CHAPTER

Edge detection

5

Edge detection provides an intrinsically more rigorous means than thresholding for initiating image segmentation. However, there is a large history of ad hoc edge detection algorithms, and this chapter aims to distinguish what is principled from what is ad hoc and to provide theory and practical knowledge underpinning available techniques.

Look out for:

- the variety of template matching (TM) operators that have been used for edge detection—e.g., the Prewitt, Kirsch, and Robinson operators
- the differential gradient (DG) approach to edge detection—exemplified by the Roberts, Sobel, and Frei−Chen operators
- theory explaining the performance of the TM operators
- methods for the optimal design of DG operators, and the value of "circular" operators
- tradeoffs between resolution, noise suppression capability, location accuracy, and orientation accuracy
- the distinction between edge enhancement and edge detection
- outlines of more modern operators—the Canny and Laplacian-based operators.

In discussing the process of edge detection, this chapter shows that it is possible to estimate edge orientation with surprising accuracy within a small window—the secret being the considerable information residing in the grayscale values. High orientation accuracy turns out to be of particular value when using the Hough transform to locate extended objects in digital images—as will be seen in several chapters in Part 2, Intermediate-Level Vision, of this book.

5.1 INTRODUCTION

In Chapter 4, The Role of Thresholding, segmentation was tackled by the general approach of finding regions of uniformity in images—on the basis that the areas found in this way would have a fair likelihood of coinciding with the surfaces and facets of objects. The most computationally efficient means of following this approach was that of thresholding, but for real images this turns out to be failure-prone or else quite difficult to implement satisfactorily. Indeed, to make it work well, it seems to require a multiresolution or hierarchical approach, coupled with

Computer Vision. DOI: http://dx.doi.org/10.1016/B978-0-12-809284-2.00005-8

sensitive measures for obtaining suitable local thresholds. Such measures have to take account of local intensity gradients as well as pixel intensities, and the possibility of proceeding more simply—by taking account of intensity gradients alone—was suggested.

In fact, edge detection has long been an alternative path to image segmentation and is the method pursued in this chapter. Whichever way is inherently the better approach, edge detection has the additional advantage in that it immediately reduces by a large factor (typically around 100) the considerable redundancy of most image data: this is useful because it significantly reduces both the space needed to store the information and the amount of processing subsequently required to analyze it.

Edge detection has gone through an evolution spanning well over 30 years. Two main methods of edge detection have been apparent over this period, the first of these being TM and the second being the DG approach. In either case the aim is to find where the intensity gradient magnitude g is sufficiently large to be taken as a reliable indicator of the edge of an object. Then g can be thresholded in a similar way to that in which intensity was thresholded in Chapter 4, The Role of Thresholding (in fact, we shall see that it is possible to look for local maxima of g instead of, or as well as, thresholding it). The TM and DG methods differ mainly in how they proceed to estimate g locally: however, there are also important differences in how they determine local edge orientation, which is an important variable in certain object detection schemes. In Section 5.11 we look at the Canny operator, which was much more rigorously designed than previous edge detectors. Finally, we consider Laplacian-based operators.

Before proceeding to discuss the performance of the various edge detection operators, it will be useful to note that there is a variety of types of edge, including in particular the “sudden step” edge, the “slanted step” edge, the “planar” edge, and various intermediate edge profiles (Fig. 5.1). In most of this chapter we shall be concerned with edges that approximate to types (A)−(D) in Fig. 5.1: later on we shall consider those that approximate to types (E) and (F) in Fig. 5.1.

5.2 BASIC THEORY OF EDGE DETECTION

Both DG and TM operators estimate local intensity gradients with the aid of suitable convolution masks. In the case of the DG type of operator, only two such masks are required—for the x and y directions. In the TM case, it is usual to employ up to 12 convolution masks capable of estimating local components of gradient in the different directions (Prewitt, 1970; Kirsch, 1971; Robinson, 1977; Abdou and Pratt, 1979).

