An Edge Detection Technique Using Local Smoothing and Statistical Hypothesis Testing

Peihua Qiu
Department of Statistics
University of Wisconsin – Madison
1210 West Dayton Street
Madison, WI 53706, USA

Suchendra M. Bhandarkar[*]
Department of Computer Science
University of Georgia
415 Boyd Graduate Studies Research Center
Athens, GA 30602 – 7404, USA

ABSTRACT

An edge detection technique based on local smoothing and statistical hypothesis testing for the detection and localization of step edges and roof edges is proposed. Smoothing and statistical hypothesis testing procedures for detection and localization of step edges and roof edges are formulated. Experimental results on gray scale images are presented. The merits, limitations and factors critical to the performance of the proposed technique are discussed. Possible improvements and future research directions are outlined.

Key Words: Edge Detection, Statistical Hypothesis Testing, Image Processing.

1 Introduction

Edge detection is the front-end processing stage in most computer vision and image understanding systems. The accuracy and reliability of edge detection is a critical factor in the overall performance of these systems. A variety of edges with varying intensity profiles have been defined in the literature; we discuss two of them in this paper. The first, called a *step edge* denotes a discontinuity in

[*]Author for correspondence

the image intensity function; the other, called a *roof edge* denotes continuity in the image intensity function but a discontinuity in the first order derivative of the image intensity function. Higher order edges can be similarly defined but step edges and roof edges are known to account for the most commonly occurring edges in real-world images. Accordingly, the edge detection technique proposed and discussed in this paper primarily considers the detection of these two edge types.

Torre and Poggio (1986) and Peli and Mallah (1982) present an excellent overview of edge detection. Traditional edge detection operators such as the gradient operator, the Laplacian operator or the Laplacian-of-Gaussian operator (Marr and Hildreth, 1980) represent high-pass filtering operations. These operators are only suitable for detecting limited types of edges and are highly susceptible to noise often resulting in fragmented edges. More recent edge detection techniques are based on optimal filtering (Canny, 1987; Dickey and Shanmugam, 1977; Lee and Wasilkowski, 1991; Sarkar and Boyer, 1990; Shen and Castan, 1992), random field models (Cressie, 1991; Hansen and Elliot, 1982; Huang and Tseng, 1988), surface fitting (Haralick, 1984; Nalwa and Binford, 1986; Sinha and Schunk, 1992), heuristic state-space search (Ashkar and Modestino, 1978; Martelli, 1976; Montanari, 1971), anisotropic diffusion (Perona and Malik, 1990; Saint-Marc it et al., 1991), residual analysis (Chen *et al.*, 1991) and global cost minimization using hill-climbing search (Tan *et al.*, 1989), simulated annealing (Tan *et al.*, 1991), mean field annealing (Acton and Bovik, 1992) and the genetic algorithm (Bhandarkar *et al.*, 1994).

We suggest an alternative approach to edge detection in this paper. Let us consider a 9×9 neighborhood (i.e. mask) centered at a given pixel P in a gray scale image. This 9×9 mask can be regarded as a union of nine 3×3 submasks. We refer to each of these submasks as *second order* (SO) pixels. In order to distinguish them from SO pixels, we sometimes refer to the original pixels as *first order* (FO) pixels or simply as *pixels*. The SO pixel which includes P is denoted as $P^{(2)}$. $P^{(2)}$ has two pairs of diagonally neighboring SO pixels comprising the set N_D and two pairs of directly neighboring SO pixels comprising the set N_4. The subscript 4 in N_4 denotes that the corresponding elements in N_4 are the 4-connected neighbors of $P^{(2)}$. For each of the SO pixels in N_D and N_4 we compute the weighted average of their constituent 3×3 FO pixels. The weight of each FO pixel in the weighted average depends on its distance from P. The weighted average is then considered to be the *gray level* value of the corresponding SO pixel. It must be noted that although the gray levels of the original pixels are typically integers in the range $0 - 255$ (i.e. 8 bits per pixel), the gray levels of the SO pixels are in general real numbers. So in the context of

SO pixels, the term *gray level* is a somewhat of an abuse of notation but it still should not be a cause for confusion to the reader. We compute the absolute value of the difference of the gray level values for each pair of opposite SO pixels in N_D and N_4. The maximum absolute value of these four difference values, denoted as M_P, is the main criterion for deciding whether P is a step edge pixel or not. If there is no step edge pixel in the 9×9 neighborhood of P, then we can expect M_P to be small since the gray levels of eight SO pixels would be fairly close to one another. If P is a step edge pixel, then we can expect to find at least one pair of SO pixels that is separated by the edge that passes through pixel P; that is to say one SO pixel from such a pair would be expected to lie on one side of the edge and the other SO pixel on the other side of the edge. Based on our experience and the results of our experiments, we have found this assumption to be valid for most images. Consequently, in such cases, the absolute value of the difference in gray level values for this SO pixel pair would be large resulting in a large value for M_P. We then use a statistical hypothesis testing procedure to select a threshold value for M_P. Two factors are taken into consideration when determining the value of this threshold; one is the variation in the gray level values caused by the inevitable presence of noise and the other is the variation in the gray level values caused by the variation of the image intensity function itself. We propose and implement local smoothing techniques to distinguish between these two kinds of variations in order to determine the threshold value. For roof edges, in addition to a condition similar to that for M_P in the case of step edge detection, we also derive another condition specifically designed for roof edge detection. If a pixel satisfies both conditions, then it is flagged as a roof edge pixel.

From the brief discussion of our method as given above, we can see that it uses the concept of local smoothing (which is reflected in the computation of the weighted average) and concept of *second order* (SO) pixels to remove the noise. It also uses the differences between the gray level values for each of the SO pixel pairs along four directions to detect the edge. Furthermore, it eliminates to some degree the effect of the variation of the underlying intensity function on the edge detection. Comparing our technique with those based on surface fitting, one can see that we do not use a parametric model in the neighborhood of an image pixel P to approximate the underlying image intensity function and to compute the estimates of its directional derivatives. Instead, we directly use the gray level value differences for each pair of SO pixels to measure the extent of gray level discontinuity at P. This makes our technique model-independent and hence more flexible with regard to the underlying image intensity variation. Unlike most other edge operators to be found in the literature our technique considers four independent directions in detecting the edges, instead of only the x and y directions. This makes our technique more flexible with regard to the local smoothness of the underlying image intensity function.

The rest of the paper is organized as follows: In Section 2, we discuss our technique for step edge detection and outline a step edge detection algorithm. In Section 3 we present and analyze the results of our algorithm on some gray scale images. We discuss some critical factors that affect the performance of our algorithm and compare it with some "classical" edge detectors such as the Sobel and the Laplacian of Gaussian (LoG) operators. In Section 4, we propose an algorithm for roof edge detection and illustrate it with a simple example. Finally, in Section 5 we conclude the paper and outline future research directions.

2 Step Edge Detection

We use the notation $P(i,j)$ to denote the gray level of the pixel at location (i,j) where $i = 1, 2, \cdots, N,\ j = 1, 2, \cdots, M$. At each image pixel $P(i,j)$, we define a 9×9 mask centered at pixel location (i,j). For pixels on the image boundary, we construct their masks assuming that the image is wrapped around at the boundaries. The 9×9 mask centered at pixel $P(i,j)$ consists of nine SO pixels. The central SO pixel is denoted as $P^{(2)}(i,j)$. Obviously, $P^{(2)}(i,j)$ has two pairs of diagonally neighboring SO pixels which constitute the set N_D and two pairs directly neighboring SO pixels which constitute the set N_4. Depending on their locations, these SO pixels are denoted as $P^{(2)}(i+r, j+t)$ where $-1 \leq r, t \leq 1$.

We compute the weighted average of the gray level values of the 3×3 FO pixels within a SO pixel and regard the weighted average to be the gray level value of the SO pixel. We illustrate the procedure for computing the weighted average by considering the SO pixel $P^{(2)}(i-1, j+1)$ shown in Fig. 2.1. The set of weights $\{a_1, a_2, a_3, a_4, a_5\}$ used in the weighted average are depicted in Fig. 2.2(a). The FO pixels in $P^{(2)}(i-1, j+1)$ with the same weight are seen to be equidistant from $P(i,j)$ in the sense of the D_4 (i.e. Manhattan) distance measure (Gonzalez and Woods, 1992). We require the weights $\{a_1, a_2, a_3, a_4, a_5\}$ to satisfy the following conditions:

(i) $a_1 + 2a_2 + 3a_3 + 2a_4 + a_5 = 1,\ a_i \geq 0,\ i = 1, 2, 3, 4, 5.$

(ii) $\{a_1, a_2, a_3, a_4, a_5\}$ is a linearly decreasing sequence and has a ratio relation as shown in Fig. 2.2(a).

In Fig. 2.2(a), the numbers $\{4, 5, 6, 7, 8\}$ on the horizontal axis denote the fact that the pixels with weight a_1 are 4 units away from $P(i, j)$ and so on. The distance measure used is the D_4 (i.e. Manhattan) distance measure. Condition (i) ensures that the weights are positive and normalized. Condition (ii) ensures that the weights decrease linearly with increasing Manhattan distance from pixel $P(i, j)$. These two conditions actually correspond to the triangular kernel density function used in statistical kernel smoothing techniques. For a detailed discussion on various smoothing kernels we refer the interested reader to the recent book by Wand and Jones (1995). We impose an additional constraint that the hypotenuse of the right-angled triangle shown in Fig. 2.2(a) intersect with the horizontal axis at point 9. This denotes the fact that pixels at Manhattan distance of 9 or greater from pixel $P(i, j)$ lie outside the neighborhood of interest. A unique weight sequence $\{a_1, a_2, a_3, a_4, a_5\}$ which satisfies all the above requirements can be then determined and is given by:

$$\{a_1, a_2, a_3, a_4, a_5\} = \{5/27, 4/27, 3/27, 2/27, 1/27\} \tag{2.1}$$

The same weights are used for the other SO pixels $\in N_D$; namely $P^{(2)}(i-1, j-1)$, $P^{(2)}(i+1, j-1)$ and $P^{(2)}(i+1, j+1)$.

