from a single photograph. *ACM Transactions on Graphics, SIGGRAPH 2006 Conference Proceedings, Boston, MA*, 25, 3:787–794.

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

Fleck, M. (1992). Some defects in finite-difference edge finders. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14:337–345.

Friedman, H. and Rubin, J. (1967). On some invariant criteria for grouping data. *Journal of the American Statistical Association*, 62:1159–1178.

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

Garlipp, T. and Müller, G. (2007). Robust jump detection in regression surface. *Sankhya (Series A)*, 69:55–86.

Gijbels, I. and Goderniaux, A. (2004). Bandwidth selection for changepoint estimation in nonparametric regression. *Technometrics*, 46:76–86.

Gonzalez, R. and Woods, R. (2002). *Digital Image Processing (2nd ed.)*. Prentice Hall: New York.

Hall, P. and Qiu, P. (2007a). Blind deconvolution and deblurring in image analysis. *Statistica Sinica*, 17:1483–1509.

Hall, P. and Qiu, P. (2007b). 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., Sarkar, S., Sanocki, T., and Bowyer, K. (1998). Comparison of edge detectors - a methodology and initial study. *Computer Vision and Image Understanding*, 69:38–54.

Hillebrand, M. and Müller, C. (2007). Outlier robust corner-preserving methods for reconstructing noisy images. *The Annals of Statistics*, 35:132–165.

Jansson, P. A. (1997). *Deconvolution of Images and Spectra*. Academic Press.

Joshi, M. 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.

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

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.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B*, 207:187–217.

Oliveira, J., Bioucas-Dias, J., and Figueiredo, M. (2009). Adaptive total variation image deblurring: A majorization-minimization approach. *Signal Processing*, 89:1683–1693.

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. (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. (2008). A nonparametric procedure for blind image deblurring. *Computational Statistics and Data Analysis*, 52:4828–4841.

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. (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.

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

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

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.

Torre, V. and Poggio, T. (1984). On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:147–163.

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

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

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