In the TM approach, the local edge gradient magnitude (for short, the edge “magnitude”) is approximated by taking the maximum of the responses for the component masks:

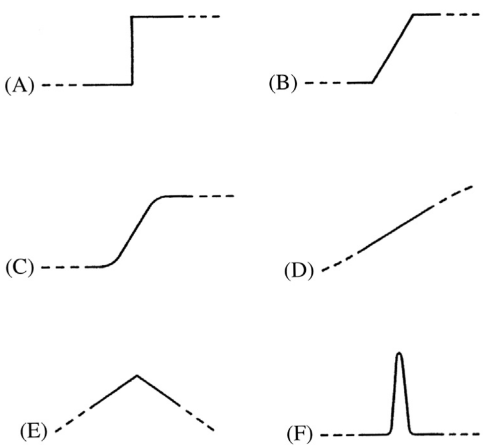


FIGURE 5.1

Edge models: (A) sudden step edge; (B) slanted step edge; (C) smooth step edge; (D) planar edge; (E) roof edge; and (F) line edge. The effective profiles of edge models are nonzero only within the stated neighborhood. The slanted step and the smooth step are approximations to realistic edge profiles: the sudden step and the planar edge are extreme forms that are useful for comparisons (see text). The roof and line edge models are shown for completeness only and are not considered further in this chapter.

$$g = \max(g_i\colon i = 1, \ldots, n) \tag{5.1}$$

where n is usually 8 or 12.

In the DG approach, the local edge magnitude may be computed vectorially using the nonlinear transformation:

$$g = (g_x^2 + g_y^2)^{1/2} \tag{5.2}$$

In order to save computational effort, it is a common practice (Abdou and Pratt, 1979) to approximate this formula by one of the simpler forms:

$$g = |g_x| + |g_y| \tag{5.3}$$

or

$$g = \max(|g_x|, |g_y|) \tag{5.4}$$

which are, on average, equally accurate (Föglein, 1983).

In the TM approach, edge orientation is estimated simply as that of the mask giving rise to the largest value of gradient in Eq. (5.1). In the DG approach, it is estimated vectorially by the more complex equation:

$$\theta = \arctan(g_y / g_x) \tag{5.5}$$

Clearly, DG Eqs. (5.2) and (5.5) require considerably more computation than TM Eq. (5.1), although they are also more accurate. However, in some situations orientation information is not required; in addition, image contrast may vary widely, so there may appear to be little gain from thresholding a more accurate estimate of g. This may explain who so many workers have employed the TM instead of the DG approach. Since both approaches essentially involve estimation of local intensity gradients, it is not surprising that TM masks often turn out to be identical to DG masks (Tables 5.1 and 5.2).

Table 5.1 Masks of Well-known Differential Edge Operators

a. Masks for the Roberts 2 × 2 operator:

$$R_{x'} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad R_{y'} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

b. Masks for the Sobel 3 × 3 operator:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

c. Masks for the Prewitt 3 × 3 "smoothed gradient" operator:

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Masks are presented in an intuitive format (viz. coefficients increasing in the positive x *and* y *directions) by rotating the normal convolution format through 180°. This convention is employed throughout this chapter. The Roberts 2 × 2 operator masks (a) can be taken as being referred to axes* x′, y′ *at 45° to the usual* x, y *axes.*

Table 5.2 Masks of Well-known 3 × 3 Template Matching Edge Operators

	0°	**45°**
a. Prewitt masks	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$
b. Kirsch masks	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$
c. Robinson "three-level" masks	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$
d. Robinson "five-level" masks	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$

The table illustrates only two of the eight masks in each set: the remaining masks can in each case be generated by symmetry operations. For the three-level and five-level operators, four of the eight available masks are inverted versions of the other four (see text).

5.3 THE TEMPLATE MATCHING APPROACH

Table 5.2 shows four sets of well-known TM masks for edge detection. These masks were originally (Prewitt, 1970; Kirsch, 1971; Robinson, 1977) introduced on an intuitive basis, starting in two cases from the DG masks shown in Table 5.1. In all cases the eight masks of each set are obtained from a given mask by permuting the mask coefficients cyclically. By symmetry, this is a good strategy for even permutations, but symmetry alone does not justify it for odd permutations—the situation is explored in more detail below.