Consider the weights $\{b_1, b_2, b_3\}$ (shown in Fig. 2.2(b)) for the SO pixel $P^{(2)}(i-1, j)$ (shown in Fig. 2.1). We require that the weights satisfy the conditions:

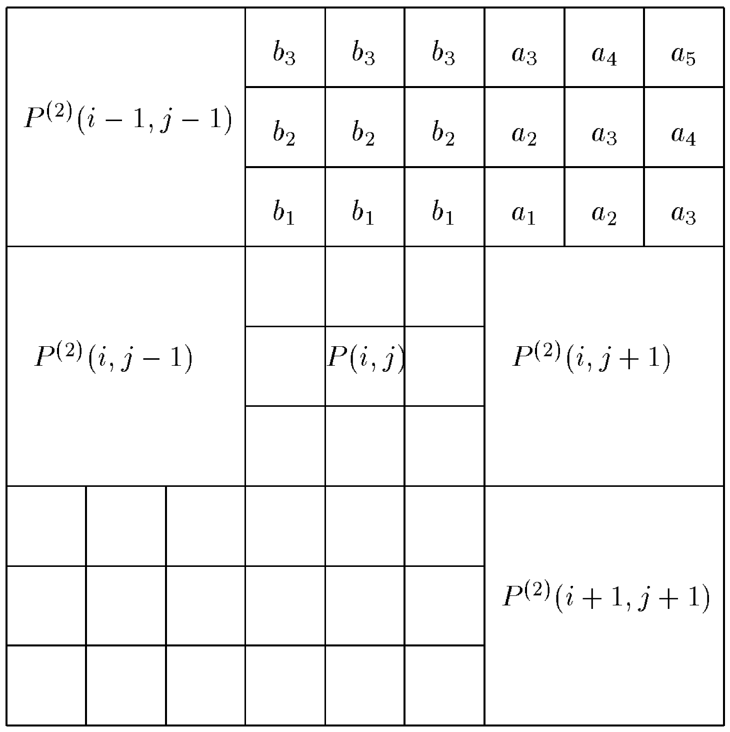
(i) $3 \cdot (b_1 + b_2 + b_3) = 1, \; b_i \geq 0, \; i = 1, 2, 3$

(ii) The weights decrease linearly with increasing Manhattan distance from the pixel $P(i, j)$.

(iii) The hypotenuse of the right-angled triangle shown in Fig. 2.2 (b) intersects the horizontal axis at point 5.

The rationale behind these conditions is the same as that cited for the weights $\{a_i; 1 \leq i \leq 5\}$. A unique solution satisfying all the above conditions is given by:

$$\{b_1, b_2, b_3\} = \{1/6, 1/9, 1/18\} \tag{2.2}$$



b_3 b_3 b_3 a_3 a_4 a_5
$P^{(2)}(i-1,j-1)$
b_2 b_2 b_2 a_2 a_3 a_4
b_1 b_1 b_1 a_1 a_2 a_3
$P^{(2)}(i,j-1)$
$P(i,j)$
$P^{(2)}(i,j+1)$
$P^{(2)}(i+1,j+1)$

Figure 2.1: The 9×9 mask centered at pixel $P(i,j)$ as a union of nine SO pixels. Weights $\{a_1, a_2, a_3, a_4, a_5\}$ and $\{b_1, b_2, b_3\}$ are used in the diagonally neighboring SO pixels and in the directly neighboring SO pixels respectively.

The same weights are used in other SO pixels $\in N_4$; namely, $P^{(2)}(i,j-1)$, $P^{(2)}(i+1,j)$ and $P^{(2)}(i,j+1)$.

Given the weights, the *gray levels* of the SO pixels $P^{(2)}(i-1,j+1)$ and $P^{(2)}(i-1,j)$ can be obtained thus:

$$\begin{aligned} P^{(2)}(i-1,j+1) &= a_1 P(i-2,j+2) + a_2[P(i-2,j+3) + P(i-3,j+2)] \\ &\quad + a_3[P(i-2,j+4) + P(i-3,j+3) + P(i-4,j+2)] \\ &\quad + a_4[P(i-3,j+4) + P(i-4,j+3)] + a_5 P(i-4,j+4) \end{aligned} \tag{2.3}$$

$$\begin{aligned} P^{(2)}(i-1,j) &= b_1[P(i-2,j-1) + P(i-2,j) + P(i-2,j+1)] \\ &\quad + b_2[P(i-3,j-1) + P(i-3,j) + P(i-3,j+1)] \\ &\quad + b_3[P(i-4,j-1) + P(i-4,j) + P(i-4,j+1)] \end{aligned} \tag{2.4}$$

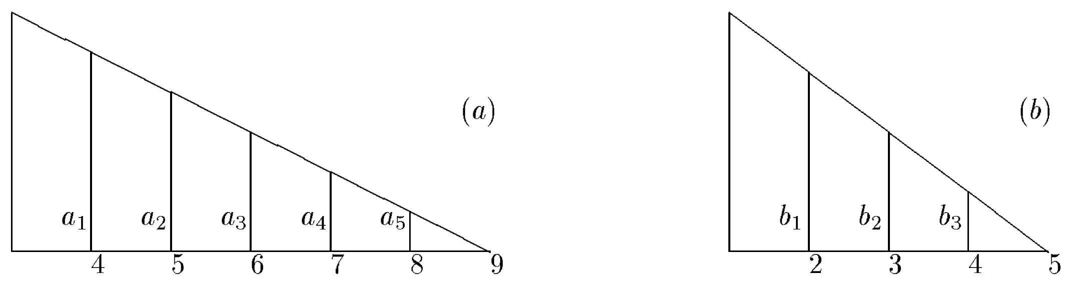


Figure 2.2: Weights $\{a_1, a_2, a_3, a_4, a_5\}$ and $\{b_1, b_2, b_3\}$ satisfy the ratio relations as depicted in (a) and (b) respectively.

The gray levels of other N_D and N_4 SO pixels can be obtained similarly. Intuitively, the differences $[P^{(2)}(i-1,j+1)-P^{(2)}(i+1,j-1)]$, $[P^{(2)}(i-1,j-1)-P^{(2)}(i+1,j+1)]$,$[P^{(2)}(i-1,j)-P^{(2)}(i+1,j)]$ and $[P^{(2)}(i,j-1)-P^{(2)}(i,j+1)]$ can be looked upon as measures of the intensity discontinuity at pixel $P(i,j)$. If there is no step edge pixel in the 9×9 mask centered at pixel $P(i,j)$, then all of the aforementioned differences could be expected to be small; on the other hand, if $P(i,j)$ is a step edge pixel, then some of them could be expected to be large. However, two other factors need to be considered as well; one is the variation of the underlying image intensity function in the 9×9 mask and the other is the variation of the gray levels caused by possible noise in the image.

Intuitively, the variation of the intensity function along a direction can be measured by its first order directional derivatives. For the direction from $P(i+3,j-3)$ to $P(i-3,j+3)$, we use

$$D(i-1,j+1) = (d_1+d_2)/2 \tag{2.5}$$

to estimate the first order directional derivative where

$$\begin{aligned} d_1 &\triangleq \frac{1}{3}[P(i-4,j+3)+P(i-4,j+4)+P(i-3,j+4)] \\ &\quad -\frac{1}{3}[P(i-3,j+2)+P(i-2,j+2)+P(i-2,j+3)] \\ d_2 &\triangleq \frac{1}{3}[P(i+2,j-3)+P(i+2,j-2)+P(i+3,j-2)] \\ &\quad -\frac{1}{3}[P(i+3,j-4)+P(i+4,j-4)+P(i+4,j-3)] \end{aligned} \tag{2.6}$$

Here, d_1 and d_2 can be regarded as the estimates of the first order directional derivatives at $P(i-3,j+3)$ and $P(i+3,j-3)$ respectively and $D(i-1,j+1)$ is their average. The famous Lagrange's Mean Value Theorem in differential calculus tells us that for a 1-D function $f(x)$ and two points $x_1 < x_2$, $f(x_2)-f(x_1) = f'(\theta)(x_2-x_1)$ for some $\theta \in [x_1,x_2]$ as long as $f(x)$ is differentiable in the interval $[x_1, x_2]$. This is equivalent to saying that the variation of $f(x)$ from x_1 to x_2 could be approximated by the derivative multiplied by the distance between those two points. We define

$$\begin{aligned}
D_*(i-1,j-1) &= \frac{25}{6} \cdot D(i-1,j-1) \\
D_*(i-1,j+1) &= \frac{25}{6} \cdot D(i-1,j+1) \\
D_*(i-1,j) &= \frac{8}{3} \cdot D(i-1,j) \\
D_*(i,j-1) &= \frac{8}{3} \cdot D(i,j-1) \qquad (2.7)
\end{aligned}$$

$D_*(i-1,j-1), D_*(i-1,j+1), D_*(i-1,j)$ and $D_*(i,j-1)$ are used to measure the variation of the intensity function along the directions from $P(i+3,j+3)$ to $P(i-3,j-3)$, from $P(i+3,j-3)$ to $P(i-3,j+3)$, from $P(i+3,j)$ to $P(i-3,j)$ and from $P(i,j+3)$ to $P(i,j-3)$ respectively. We refer to them as *correction terms.* The multiplicative factors of $\frac{25}{6}$ and $\frac{8}{3}$ are chosen such that the value of the edge detection criterion equals to zero when the intensity function is linear and when no edges exist at pixel (i,j). In other words, the linear variation of the intensity function is eliminated from the edge detection process. The derivation of these two multiplicative factors is given in the Appendix.

We define

$$\begin{aligned}
\xi_1 &\triangleq P^{(2)}(i-1,j+1) - P^{(2)}(i+1,j-1) - D_*(i-1,j+1) \\
\xi_2 &\triangleq P^{(2)}(i-1,j) - P^{(2)}(i+1,j) - D_*(i-1,j) \\
\xi_3 &\triangleq P^{(2)}(i-1,j-1) - P^{(2)}(i+1,j+1) - D_*(i-1,j-1) \\
\xi_4 &\triangleq P^{(2)}(i,j-1) - P^{(2)}(i,j+1) - D_*(i,j-1) \qquad (2.8)
\end{aligned}$$

$\{\xi_1, \xi_2, \xi_3, \xi_4\}$ can all be expected be very small if there is no step edge in the 9×9 mask centered at $P(i,j)$. Since in reality, the gray levels are usually contaminated by noise, we have to consider the variation caused by the noise as well. We assume that the observed gray levels in the image are a realization of the following model:

$$P(i,j) = g(i,j) + n(i,j),\ i = 1,2,\cdots,N,\ j = 1,2,\cdots,M$$

where $\{g(i,j)\}$ are the true gray levels and $\{n(i,j)\}$ are independent and identically distributed (i.i.d) noise samples with zero mean and variance σ^2. With some simple algebraic manipulations (detailed in the Appendix), we can show that

$$std(\xi_1) = std(\xi_3) = 2.6202\sigma$$

$$std(\xi_2) = std(\xi_4) = 1.7951\sigma \tag{2.9}$$

where $std(\xi_i)$ denotes the standard deviation of ξ_i.

We define

$$\begin{aligned} \eta_i &\triangleq \xi_i/2.6202, \, for \; i = 1, 3 \\ \eta_i &\triangleq \xi_i/1.7951, \, for \; i = 2, 4 \end{aligned} \tag{2.10}$$

If there is no edge pixel in the 9×9 mask centered at $P(i, j)$, then by the Central Limit Theorem in statistics (Bhattacharya and Johnson, 1977), $\{\eta_1, \eta_2, \eta_3, \eta_4\}$ are all approximately normally distributed with means zero and variances σ^2 since each of them is a linear combination of 18 observed gray levels. We then define

$$M_{P(i,j)} \triangleq \max\{|\eta_1|, |\eta_2|, |\eta_3|, |\eta_4|\} \tag{2.11}$$

as the step edge detection criterion. The statistical test corresponding to $M_{P(i,j)}$ can be formulated as

$$\begin{aligned} & P(M_{P(i,j)} > C(\alpha)) = \alpha \\ \Leftrightarrow \quad & 1 - [P(|\eta_1| < C(\alpha))]^4 = \alpha \\ \Leftrightarrow \quad & P(|\eta_1| < C(\alpha)) = (1 - \alpha)^{1/4} \\ \Leftrightarrow \quad & C(\alpha) = Z_{[1+(1-\alpha)^{1/4}]/2} \cdot \sigma \end{aligned}$$

where $Z_{[1+(1-\alpha)^{1/4}]/2}$ is the $[1 + (1 - \alpha)^{1/4}]/2$ percentile of the standard normal distribution.

Statistically speaking, when there is no edge pixel in the 9×9 mask centered at pixel $P(i, j)$ and when α is very small, $M_{P(i,j)}$ has little chance of exceeding $C(\alpha)$. In other words, if $M_{P(i,j)}$ is larger than $C(\alpha)$, then we have sufficient reason to conclude that there are edge pixels in the 9×9 mask. In such cases, we flag $P(i, j)$ as a step edge pixel. Thus, $C(\alpha)$ can be used as the threshold value for $M_{P(i,j)}$. But in most applications, since we do not know the value of σ, it has to be estimated from the image data. In the following discussion, we will suggest two estimators of σ. The applicable threshold value for $M_{P(i,j)}$ can then be computed as:

$$C(\alpha) = Z_{[1+(1-\alpha)^{1/4}]/2} \cdot \hat{\sigma} \tag{2.12}$$

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

Table 2.1: Several significance levels α and their corresponding $[1+(1-\alpha)^{1/4}]/2$ values and $Z_{[1+(1-\alpha)^{1/4}]/2}$ values. The notation $Ae-B$ denotes $A \times 10^{-B}$.

α	$[1+(1-\alpha)^{1/4}]/2$	$Z_{[1+(1-\alpha)^{1/4}]/2}$	α	$[1+(1-\alpha)^{1/4}]/2$	$Z_{[1+(1-\alpha)^{1/4}]/2}$
5.0e-01	0.9204482	1.408093	1.0e-05	0.9999987	4.708129
4.5e-01	0.9305868	1.480175	5.0e-06	0.9999994	4.847543
4.0e-01	0.9400559	1.555243	1.0e-06	0.9999999	5.157701
3.5e-01	0.9489504	1.634761	5.0e-07	0.9999999	5.286029
3.0e-01	0.9573456	1.720681	1.0e-07	1.0000000	5.573271
2.5e-01	0.9653024	1.815839	5.0e-08	1.0000000	5.692763
2.0e-01	0.9728708	1.924768	1.0e-08	1.0000000	5.961456
1.5e-01	0.9800923	2.055659	5.0e-09	1.0000000	6.073695
1.0e-01	0.9870019	2.226268	1.0e-09	1.0000000	6.326984
5.0e-02	0.9936293	2.490915	1.0e-10	1.0000000	6.673367
1.0e-02	0.9987453	3.022202	1.0e-11	1.0000000	7.003303
5.0e-03	0.9993738	3.226681	1.0e-12	1.0000000	7.318952
1.0e-03	0.9998750	3.662164	1.0e-13	1.0000000	7.622118
5.0e-04	0.9999375	3.836061	1.0e-14	1.0000000	7.905250
1.0e-04	0.9999875	4.214791	1.0e-15	1.0000000	8.209103
5.0e-05	0.9999937	4.368675	1.0e-16	1.0000000	Infinite

Remark 2.1: $Z_{[1+(1-\alpha)^{1/4}]/2}$ depends on the significance level α. Without confusion, we may sometimes refer to it as *the threshold value*. Table 2.1 lists several α values and their corresponding $[1+(1-\alpha)^{1/4}]/2$ values and $Z_{[1+(1-\alpha)^{1/4}]/2}$ values.

In the above hypothesis testing procedure, the probability of detecting a false edge at each pixel is α. On the other hand, edges with jump magnitude less than $1.7951 Z_{[1+(1-\alpha)^{1/4}]/2} \cdot \hat{\sigma}$ (edges with low SNR, c.f. equation (2.9)) could probably be missed. Thus $Z_{[1+(1-\alpha)^{1/4}]/2}$ works as a tuning parameter by which to balance missing-edge/false-edge possibilities. Although it is related to the confidence level α, this relationship is only *approximate* because of the use of the Central Limit Theorem which provides an *approximate* normal distribution for a finite sample size (only 18 in our case). In typical applications, the threshold value still needs to be somewhat heuristically adjusted. Theoretically, it is not difficult to establish the statistical consistency of the aforementioned edge detection procedure. Also, the missing-edge/false-edge probabilities both tend to zero with increasing spatial resolution of the image. We refer the interested readers to the related discussions in Qiu and Yandell (1995) for a more detailed treatment of this topic.

A nature estimator of σ is the sample standard deviation of the gray levels in the 9×9 mask centered at $P(i,j)$. But if there are edge pixels in the mask, this estimator performs poorly. We therefore propose to estimate σ in the following manner: In each of the eight neighboring SO pixels, we compute the sample variance of the gray levels of its 3×3 FO pixels. These variances are denoted as $s^2(i-r, j-t)$, where $r, t = -1, 0, 1$, $(r,t) \neq (0,0)$. We define

$$\hat{\sigma}(i,j) = \text{median}\{\tau(i,j-1), \tau(i-1,j-1), \tau(i-1,j), \tau(i-1,j+1)\} \tag{2.13}$$

where

$$\begin{aligned} \tau(i,j-1) &= \sqrt{[s^2(i,j-1)+s^2(i,j+1)]/2} \\ \tau(i-1,j-1) &= \sqrt{[s^2(i-1,j-1)+s^2(i+1,j+1)]/2} \\ \tau(i-1,j) &= \sqrt{[s^2(i-1,j)+s^2(i+1,j)]/2} \\ \tau(i-1,j+1) &= \sqrt{[s^2(i-1,j+1)+s^2(i+1,j-1)]/2} \end{aligned} \tag{2.14}$$

Obviously, each of $\tau(i,j-1), \tau(i-1,j-1), \tau(i-1,j)$ and $\tau(i-1,j+1)$, which is derived from the SO pixel pair $[P^{(2)}(i,j-1), P^{(2)}(i,j+1)]$, $[P^{(2)}(i-1,j-1), P^{(2)}(i+1,j+1)]$, $[P^{(2)}(i-1,j), P^{(2)}(i+1,j)]$ and $[P^{(2)}(i-1,j+1), P^{(2)}(i+1,j-1)]$ respectively, is a nature estimator of σ. We assume that if $P(i,j)$ is an edge pixel, then at least one of the SO pairs cited above is such that one of the SO pixels in the pair is on one side of the edge and the other SO pixel is on the other side. Estimators of σ based on the above SO pixel pairs will not be affected by the presence of edges in the 9×9 mask centered at pixel $P(i,j)$. In other words, $\hat{\sigma}(i,j)$ provides a good estimator of σ whether or not there are edge pixels in the 9×9 mask centered at pixel $P(i,j)$. We term $\hat{\sigma}(i,j)$ as *the zeroth-order estimator of* σ because the sample variance $s^2(i-r, j-t)$ can be regarded as the residual mean squares when the sample mean is regarded as the estimator of the true FO pixel gray levels in the SO pixel $P^{(2)}(i-r, j-t)$. The sample mean is a zeroth-order estimator in the sense that it is a zeroth-order polynomial.