Note first that four of the "three-level" and four of the "five-level" masks can be generated from the other four of their set by sign inversion. This means that in either case only four convolutions need to be performed at each pixel neighborhood, thereby saving computation. This is an obvious procedure if the basic idea of the TM approach is regarded as one of comparing intensity gradients in the eight directions. The two operators that do not employ this strategy were developed much earlier on some unknown intuitive basis.

Before proceeding, we note the rationale behind the Robinson "five-level" masks. These were intended (Robinson, 1977) to emphasize the weights of diagonal edges in order to compensate for the characteristics of the human eye, which tends to enhance vertical and horizontal lines in images. Normally, image analysis is concerned with computer interpretation of images and an isotropic set of responses is required. Thus the "five-level" operator is a special purpose one that need not be discussed further here.

These considerations show that the four template operators mentioned above have limited theoretical justification. It is therefore worth studying the situation in more depth: this is done in the next section.

5.4 THEORY OF 3 × 3 TEMPLATE OPERATORS

In the following it is assumed that eight masks are to be used, with angles differing by 45°. In addition, four of the masks differ from the others only in sign, since this seems unlikely to result in any loss of performance. Symmetry requirements then lead to the following masks for 0° and 45°, respectively.

$$\begin{bmatrix} -A & 0 & A \\ -B & 0 & B \\ -A & 0 & A \end{bmatrix} \quad \begin{bmatrix} 0 & C & D \\ -C & 0 & C \\ -D & -C & 0 \end{bmatrix}$$

It is clearly of great importance to design masks, so that they give consistent responses in different directions. To find how this affects the mask coefficients, we employ the strategy of ensuring that intensity gradients follow the rules of vector addition. If the pixel intensity values within a 3 × 3 neighborhood are

$$\boxed{\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array}}$$

then the above masks will give the following estimates of gradient in the 0°, 90°, and 45° directions:

$$g_0 = A(c + i - a - g) + B(f - d) \tag{5.6}$$

$$g_{90} = A(a + c - g - i) + B(b - h) \tag{5.7}$$

$$g_{45} = C(b + f - d - h) + D(c - g) \tag{5.8}$$

If vector addition is to be valid, then:

$$g_{45} = (g_0 + g_{90})/\sqrt{2} \tag{5.9}$$

Equating coefficients of a, b, …, i leads to the self-consistent pair of conditions:

$$C = B/\sqrt{2} \tag{5.10}$$

$$D = A\sqrt{2} \tag{5.11}$$

A further requirement is for the 0° and 45° masks to give equal responses at 22.5°. This can be shown to lead to the formula

$$B/A = \sqrt{2}\,\frac{9t^2 - \left(14 - 4\sqrt{2}\right)t + 1}{t^2 - \left(10 - 4\sqrt{2}\right)t + 1} \tag{5.12}$$

where $t = \tan 22.5°$, so that

$$B/A = \left(13\sqrt{2} - 4\right)/7 = 2.055 \tag{5.13}$$

We can now summarize our findings with regard to the design of TM masks. First, obtaining sets of masks by permuting coefficients "cyclically" in a square neighborhood is ad hoc and cannot be relied upon to produce useful results. Next, following the rules of vector addition and the need to obtain consistent responses in different directions, we have shown that ideal TM masks need to closely match the Sobel coefficients; we have also rigorously derived an accurate value for the ratio *B*/*A*.

Having obtained some insight into the process of designing TM masks for edge detection, we next move on to study the design of DG masks.