Clearly, each of $s^2(i-r, j-t)$, $r, t = -1, 0, 1$, $(r,t) \neq (0,0)$, is a rough estimator of σ^2 because the sample mean is a rough estimator of the true FO pixel gray levels in the corresponding SO pixel.

A more accurate estimator of σ can be constructed as follows: In each of the eight neighboring SO pixels $P^{(2)}(i-r, j-t)$, we perform a least-squares planar fit and then obtain the residual mean squares $s_*^2(i-r, j-t)$. After some algebraic manipulations (detailed in the Appendix), we have the following expression for $s_*^2(i-r, j-t)$:

$$s_*^2(i-r, j-t) = \frac{1}{6}(\underline{Y}'\underline{Y} - \underline{Y}'P_X\underline{Y}) \tag{2.15}$$

where $\underline{Y}$ is a 9×1 vector whose elements are the gray levels of the 3×3 FO pixels in $P^{(2)}(i-r, j-t)$ scanned in a row-major (i.e. raster-scan) fashion (i.e. left-to-right and top-to-bottom). The 9×9 matrix P_X is given by:

$$P_X = \frac{1}{18} \times \begin{bmatrix} 8 & 5 & 2 & 5 & 2 & -1 & 2 & -1 & -4 \\ 5 & 5 & 5 & 2 & 2 & 2 & -1 & -1 & -1 \\ 2 & 5 & 8 & -1 & 2 & 5 & -4 & -1 & 2 \\ 5 & 2 & -1 & 5 & 2 & -1 & 5 & 2 & -1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -1 & 2 & 5 & -1 & 2 & 5 & -1 & 2 & 5 \\ 2 & -1 & -4 & 5 & 2 & -1 & 8 & 5 & 2 \\ -1 & -1 & -1 & 2 & 2 & 2 & 5 & 5 & 5 \\ -4 & -1 & 2 & -1 & 2 & 5 & 2 & 5 & 8 \end{bmatrix} \tag{2.16}$$

We then define

$$\hat{\sigma}_*(i,j) = median\{\tau_*(i,j-1), \tau_*(i-1,j-1), \tau_*(i-1,j), \tau_*(i-1,j+1)\} \tag{2.17}$$

where

$$\begin{aligned} \tau_*(i,j-1) &= \sqrt{[s_*^2(i,j-1) + s_*^2(i,j+1)]/2} \\ \tau_*(i-1,j-1) &= \sqrt{[s_*^2(i-1,j-1) + s_*^2(i+1,j+1)]/2} \\ \tau_*(i-1,j) &= \sqrt{[s_*^2(i-1,j) + s_*^2(i+1,j)]/2} \\ \tau_*(i-1,j+1) &= \sqrt{[s_*^2(i-1,j+1) + s_*^2(i+1,j-1)]/2} \end{aligned} \tag{2.18}$$

We term $\hat{\sigma}_*(i,j)$ as the *first-order estimator of* σ.

We summarize the step edge detection method discussed above in the following algorithm:

Step Edge Detection Algorithm

1. *For every pixel,* $P(i,j), i = 1,2,\cdots,N, j = 1,2,\cdots,M$, *consider the* 9×9 *mask centered at* $P(i,j)$. *Use equations (2.1)-(2.4) to compute* $P^{(2)}(i-r, j-t)$, *where* $r,t = -1,0,1$ *and* $(r,t) \neq (0,0)$.

2. *Use equations (2.5)-(2.7) to compute* $D_*(i-1,j+1), D_*(i-1,j), D_*(i-1,j-1)$ *and* $D_*(i,j-1)$.

3. *Use equations (2.8), (2.10) and (2.11) to compute the value of* $M_{P(i,j)}$.

4. *Use equations (2.13)-(2.14) or (2.17)-(2.18) to obtain an estimator of* σ *and then use equation (2.12) to obtain a threshold value* $C(\alpha)$. *In most practical applications,* $Z_{[1+(1-\alpha)^{1/4}]/2}$ *can be chosen between 7.5 and 10.*

5. *Compare* $M_{P(i,j)}$ *with* $C(\alpha)$. *If* $M_{P(i,j)} > C(\alpha)$, *then* $P(i,j)$ *is flagged as a step edge pixel.*

We have the following remarks to make on our step edge detection technique:

Remark 2.2: In equations (2.3) and (2.4) we use the unequal weights given by equations (2.1) and (2.2) instead of equal weights because of the following considerations: If we use the unequal weights given by equations (2.1) and (2.2) in equations (2.3) and (2.4), then $std(P^{(2)}(i-1,j+1)) \approx 0.3572\sigma$ and $std(P^{(2)}(i-1,j)) \approx 0.36\sigma$. If we use equal weights in equations (2.3) and (2.4), then $std(P^{(2)}(i-1,j+1)) = std(P^{(2)}(i-1,j)) \approx 0.3333\sigma$. These standard deviation values indicate that the noise removal capability of the technique with equal weights or with unequal weights is almost the same. But the blurring effect with unequal weights will be much less than with equal weights. For a more detailed discussion on the selection of weights we refer the interested reader to Chapter 4, Section 4.3 in the book by Gonzalez and Woods (1992). Another consideration for using unequal weights instead of equal weights is to increase the accuracy of the detected edges. If we use equal weights, then pixels 3 units away (Manhattan distance) from the true edges could still be detected with large probability. If we use the unequal weights in the 9×9 window centered at $P(i,j)$ as we did in equations (2.3)-(2.4), the weighted centers of the SO pixels are closer to $P(i,j)$, thus improving the localization of the detected edges. Pixels about 2 units away from the true edges, in this case, would hardly be detected.

Remark 2.3: We recommend the use of the zeroth-order or the first-order estimator of σ in equation (2.12). Higher order estimators of σ could also be formulated and used but would involve a greater computational overhead. Our experimental results have shown that the first-order estimator provides enough accuracy for most applications because the least-squares plane provides a good fit to the true image intensity surface in a small region where the image intensity function is continuous.

Remark 2.4: The 9×9 mask of $P(i,j)$ is the smallest size mask that contains non-overlapping SO pixels that are eight-neighbors of the central SO pixel. This simplifies our theoretical analysis.

Remark 2.5: In the model, we assumed that noise samples $n(i,j)$ are independent and identically distributed and have an identical variance σ^2. In many applications, however, the noise samples might be correlated and their variances may be a function of the spatial coordinates (i,j) (i.e. non-stationary noise). In the latter case (i.e. non-stationary noise), our method should still work well since the estimators of the variance that we propose are *locally* constructed. We do use the assumption of independence of the noise samples in equation (2.9). If noises are correlated, the coefficients in equation (2.9) would need to be modified accordingly where the modification depends on the pattern of correlation. For a detailed discussion on the estimation of spatial correlation we refer the interested reader to Cressie (1991).

Comparing with some "traditional" edge detectors such as the Sobel, Roberts and Laplacian of Gaussian (LoG), our edge detection criterion has the following features:

1. The variation of the underlying intensity function is almost deleted from the edge detection criterion by using the correction terms. More specifically, $M_{P(i,j)}$ has the property that
$$M_{P(i,j)} \simeq \begin{cases} C(i,j) + \frac{h^2}{2} g''(i,j), & \text{if the } (i,j)\text{th pixel is an edge pixel} \\ \frac{h^2}{2} g''(i,j), & \text{otherwise} \end{cases}$$
where $C(i,j)$ denotes the step magnitude of the intensity function at pixel (i,j) and h is the window width (i.e. window of size $h \times h$). The effect of the variation of the underlying intensity function on the edge detection is reflected by the second order term $\frac{h^2}{2} g''(i,j)$. For most other existing edge detectors, this effect is approximately $hg'(i,j) + \frac{h^2}{2} g''(i,j)$. Thus the linear term $hg'(i,j)$ has been eliminated from our edge detection criterion.

2. We consider four directions instead of only the x and y directions in the construction of the edge detection criterion. That makes the procedure more flexible in its capability of dealing with the local smoothness of the image intensity function.

3. The correction terms result in the introduction of noise in edge detection process. This noisy effect is, however, greatly reduced by introducing the concepts of FO and SO pixels and by

using local smoothing in each SO. The variance of the noise in the gray level of each SO is about 1/9 the variance of the noise in each FO.

The experimental results in the following section will clarify the above remarks.

3 Experimental Results for the Step Edge Detection Algorithm

In this section, we present experimental results of our step edge detection algorithm when applied to gray scale intensity images. We also discuss the effects of various parameters on the performance of the algorithm. The gray scale image of size 512×512 pixels with a gray scale resolution of 8 bits per pixel shown in Fig. 3.1 was used for the purpose of testing our algorithm. We used our step edge detection algorithm to detect step intensity edges within this image. The output edge image is a binary image; if pixel $P(i,j)$ is flagged as a step edge pixel, then we set its gray level value to 255 else we reset its gray level value to 0. Fig. 3.2 (a) – (d) shows the result of our algorithm with four different threshold values $Z_{[1+(1-\alpha)^{1/4}]/2} = 3.5,\ 7.5,\ 10,\ 15$ using the first-order estimator of σ. These results show that the step edge images are best detected with the threshold values in the range $[7.5, 10]$. If the threshold value is too small, then many false edges are detected. On the other hand, if the threshold value is too large, then many true edges are missed. Based on a number of experiments, we recommend threshold values between 7.5 and 10. Fig. 3.3 (a)–(d) shows two other gray scale images and the corresponding edge images wherein the step edges were detected with a threshold value of 8.

Fig. 3.1 goes here

Fig. 3.2 (a), (b), (c), (d) goes here

Fig. 3.3 (a), (b), (c), (d) goes here

We also experimented with the zeroth-order estimator of σ and the results are shown in Fig. 3.4 (a)–(d). The experimental results show that the threshold values need to be smaller in this case. The reason behind this phenomenon is that the zeroth-order estimator of σ is usually larger than the first-order estimator because the former includes a greater variation of the image intensity function. Consequently the zeroth-order estimator is not as accurate as the first-order estimator resulting in the detection of several false edges.

Fig. 3.4 (a), (b), (c), (d) goes here

Next we study the effect of exclusion of the correction terms $D_*(.,.)$ from equations (2.8). The first-order estimator of σ is used. Accordingly the formulas for η_i in equation (2.10) replaced by the following ones:

$$\begin{aligned}\eta_i &= \xi_i/0.5051, \text{ for } i = 1,3 \\ \eta_i &= \xi_i/0.5092, \text{ for } i = 2,4\end{aligned}$$

The results are shown in Fig. 3.5 (a)–(d). From our experimental results, we make the following observations: Our first observation that very large threshold values need to be chosen. This is a result of the fact that we do not exclude the variations of the intensity function from our criterion $M_{P(i,j)}$ as was done previously thus resulting in a larger value for $M_{P(i,j)}$. The other observation is that the detected edges are very thick.

Fig. 3.5 (a), (b), (c), (d) goes here

Finally, we compare the performance of our edge detection technique and two "classical" edge detectors: the Sobel and LoG. Fig. 3.6 (a) shows a synthesized gray scale image. Pixels on scan lines 1-86 have a gray level value of 155 whereas those on scan lines 87-172 have a gray level value of 165. From scan line 173 to line 212, the gray level values decrease linearly and are described by the linear function $255 - 6 \times (i - 172)$ where i is the scan line number. The image between the scan lines 213 and 512 is divided into two portions. The left portion consists of 8 sections where the gray level values in each section are identical. The gray level values of adjacent sections alternate between 155 and 165. In the right portion, the gray level values on scan lines 213-312 are 100, on scan lines 313-412 are 75 and on scan lines 413-512 are 100 again. White noise from a uniform distribution $U(-10, 10)$ is added to the gray level value of each pixel in the right portion.

Fig. 3.6 (b) shows the result of the Sobel edge detector. In the top portion of the image, it misses the edge between scan lines 86-87 and falsely detects the region between scan lines 173-212. In the region between scan lines 173-212, the gray levels have a steep linear trend, but no edges exist. Since the Sobel edge detector does not exclude the linear trend of the underlying intensity function from edge detection process, it cannot overcome this false-edge detection problem and also detect the true edge between scan lines 86-87 at the same time. In the left portion of the bottom half of the image, there are two diagonal edge boundaries, one vertical edge boundary and one horizontal edge boundary; all of which have the same intensity step magnitudes. However, as we can be seen from Figure 3.6 (b), the Sobel detector could detect two diagonal edge boundaries whereas it misses the other two edge boundaries. The reason behind this is that the Sobel operator only considers two orthogonal directions (x and y) in the edge detection process. The Sobel operator detects the true edge boundaries in the noisy portion (i.e. right lower portion) of the image but also several noisy edge pixels as well.

Fig. 3.6 (c) shows the edges detected by the LoG operator. The LoG operator suffers from some of the same drawbacks as the Sobel operator. Note that although the false-edge detection problem in the region between scan lines 173-212 has been somewhat alleviated, the true edge between scan lines 86-87 has been missed. The LoG operator misses most of the edge boundaries in the noisy portion of the image although it suppresses the noisy edge pixels in doing so. It also does not detect all the eight edge boundaries in the left lower portion of the image.

The result of the proposed edge detection technique is shown in Fig. 3.6 (d). We can see that the false-edge detection problem due to the steep linear trend in image intensity values has been alleviated. All the true step edges in the upper portion of the image are detected. Also, all the 8 edge boundaries at various orientations in the left lower portion of the image are detected. In the noisy portion of the image, the true edge pixels and fewer noisy pixels are detected. Our method however also has its limitations. One, is that the detected edges are quite thick. This is the price we pay for the use of larger windows (9×9) in order to eliminate the effect of the linear variation of the intensity function. The proposed technique also detects noisy edge pixels. These limitations can be overcome by postprocessing techniques that thin the detected edges, connect fragmented edges and eliminate scattered noisy edge points. A few such postprocessing techniques have been described in Chapter 7, Section 7.2 in the book by Gonzalez and Woods (1992). Fig. 3.6 (e) shows the results after the use of the postprocessing procedures proposed by Qiu and Yandell (1995) (these procedures are described in the Appendix) to the results depicted in Fig. 3.6 (d). We can see the improvement in results. The postprocessing results around the intersections of edge boundaries are still not quite satisfactory. This is partly due to the inherent limitations of the postprocessing procedures. For a more detailed discussion on this issue, we refer the interested

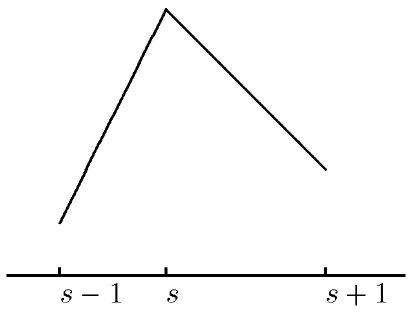


Figure 4.1: Local cross section of an image intensity surface where pixel s is an ideal roof edge pixel.

reader to the recent work of Qiu and Yandell (1995).

Fig. 3.6 (a), (b), (c), (d), (e) goes here

4 Roof Edge Detection

In this section, we discuss the extension of our step edge detection technique to roof edge detection. As already mentioned, a roof edge signifies a discontinuity in the first derivative but continuity in the value of the image intensity function. Fig. 4.1 shows the cross section of the image intensity surface in the vicinity of a roof edge. Pixel s in Fig. 4.1 is an ideal roof edge pixel. From Fig. 4.1, we make the following observations:

(1) $|g'(s+1) - g'(s-1)|$ is large i.e. $g'(\cdot)$ has a step discontinuity at s. Here, $g'(\cdot)$ denotes the first order derivative of the intensity function $g(\cdot)$.

(2) $g'(s-1) \geq g'(s) \geq g'(s+1)$. Furthermore, $g'(s-1) + g'(s+1) - 2g'(s) = 0$ where $g'(s)$ is defined as $g'(s) = \lim_{\Delta s \to 0}[g(s+\Delta s) - g(s-\Delta s)]/2\Delta s$.

The above two observations are the basis of our roof edge detection algorithm.

At a given pixel location $P(i,j)$, we consider the 9×9 mask centered at that pixel (c.f. Fig. 2.1) consisting of nine SO pixels $P^{(2)}(i+s, j+t)$, $s,t = -1, 0, 1$. As in the case of step edge detection, we consider four directions: the two diagonal directions and the two orthogonal directions along the image rows and columns. For the direction from $P(i+3, j-3)$ to $P(i-3, j+3)$, we use d_1 and d_2 which are computed as shown in equations (2.6) to estimate the directional derivatives at pixels $P(i-3, j+3)$ and $P(i+3, j-3)$ respectively. Similarly, we compute

$$\begin{aligned} d_3 &\triangleq \frac{1}{3}[P(i-1,j)+P(i-1,j+1)+P(i,j+1)] \\ &\quad -\frac{1}{3}[P(i,j-1)+P(i+1,j-1)+P(i+1,j)] \end{aligned} \tag{4.1}$$

which is used as an estimator of the first order directional derivative at pixel $P(i, j)$. An additional correction term is used to subtract the variation of the first order directional derivative of the image intensity function along that direction. Let

$$\begin{aligned} c_1 &\triangleq \frac{1}{3}[P(i-4,j+3)+P(i-4,j+4)+P(i-3,j+4) \\ &\quad +P(i-3,j+2)+P(i-2,j+2)+P(i-2,j+3)] \\ &\quad -\frac{2}{3}[P(i-4,j+2)+P(i-3,j+3)+P(i-2,j+4)] \\ c_2 &\triangleq \frac{1}{3}[P(i+2,j-3)+P(i+2,j-2)+P(i+3,j-2) \\ &\quad +P(i+3,j-4)+P(i+4,j-4)+P(i+4,j-3)] \\ &\quad -\frac{2}{3}[P(i+2,j-4)+P(i+3,j-3)+P(i+4,j-2)] \end{aligned} \tag{4.2}$$

Then c_1 and c_2 can be regarded as the estimators of the second order directional derivatives of the image intensity function at $P(i-3, j+3)$ and $P(i+3, j-3)$ respectively. Also

$$c \triangleq \frac{c_1 + c_2}{2} \tag{4.3}$$

can be used as an estimator of the second order directional derivative at $P(i, j)$. We could choose the multiplicative factors as we did in equations (2.7) to delete the quadratic variation of the image intensity function from the process of roof edge detection. Given the complexity of this computation, we suggest using 5 as the multiplicative factor which is the Manhattan distance from $P(i-3, j+3)$ to $P(i+3, j-3)$. We then define

$$\delta_1 \triangleq d_1 - d_2 - 5c \tag{4.4}$$

$$\gamma_1 \triangleq d_1 + d_2 - 2d_3 \tag{4.5}$$

We similarly define the corresponding quantities $\delta_2, \delta_3, \delta_4$ and $\gamma_2, \gamma_3, \gamma_4$ along the other three directions. Clearly, the quantities $\{\delta_1, \delta_2, \delta_3, \delta_4\}$ are a measure of observation (1) made at at the beginning of this section. Similarly, the quantities $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ are a measure of observation (2).