Table 5.3 Masks for Estimating Components of Gradient in Square Neighborhoods

	$\mathbf{M}_x$	$\mathbf{M}_y$
a. 2 × 2 neighborhood	$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$
b. 3 × 3 neighborhood	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
c. 4 × 4 neighborhood	$\begin{bmatrix} -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -3 & -3 & -3 & -3 \end{bmatrix}$
d. 5 × 5 neighborhood	$\begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 & -2 \end{bmatrix}$

The above masks can be regarded as extended Prewitt masks. The 3 × 3 masks are Prewitt masks, included in this table for completeness. In all cases weighting factors have been omitted in the interests of simplicity, as they are throughout this chapter.

5.5 THE DESIGN OF DIFFERENTIAL GRADIENT OPERATORS

This section studies the design of DG operators. These include the Roberts 2 × 2 operator and the Sobel and Prewitt 3 × 3 operators (Roberts et al., 1965; Prewitt, 1970; for the Sobel operator, see Pringle, 1969; Duda and Hart, 1973, p. 271) (Table 5.1). The Prewitt or "gradient smoothing" type of operator has been extended to larger pixel neighborhoods by Prewitt (1970) and others (Brooks, 1978; Haralick, 1980) (Table 5.3). In these instances the basic rationale is to model local edges by the best fitting plane over a convenient size of neighborhood. Mathematically, this amounts to obtaining suitably weighted averages to estimate slope in the x and y directions. As pointed out by Haralick (1980), the use of equally weighted averages to measure slope in a given direction is incorrect: the proper weightings to use are given by the masks listed in Table 5.3. Thus the Roberts and Prewitt operators are apparently optimal, whereas the Sobel operator is not. This will be discussed in more detail below.

A full discussion of the edge detection problem involves consideration of the accuracy with which edge magnitude and orientation can be estimated when the local intensity pattern cannot be assumed to be planar. In fact, there have been a number of analyses of the angular dependencies of edge detection operators for a step edge approximation. In particular, O'Gorman (1978) considered the variation of estimated versus actual angle resulting from a step edge observed within a square neighborhood (see also Brooks, 1978): note that the case considered was that of a continuum rather than a discrete lattice of pixels. This was found to lead to a smooth variation with angular error varying from zero at 0° and 45° to a maximum of 6.63° at 28.37° (where the estimated orientation was 21.74°), the variation for angles outside this range being replicated by symmetry. Abdou and Pratt (1979) obtained similar variations for the Sobel and Prewitt operators in a discrete lattice, the respective maximum angular errors being 1.36° and 7.38° (Davies, 1984b). It seems that the Sobel operator has angular accuracy that is close to optimal because it is close to being a "truly circular" operator. This point is discussed in more detail below.

5.6 THE CONCEPT OF A CIRCULAR OPERATOR

It was stated above that when step edge orientation is estimated in a square neighborhood, an error of up to 6.63° can result. Such an error does not arise with a planar edge approximation, since fitting of a plane to a planar edge profile within a square window can be carried out exactly. Errors appear only when the edge profile differs from the ideal planar form, within the square neighborhood—with the step edge probably being something of a "worst case."

One way to limit errors in the estimation of edge orientation might be to restrict observation of the edge to a circular neighborhood. In the continuous case this is sufficient to reduce the error to zero for all orientations, since symmetry dictates that there is only one way of fitting a plane to a step edge within a circular neighborhood, assuming that all planes pass through the same central point; the estimated orientation θ is then equal to the actual angle φ. A rigorous calculation along the lines indicated by Brooks (1976), which results in the following formula for a square neighborhood (O'Gorman, 1978):

$$\tan\theta = 2\tan\varphi/(3-\tan^2\varphi) \quad 0° \leq \varphi \leq 45° \tag{5.14}$$

leads to the following formula

$$\tan\theta = \tan\varphi, \quad \text{i.e., } \theta = \varphi \tag{5.15}$$

for a circular neighborhood (Davies, 1984b). Similarly, zero angular error results from fitting a plane to an edge of *any* profile within a circular neighborhood, in the continuous approximation. Indeed, for an edge surface of arbitrary shape, the only problem is whether the mathematical best-fit plane coincides with one that is subjectively desirable (and, if not, a fixed angular correction will be required). Ignoring such cases, the basic problem is how to approximate a circular neighborhood in a digitized image of small dimensions, containing typically 3 × 3 or 5 × 5 pixels.