If there are no step edge pixels and roof edge pixels in the 9×9 mask centered at pixel $P(i,j)$, then $\delta_1, \delta_2, \delta_3$ and δ_4 are all approximately normally distributed with zero means and standard deviations σ_δ. From equations (2.6) and (4.2)-(4.4), we can easily compute the value of σ_δ to be $\sigma_\delta = 5.1316\sigma$. By the same reasoning as in the case of step edge detection, if there is some δ_i which satisfies the relation

$$|\delta_i| > Z_{1-\alpha/2} \cdot 5.1316\hat{\sigma} \tag{4.6}$$

then we have sufficient reason to believe that $P(i,j)$ is a roof edge pixel. However, if there are some step edge pixels in the 9×9 mask of $P(i,j)$, they too could result in large values for some of the δ_i's. We therefore need to check an additional condition before making a final decision whether or not $P(i,j)$ is a roof edge pixel. The other condition is derived from the quantities $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$. As pointed out earlier, if $P(i,j)$ is a roof edge pixel, then some of $\{\gamma_i\}$'s should be small (corresponding to observation (2) that $g'(s-1) + g'(s+1) - 2g'(s) = 0$). From equations (2.6), (4.1) and (4.5), we can easily show that $std(\gamma_i) = 2\sigma,\ i = 1,2,3,4$. That γ_i is small can be defined statistically as follows:

$$|\gamma_i| \leq Z_{1-\alpha/2} \cdot 2\hat{\sigma} \tag{4.7}$$

At $P(i,j)$, let us suppose that $\{\delta_{i_1}, \delta_{i_2}, \cdots, \delta_{i_r}\} \subseteq \{\delta_1, \delta_2, \delta_3, \delta_4\}$ contains the δ values which satisfy condition (4.6) where $r \leq 4$. Then we test the corresponding γ values $\{\gamma_{i_1}, \gamma_{i_2}, \cdots, \gamma_{i_r}\}$ using condition (4.7). If at least one of the γ values satisfies condition (4.7), then we decide that $P(i,j)$ is a roof edge pixel. We summarize our roof edge detection technique in the following algorithm.

Roof Edge Detection Algorithm

1. *For a given pixel $P(i,j), 1 \leq i \leq N, 1 \leq j \leq M$, consider the 9×9 mask centered at that pixel. Compute $\{\delta_1, \delta_2, \delta_3, \delta_4\}$ and $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ using equations (2.6) and (4.1)-(4.5).*

2. *Consider the pairs $(\delta_1, \gamma_1), (\delta_2, \gamma_2), (\delta_3, \gamma_3)$ and (δ_4, γ_4) one at a time. If at least one of them satisfies both conditions (4.6) and (4.7), then $P(i,j)$ is flagged as a roof edge pixel.*

We have tested our roof edge detection algorithm on some synthetic images. One such image is shown in Fig. 4.2 (a). Pixels that lie on the horizontal line in the center of the image are assigned a gray level value of 255. The gray level values decrease linearly to zero along the columns of the image on either side of the horizontal line. Thus we have a roof edge along the horizontal line in the center of the image. The resulting edge image from the roof edge detection algorithm is shown in Fig. 4.2 (b). As can be seen, the roof edge is completely detected.

Fig. 4.2 (a), (b) goes here

5 Conclusions

We have proposed algorithms for step edge detection and roof edge detection in this paper. Our algorithms combine local smoothing with statistical hypothesis testing to detect step and roof edges in gray scale images. We have used some locally-weighted averaging procedures to remove the noise and derived the appropriate edge detection criteria thereby giving our algorithms the ability to detect edges in noisy images as well. At each pixel location $P(i,j)$, we consider a 9×9 mask centered at that pixel. This mask is regarded as a union of nine SO pixels with four pairs of SO pixels along each of the four directions – horizontal, vertical and the two diagonal directions. Our algorithms detect the edges along these four directions and are very flexible with regard to the local smoothness properties of the image intensity function. Another important feature of our algorithms is their ability to eliminate the effect of the linear variation of the image intensity function from the process of edge detection which is seen to improve the results of the edge detection process.

The proposed edge detection technique, however, has room for improvement. First, we have used a fixed-size 9×9 mask all throughout. In certain applications, masks with variable or adaptable sizes may be more appropriate. In some planar regions of the image intensity function, we could use large-size masks whereas in highly textured regions, the sizes of the masks could be small. Secondly, in detecting an edge at $P(i,j)$, we use only the pixels in its 9×9 mask in making a decision, i.e. no other information is used. In many cases, some additional information, typically of a contextual nature, may be useful for edge detection. Thirdly, as in the case of some other local smoothing filters, the detected edges resulting from our algorithms may be thick and fragmented at some places. This would call for some global smoothing/sharpening techniques (Acton and Bovik, 1992; Bhandarkar *et al.*, 1994; Tan *et al.*, 1989; 1991) to be used in conjunction with the algorithms presented here. We will address these issues in our future research.

Acknowledgments The authors thank Ms. Peiying Jia, Department of Biological and Agricultural Engineering, University of Georgia, who helped them with the coding and testing of their algorithms. The authors thank the editor and the anonymous referees whose insightful comments greatly improved the paper.

References

Acton, S.T., and Bovik, A.C. (1992). Anisotropic edge detection using mean field annealing, *Proc. IEEE Intl. Conf. Acous. Speech Sig. Proc.* II, 393-396.

Ashkar, G.P., and Modestino, J.W. (1978). The contour extraction problem with biomedical applications. *Comp. Graph. Img. Proc.* 7, 331-355.

Bhandarkar, S.M., Zhang, Y., and Potter, W.D. (1994). An edge detection technique using genetic algorithm-based optimization. *Pattern Recognition.* 27(9), 1159-1180.

Bhattacharya, G.K., and Johnson, R.A. (1977). *Statistical Concepts and Methods.* John Wiley and Sons Inc., New York, NY.

Canny, J.F. (1986). A computational approach to edge detection, *IEEE Trans. Patt. Anal. Mach. Intell.* 8(6), 679-698.

Chen, M.H., Lee, D., and Pavlidis, T. (1991). Residual analysis for feature detection. *IEEE Trans. Patt. Anal. Mach. Intell.* 13(1), 30-40.

Cressie, N. (1991). *Statistics for Spatial Data.* John Wiley Inc., New York, NY.

Dickey, F.M., and Shanmugam, K.S. (1977). Optimum edge detection filter. *Appl. Optics.* 16(1), 145-148.

Gonzalez, R.C., and Woods, R.E. (1992). *Digital Image Processing.* Addison-Wesley Pub. Co. Reading MA.

Hansen, F.R., and Elliot, H. (1982). Image segmentation using simple Markov field models. *Comp. Graph. Img. Proc.* 20, 101-132.

Haralick, R.M. (1984). Digital step edges from zero crossing of second directional derivatives. *IEEE Trans. Patt. Anal. Mach. Intell.* 6(1), 58-68.

Huang, J.S., and Tseng, D.H. (1988). Statistical theory of edge detection. *Comp. Vis. Graph. Img. Proc.* 43, 337-346.

Lee, D., and Wasilkowski, G.W. (1991). Discontinuity detection and thresholding – a stochastic approach. *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* 208-214.

Marr, D., and Hildreth, E. (1980). Theory of edge detection. *Proc. Royal Soc. London.* B 207, 187-217.

Martelli, A. (1976). An application of heuristic search to edge and contour detection. *Comm. ACM.* 19(2), 73-83.

Montanari, U. (1971). On the optimal detection of curves in noisy pictures. *Comm. ACM.* 14(5), 335-345.

Nalwa, V., and Binford, T. (1986). On detecting edges. *IEEE Trans. Patt. Anal. Mach. Intell.* 8(6), 699-714.

Peli, T., and Malah, D. (1982). A study of edge detection algorithms. *Computer Graphics and Image Processing.* 20, 1-21.

Perona, P., and Malik, J. (1990). Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.* 12(7), 629-639.

Qiu, P., and Yandell, B. (1995). Jump detection in regression surfaces. Technical Report 948. Dept. of Statistics, University of Wisconsin-Madison, Madison, WI 53706, USA.

Saint-Marc, P., Chen, J., and Medioni, G. (1991). Adaptive smoothing: a general tool for early vision, *IEEE Trans. Patt. Anal. Mach. Intell.* 13(6), 514-529.

Sarkar, S.S., and Boyer, K.L. (1990). On optimal infinite impulse response edge detection filters. *IEEE Trans. Patt. Anal. Mach. Intell.* 13, 1154-1171.

Shen, J., and Castan, S. (1992). An optimal linear operator for step edge detection. *CVGIP: Graph. Models Img. Proc.* 54(2), 112-133.

Sinha, S.S., and Schunk, B.G. (1992). A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Trans. Patt. Anal. Mach. Intell.* 14, 36-55.

Tan, H.L., Gelfand, S.B., and Delp, E.J. (1989). A comparative cost function approach to edge detection. *IEEE Trans. Sys. Man Cyber.* 19(6), 1337-1349.

Tan, H.L., Gelfand, S.B., and Delp, E.J. (1991). A cost minimization approach to edge detection using simulated annealing. *IEEE Trans. Patt. Anal. Mach. Intell.* 14(1), 3-18.