To proceed systematically, we first recall a fundamental principle stated by Haralick (1980):

> *the fact that the slopes in two orthogonal directions determine the slope in any direction is well known in vector calculus. However, it seems not to be so well known in the image processing community.*

Essentially, appropriate estimates of slopes in two orthogonal directions permit the slope in any direction to be computed. For this principle to apply, appropriate estimates of the slopes have first to be made: if the components of slope are inappropriate, they will not act as components of true vectors and the resulting estimates of edge orientation will be in error. This appears to be the main source of error with the Prewitt and other operators—it is not so much that the components of slope are in any instance incorrect, but rather that they are inappropriate for the purpose of vector computation since *they do not match one another adequately in the required way* (Davies, 1984b).

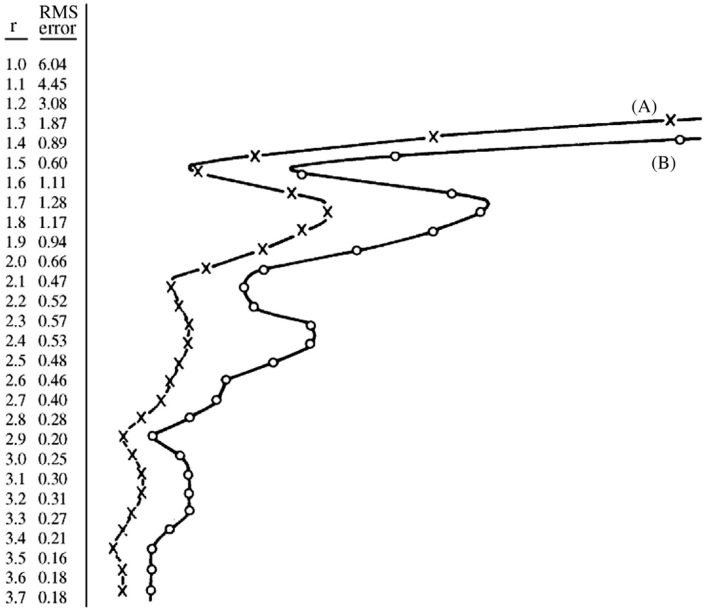
Following the arguments for the continuous case discussed earlier, slopes must be rigorously estimated within a circular neighborhood. Then the operator design problem devolves into determining how best to simulate a circular neighborhood on a discrete lattice so that errors are minimized. In order to carry this out, it is necessary to apply a close to circular weighting while computing the masks, so that correlations between the gradient weighting and circular weighting factors are taken properly into account.

5.7 DETAILED IMPLEMENTATION OF CIRCULAR OPERATORS

In practice, the task of computing angular variations and error curves has to be tackled numerically, dividing each pixel in the neighborhood into arrays of suitably small subpixels. Each subpixel is then assigned a gradient weighting (equal to the x or y displacement) and a neighborhood weighting (equal to 1 for inside and 0 for outside a circle of radius r). Clearly, the angular accuracy of "circular" DG edge detection operators must depend on the radius of the circular neighborhood. In particular, poor accuracy would be expected for small values of r and reasonable accuracy for large values of r, as the discrete neighborhood approaches a continuum.

The results of such a study are presented in Fig. 5.2. The variations depicted represent (1) root mean square (RMS) angular errors and (2) maximum angular errors in the estimation of edge orientation. The structures on each variation are surprisingly smooth: they are so closely related and systematic that they can only represent statistics of the arrangement of pixels in neighborhoods of various sizes. Details of these statistics are discussed in Section 5.8.