Torre, V., and Poggio, T.A. (1986). On edge detection. *IEEE Trans. Patt. Anal. Mach. Intell.* 8(2), 147-163.

Wand, M.P., and Jones, M.C. (1995). *Kernel Smoothing.* Chapman & Hall, London, UK.

APPENDIX

1. Derivation of the multiplicative factors in equations (2.7)

The multiplicative factors in equations (2.7) are chosen such that the linear variation of the intensity function is deleted from the edge detection criterion. Let us assume that the intensity function has a linear form $a * i + b * j + c$ at each pixel (i, j), where a, b and c are constants. We then choose the multiplicative factors such that $M_{P(i,j)} = 0$. Without loss of generality, the case of $M_{P(5,5)}$ is discussed in the following example:

In the 9×9 mask of $P(5,5)$, according to equations (2.3)-(2.4),

$$\begin{aligned} P^{(2)}(5-1,5+1) &= \frac{5}{27}P(3,7) + \frac{4}{27}[P(3,8)+P(2,7)] + \frac{3}{27}[P(3,9)+P(2,8)+P(1,7)] \\ &\quad + \frac{2}{27}[P(2,9)+P(1,8)] + \frac{1}{27}P(1,9) \\ &= \frac{5}{27}(3a+7b+c) + \frac{4}{27}(5a+15b+2c) + \frac{3}{27}(6a+24b+3c) + \\ &\quad \frac{2}{27}(3a+17b+2c) + \frac{1}{27}(a+9b+c) \\ &= \frac{20}{9}a + \frac{70}{9}b + c \end{aligned}$$

$$\begin{aligned} P^{(2)}(5+1,5-1) &= \frac{5}{27}P(7,3) + \frac{4}{27}[P(8,3)+P(7,2)] + \frac{3}{27}[P(9,3)+P(8,2)+P(7,1)] \\ &\quad + \frac{2}{27}[P(9,2)+P(8,1)] + \frac{1}{27}P(9,1) \\ &= \frac{5}{27}(7a+3b+c) + \frac{4}{27}(15a+5b+2c) + \frac{3}{27}(24a+6b+3c) + \\ &\quad \frac{2}{27}(17a+3b+2c) + \frac{1}{27}(9a+b+c) \\ &= \frac{70}{9}a + \frac{20}{9}b + c \end{aligned}$$

From equations (2.5) and (2.6),

$$\begin{aligned} D(5-1,5+1) &= \frac{1}{6}[(P(1,8)+P(1,9)+P(2,9)) - (P(2,7)+P(3,7)+P(3,8)) \\ &\quad + (P(7,2)+P(7,3)+P(8,3)) - (P(8,1)+P(9,1)+P(9,2))] \\ &= \frac{1}{6}[(4a+26b+3c) - (8a+22b+3c) + (22a+8b+3c) - (26a+4b+3c)] \\ &= \frac{4}{3}(b-a) \end{aligned}$$

In equations (2.7), let us assume that

$$D_*(5-1,5+1) = m_1 \cdot D(5-1,5+1)$$

where m_1 is the first multiplicative factor. Then from equations (2.8)

$$\begin{aligned}
\xi_1 &= P^{(2)}(5-1,5+1) - P^{(2)}(5+1,5-1) - D_*(5-1,5+1) \\
&= (\frac{20}{9}a + \frac{70}{9}b + c) - (\frac{70}{9}a + \frac{20}{9}b + c) - \frac{4}{3}m_1(b-a)
\end{aligned}$$

If we let $\xi_1 = 0$, then $m_1 = \frac{25}{6}$. Hence the first multiplicative factor should be chosen as $\frac{25}{6}$ in order to make $M_{P(5,5)} = 0$.

Similarly,

$$\begin{aligned}
P^{(2)}(5-1,5) &= \frac{1}{6}[P(3,4)+P(3,5)+P(3,6)] + \frac{1}{9}[P(2,4)+P(2,5)+P(2,6)] + \\
&\quad \frac{1}{18}[P(1,4)+P(1,5)+P(1,6)] \\
&= \frac{1}{6}(9a+15b+3c) + \frac{1}{9}(6a+15b+3c) + \frac{1}{18}(3a+15b+3c) \\
&= \frac{7}{3}a + \frac{15}{3}b + c,
\end{aligned}$$

$$\begin{aligned}
P^{(2)}(5+1,5) &= \frac{1}{6}[P(7,4)+P(7,5)+P(7,6)] + \frac{1}{9}[P(8,4)+P(8,5)+P(8,6)] + \\
&\quad \frac{1}{18}[P(9,4)+P(9,5)+P(9,6)] \\
&= \frac{1}{6}(21a+15b+3c) + \frac{1}{9}(24a+15b+3c) + \frac{1}{18}(27a+15b+3c) \\
&= \frac{23}{3}a + \frac{15}{3}b + c,
\end{aligned}$$

$$\begin{aligned}
D(5-1,5) &= \frac{1}{6}[(P(1,4)+P(1,5)+P(1,6)) - (P(3,4)+P(3,5)+P(3,6)) \\
&\quad +(P(7,4)+P(7,5)+P(7,6)) - (P(9,4)+P(9,5)+P(9,6))] \\
&= \frac{1}{6}[(3a+15b+3c) - (9a+15b+3c) + (21a+15b+3c) - (27a+15b+3c)] \\
&= -2a
\end{aligned}$$

If we denote the second multiplicative factor as m_2 and we let $\xi_2 = 0$, then from equations (2.7) and (2.8), we have $m_2 = \frac{8}{3}$.

2. Derivation of equations (2.9)

From equations (2.5), (2.7) and (2.8),

$$\begin{aligned}
\xi_1 &= P^{(2)}(i-1,j+1) - P^{(2)}(i+1,j-1) - D_*(i-1,j+1) \\
&= P^{(2)}(i-1,j+1) - P^{(2)}(i+1,j-1) - \frac{25}{6}\cdot D(i-1,j+1) \\
&= P^{(2)}(i-1,j+1) - P^{(2)}(i+1,j-1) - \frac{25}{12}d_1 - \frac{25}{12}d_2 \\
&= [P^{(2)}(i-1,j+1) - \frac{25}{12}d_1] - [P^{(2)}(i+1,j-1) + \frac{25}{12}d_2]
\end{aligned}$$

Clearly, $P^{(2)}(i-1,j+1) - \frac{25}{12}d_1$ and $P^{(2)}(i+1,j-1) + \frac{25}{12}d_2$ are independent and have the same variances. So

$$Var(\xi_1) = 2Var[P^{(2)}(i-1,j+1) - \frac{25}{12}d_1].$$

From equations (2.3) and (2.6),

$$\begin{aligned}
& P^{(2)}(i-1,j+1) - \frac{25}{12}d_1 \\
= \; & \frac{5}{27}P(i-2,j+2) + \frac{4}{27}[P(i-2,j+3) + P(i-3,j+2)] \\
& + \frac{3}{27}[P(i-2,j+4) + P(i-3,j+3) + P(i-4,j+2)] \\
& + \frac{2}{27}[P(i-3,j+4) + P(i-4,j+3)] + \frac{1}{27}P(i-4,j+4) \\
& - \frac{25}{36}[P(i-4,j+3) + P(i-4,j+4) + P(i-3,j+4)] \\
& + \frac{25}{36}[P(i-3,j+2) + P(i-2,j+2) + P(i-2,j+3)] \\
= \; & (\frac{5}{27} + \frac{25}{36})P(i-2,j+2) + (\frac{4}{27} + \frac{25}{36})[P(i-2,j+3) + P(i-3,j+2)] \\
& + \frac{3}{27}[P(i-2,j+4) + P(i-3,j+3) + P(i-4,j+2)] \\
& + (\frac{2}{27} - \frac{25}{36})[P(i-3,j+4) + P(i-4,j+3)] + (\frac{1}{27} - \frac{25}{36})P(i-4,j+4)
\end{aligned}$$

So

$$\begin{aligned}
& Var[P^{(2)}(i-1,j+1) - \frac{25}{12}d_1] \\
= \; & [(\frac{5}{27} + \frac{25}{36})^2 + 2(\frac{4}{27} + \frac{25}{36})^2 + 3(\frac{3}{27})^2 + 2(\frac{2}{27} - \frac{25}{36})^2 + (\frac{1}{27} - \frac{25}{36})^2]\sigma^2 \\
= \; & 3.4326\sigma^2
\end{aligned}$$

Hence

$$\begin{aligned}
Var(\xi_1) &= 6.8652\sigma^2 \\
std(\xi_1) &= 2.6202\sigma
\end{aligned}$$

From equations (2.7) and (2.8),

$$\begin{aligned}
\xi_2 &= P^{(2)}(i-1,j) - P^{(2)}(i+1,j) - D_*(i-1,j) \\
&= P^{(2)}(i-1,j) - P^{(2)}(i+1,j) - \frac{8}{3}\cdot D(i-1,j) \\
&= P^{(2)}(i-1,j) - \frac{4}{9}[P(i-4,j-1)+P(i-4,j)+P(i-4,j+1)] \\
&\quad +\frac{4}{9}[P(i-2,j-1)+P(i-2,j)+P(i-2,j+1)] \\
&\quad -P^{(2)}(i+1,j) + \frac{4}{9}[P(i+4,j-1)+P(i+4,j)+P(i+4,j+1)] \\
&\quad -\frac{4}{9}[P(i+2,j-1)+P(i+2,j)+P(i+2,j+1)]
\end{aligned}$$