Overall, three features of Fig. 5.2 are worthy of note. First, as expected, there is a general trend to zero angular error as r tends to infinity. Second, there is a very marked periodic variation, with particularly good accuracy resulting where the circular operators best match the tessellation of the digital lattice. The third feature of interest is the fact that errors do not vanish for any finite value of r—clearly, the constraints of the problem do not permit more than the minimization of errors. These curves show that it is possible to generate a family of optimal



r	RMS error
1.0	6.04
1.1	4.45
1.2	3.08
1.3	1.87
1.4	0.89
1.5	0.60
1.6	1.11
1.7	1.28
1.8	1.17
1.9	0.94
2.0	0.66
2.1	0.47
2.2	0.52
2.3	0.57
2.4	0.53
2.5	0.48
2.6	0.46
2.7	0.40
2.8	0.28
2.9	0.20
3.0	0.25
3.1	0.30
3.2	0.31
3.3	0.27
3.4	0.21
3.5	0.16
3.6	0.18
3.7	0.18

FIGURE 5.2

Variations in angular error as a function of radius *r*: (A) RMS angular error; (B) maximum angular error.

operators (at the minima of the error curves), the first of which corresponds closely to an operator (the Sobel operator) that is known to be nearly optimal.

The variations shown in Fig. 5.2 can be explained (Davies, 1984b) as pixel centers lying in well packed or "closed" bands approximating to continua—indicated by the low error points in Fig. 5.2—between which centers would be more loosely packed. Thus we get the "closed band" operators listed in Table 5.4: their angular variations appear in Table 5.5. It is seen that the Sobel operator, which is already the most accurate of the 3 × 3 edge gradient operators suggested previously, can be made some 30% more accurate by adjusting its coefficients to make it more circular. In addition, the closed bands idea indicates that the corner pixels of 5 × 5 or larger operators are best removed altogether: not only does this require less computation but also it actually improves performance. It also seems likely that this situation would apply for many other operators and would not be specific to edge detection.

Before leaving this topic, note that the optimal 3 × 3 masks obtained above numerically by consideration of circular operators are very close indeed to those obtained purely analytically in Section 5.4, for TM masks, following the rules of vector addition. In the latter case a value of 2.055 was obtained for the ratio of the two mask coefficients, whereas for circular operators the value

Table 5.4 Masks of "Closed Band" Differential Gradient Edge Operators

a. Band containing shells a–c (effective radius = 1.500)

$$\begin{bmatrix} -0.464 & 0.000 & 0.464 \\ -0.959 & 0.000 & 0.959 \\ -0.464 & 0.000 & 0.464 \end{bmatrix}$$

b. Band containing shells a–e (effective radius = 2.121)

$$\begin{bmatrix} 0.000 & -0.294 & 0.000 & 0.294 & 0.000 \\ -0.582 & -1.000 & 0.000 & 1.000 & 0.582 \\ -1.085 & -1.000 & 0.000 & 1.000 & 1.085 \\ -0.582 & -1.000 & 0.000 & 1.000 & 0.582 \\ 0.000 & -0.294 & 0.000 & 0.294 & 0.000 \end{bmatrix}$$

c. Band containing shells a–h (effective radius = 2.915)

$$\begin{bmatrix} 0.000 & 0.000 & -0.191 & 0.000 & 0.191 & 0.000 & 0.000 \\ 0.000 & -1.085 & -1.000 & 0.000 & 1.000 & 1.085 & 0.000 \\ -0.585 & -2.000 & -1.000 & 0.000 & 1.000 & 2.000 & 0.585 \\ -1.083 & -2.000 & -1.000 & 0.000 & 1.000 & 2.000 & 1.083 \\ -0.585 & -2.000 & -1.000 & 0.000 & 1.000 & 2.000 & 0.585 \\ 0.000 & -1.085 & -1.000 & 0.000 & 1.000 & 1.085 & 0.000 \\ 0.000 & 0.000 & -0.191 & 0.000 & 0.191 & 0.000 & 0.000 \end{bmatrix}$$

In all cases only the x-*mask is shown: the* y-*mask may be obtained by a trivial symmetry operation. Mask coefficients are accurate to* ~0.003 *but would in normal practical applications be rounded to 1- or 2-figure accuracy.*