So

$$\begin{aligned}
Var(\xi_2) &= 2Var[P^{(2)}(i-1,j) - \frac{4}{9}[P(i-4,j-1)+P(i-4,j)+P(i-4,j+1)] + \\
&\quad \frac{4}{9}[P(i-2,j-1)+P(i-2,j)+P(i-2,j+1)]]
\end{aligned}$$

From equation (2.4),

$$\begin{aligned}
& P^{(2)}(i-1,j) - \frac{4}{9}[P(i-4,j-1)+P(i-4,j)+P(i-4,j+1)] \\
& +\frac{4}{9}[P(i-2,j-1)+P(i-2,j)+P(i-2,j+1)] \\
=\ & \frac{1}{6}[P(i-2,j-1)+P(i-2,j)+P(i-2,j+1)] \\
& +\frac{1}{9}[P(i-3,j-1)+P(i-3,j)+P(i-3,j+1)] \\
& +\frac{1}{18}[P(i-4,j-1)+P(i-4,j)+P(i-4,j+1)] \\
& -\frac{4}{9}[P(i-4,j-1)+P(i-4,j)+P(i-4,j+1)] \\
& +\frac{4}{9}[P(i-2,j-1)+P(i-2,j)+P(i-2,j+1)] \\
=\ & (\frac{1}{6}+\frac{4}{9})[P(i-2,j-1)+P(i-2,j)+P(i-2,j+1)] \\
& +\frac{1}{9}[P(i-3,j-1)+P(i-3,j)+P(i-3,j+1)] \\
& +(\frac{1}{18}-\frac{4}{9})[P(i-4,j-1)+P(i-4,j)+P(i-4,j+1)]
\end{aligned}$$

So

$$\begin{aligned}
Var(\xi_2) &= 2\times[(\frac{1}{6}+\frac{4}{9})^2 + (\frac{1}{9})^2 + (\frac{1}{18}-\frac{4}{9})^2]\times 3\sigma^2 \\
&= 3.22222\sigma^2 \\
std(\xi_2) &= 1.7951\sigma
\end{aligned}$$

Clearly,

$$Std(\xi_3) = Std(\xi_1)$$

and

$$Std(\xi_4) = Std(\xi_2)$$

3. Derivation of equation (2.15)

We assume that the gray levels in $P^{(2)}(i-r, j-t)$ are a realization of the following linear model:

$$Y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \epsilon_j,\ j = 1, 2, \cdots, 9$$

where Y_j is the gray level value of the j-th pixel in the SO pixel $P^{(2)}(i-r, j-t)$, (x_{1j}, x_{2j}) is its location within the SO pixel, $\underline{\beta} := (\beta_0, \beta_1, \beta_2)'$ is the parameter vector and $\{\epsilon_j, j = 1, 2, \cdots, 9\}$ are independent and identically distributed (i.i.d.) errors. Then the above model can be expressed in the following matrix form:

$$\underline{Y} = X\underline{\beta} + \underline{\epsilon}$$

where $\underline{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_9 \end{pmatrix}$, $X = \begin{pmatrix} 1, & x_{11}, & x_{21} \\ 1, & x_{12}, & x_{22} \\ \vdots & \vdots & \vdots \\ 1, & x_{19}, & x_{29} \end{pmatrix}$ and $\underline{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_9 \end{pmatrix}$. It is well known that the least squares estimator of $\underline{\beta}$ is $\underline{\hat{\beta}} = (X'X)^{-1}X'\underline{Y}$ where X' is the transpose of X. So the residual vector is

$$\underline{R} = \underline{Y} - X\underline{\hat{\beta}} = \underline{Y} - X(X'X)^{-1}X'\underline{Y}$$

The residual mean squares is

$$\frac{1}{6}\underline{R}'\underline{R} = \frac{1}{6}[\underline{Y} - X(X'X)^{-1}X'\underline{Y}]'[\underline{Y} - X(X'X)^{-1}X'\underline{Y}] = \frac{1}{6}\{\underline{Y}'\underline{Y} - \underline{Y}'[X(X'X)^{-1}X']\underline{Y}\}$$

In the above equations, we have used the matrix property that

$$[X(X'X)^{-1}X'] \cdot [X(X'X)^{-1}X'] = X(X'X)^{-1}X'.$$

Comparing the above expression of the residual mean squares with equation (2.15), we can see that $P_X = X(X'X)^{-1}X'$. Clearly, $I_{9\times 9} - P_X$ is independent of the location of $P^{(2)}(i-r, j-t)$ where

$I_{9\times 9}$ is the 9×9 identity matrix. We can compute $P_X = X(X'X)^{-1}X'$ and after some matrix manipulation, show that P_X has the form given in equation (2.16).

4. Postprocessing procedures in Qiu and Yandell (1995)

We use the notation (x_i, y_j) to denote the location of the (i,j)th pixel. A window centered at (x_i, y_j) with width $k = 2\ell + 1$ is defined by

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

A least squares planar fit in this neighborhood is given by:

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

where $\hat{z}_{ij}(x,y)$ is the least squares estimate of the gray level of the pixel at position (x,y) and $\hat{\beta}_0^{(i,j)}, \hat{\beta}_1^{(i,j)}$ and $\hat{\beta}_2^{(i,j)}$ are the least squares coefficients. $\theta \in [-\pi/4, 7\pi/4]$ is the angle formed by vector $(\hat{\beta}_1^{(i,j)}, \hat{\beta}_2^{(i,j)})$ and the positive direction of the x-axis. Then we propose the following two postprocessing procedures (PP's).

Edge thinning postprocessing procedure PP1: For each $(3k+1)/2 \le j \le n - (3k-1)/2$, consider the pixel positions $\{(x_i, y_j) : (3k+1)/2 \le i \le n - (3k-1)/2\}$ on line $y = y_j$. Among them, the detected edge positions which satisfy $-\pi/4 \le \theta < \pi/4$ or $3\pi/4 \le \theta < 5\pi/4$ are denoted as $\{(x_{i_s}, y_j) : 1 \le s \le n_1\}$. If there are $r_1 < r_2$ such that the increments of the sequence $i_{r_1} < i_{r_1+1} < \cdots < i_{r_2}$ are all less than the window width k, but $i_{r_1} - i_{r_1-1} > k$ and $i_{r_2+1} - i_{r_2} > k$, then we say that $\{(x_{i_s}, y_j) : r_1 \le s \le r_2\}$ forms a *tie* and we select the midpoint $((x_{i_{r_1}} + x_{i_{r_2}})/2, y_j)$ as a new detected edge pixel to replace the tie set. That is, we reduce the tie set to one representative point i.e. the midpoint. Along the y direction, for each $(3k+1)/2 \le i \le n - (3k-1)/2$, consider the pixel positions $\{(x_i, y_j) : (3k+1)/2 \le j \le n - (3k-1)/2\}$ on line $x = x_i$. We carry out the same procedure as the one along the x direction except that this time only those detected edge pixels satisfying the conditions $\pi/4 \le \theta < 3\pi/4$ or $5\pi/4 \le \theta < 7\pi/4$ are considered.

Deletion of scattered noisy edge pixels PP2: For any detected edge position (x_i, y_j), if the number of other detected edge pixels in $N(x_i, y_j)$ is less than $(k-1)/2$, then we delete the pixel at position (x_i, y_j) from the set of detected edge pixels.

Remark A.1 In PP1, along the x direction, the condition $-\pi/4 \le \theta < \pi/4$ or $3\pi/4 \le \theta < 5\pi/4$ implies that the possible edge boundaries form an acute angle $\le \pi/4$ with the y-axis at the detected edge positions. This avoids the deletion of the real edge boundaries that are parallel to the x-axis.

Legends of some figures in Section 3 and Section 4

Figure 3.1: The gray scale image with 512×512 pixels; 8 bits per pixel.

Figure 3.2: Edge images resulting from step edge detection algorithm with different threshold values using first-order estimator of σ and the correction terms (a) $Z_{[1+(1-\alpha)^{1/4}]/2} = 3.5$ (b) $Z_{[1+(1-\alpha)^{1/4}]/2} = 7.5$ (c) $Z_{[1+(1-\alpha)^{1/4}]/2} = 10$ (d) $Z_{[1+(1-\alpha)^{1/4}]/2} = 15$

Figure 3.3: (a) and (c) are two gray scale images; (b) and (d) are the corresponding edge images resulting from the step edge detection algorithm with threshold value $Z_{[1+(1-\alpha)^{1/4}]/2} = 8$ using the first-order estimator of σ and the correction terms.

Figure 3.4: Edge images resulting from step edge detection algorithm on gray scale image in Fig. 3.1 with zeroth-order estimator of σ and the correction terms: (a) $Z_{[1+(1-\alpha)^{1/4}]/2} = 1.5$ (b) $Z_{[1+(1-\alpha)^{1/4}]/2} = 3.5$ (c) $Z_{[1+(1-\alpha)^{1/4}]/2} = 4.0$ (d) $Z_{[1+(1-\alpha)^{1/4}]/2} = 7.5$

Figure 3.5: Edge images resulting from step edge detection algorithm on gray scale image in Fig. 3.1 without correction terms and with the first-order estimator of σ: (a) $Z_{[1+(1-\alpha)^{1/4}]/2} = 4$ (b) $Z_{[1+(1-\alpha)^{1/4}]/2} = 25$ (c) $Z_{[1+(1-\alpha)^{1/4}]/2} = 40$ (d) $Z_{[1+(1-\alpha)^{1/4}]/2} = 60$

Figure 3.6: (a) Original image; (b) Sobel edge image (c) LoG edge image, $\sigma = 1$ (d) Edge image using proposed technique (e) Edge image using the postprocessing procedures in Qiu and Yandell (1995) to the result in (d).

Figure 4.2: (a) Synthetic gray scale image. (b) Output of roof edge detection algorithm.