Table 5.5 Angular Variations for the Best Operators Tested

Actual Angle (degrees)	**Estimated Angle (degrees)**[a]					
	Prew	***Sob***	***a–c***	***circ***	***a–e***	***a–h***
0	0.00	0.00	0.00	0.00	0.00	0.00
5	3.32	4.97	5.05	5.14	5.42	5.22
10	6.67	9.95	10.11	10.30	10.81	10.28
15	10.13	15.00	15.24	15.52	15.83	14.81
20	13.69	19.99	20.29	20.64	20.07	19.73
25	17.72	24.42	24.73	25.10	24.62	25.00
30	22.62	28.86	29.14	29.48	29.89	30.02
35	28.69	33.64	33.86	34.13	35.43	34.86
40	35.94	38.87	39.00	39.15	40.30	39.71
45	45.00	45.00	45.00	45.00	45.00	45.00
RMS error	5.18	0.73	0.60	0.53	0.47	0.19

Key: Prew, *Prewitt.* Sob, *Sobel.* a–c, *theoretical optimum—closed band containing shells a–c.* circ, *actual optimum circular operator (as defined by the first minimum in Fig. 6.2).* a–e, *theoretical optimum—closed band containing shells a–e.* a–h, *theoretical optimum—closed band containing shells a–h.*

[a]*Values are accurate to within ~0.02° in each case.*

$0.959/0.464 = 2.067 \pm 0.015$ is obtained. Clearly this is no accident and it is very satisfying that a coefficient that was formerly regarded as ad hoc (Kittler, 1983) is in fact optimizable and can be obtained in closed form (Section 5.4).

5.8 THE SYSTEMATIC DESIGN OF DIFFERENTIAL EDGE OPERATORS

The family of "circular" DG edge operators studied in Sections 5.6 and 5.7 incorporates only one design parameter—the radius r. Only a limited number of values of this parameter permit optimum accuracy for estimation of edge orientation to be attained.

It is worth considering what additional properties this one parameter can control and how it should be adjusted during operator design. In fact, it affects signal-to-noise ratio, resolution, measurement accuracy, and computational load. To understand this, note first that signal-to-noise ratio varies linearly with the radius of the circular neighborhood, since signal is proportional to area and Gaussian noise is proportional to the square root of area. Likewise, the measurement accuracy is determined by the number of pixels over which averaging occurs and hence is proportional to operator radius. Resolution and "scale" also vary with radius, since relevant linear properties of the image are averaged over the active area of the neighborhood. Finally, computational load, and the associated cost of hardware for speeding up the processing, is generally at least in proportion to the number of pixels in the neighborhood, and hence proportional to r^2.

Overall, the fact that four important parameters vary in a fixed way with the radius of the neighborhood means that there are exact tradeoffs between them, and that improvements in some are only obtained by losses to others: from an engineering point of view compromises between them will have to be made according to circumstances.

5.9 PROBLEMS WITH THE ABOVE APPROACH—SOME ALTERNATIVE SCHEMES

Although the above ideas may be interesting, they have their own inherent problems. In particular, they take no account of the displacement E of the edge from the center of the neighborhood or of the effects of noise in biasing the estimates of edge magnitude and orientation. In fact, it is possible to show that a Sobel operator gives *zero* error in the estimation of step edge orientation under the following condition:

$$|\theta| \le \arctan(1/3) \quad \text{and} \quad |E| \le (\cos\theta - 3\sin|\theta|)/2 \tag{5.16}$$

Furthermore, for a 3 × 3 operator of the form

$$\begin{bmatrix} -1 & 0 & 1 \\ -B & 0 & B \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & B & 1 \\ 0 & 0 & 0 \\ -1 & -B & -1 \end{bmatrix}$$

applied to the edge

$$\begin{array}{|lll|}\hline a & a+h(0.5-E\sec\theta+\tan\theta) & a+h \\ a & a+h(0.5-E\sec\theta) & a+h \\ a & a+h(0.5-E\sec\theta-\tan\theta) & a+h \\ \hline \end{array}$$

Lyvers and Mitchell (1988) found that the estimated orientation is:

$$\varphi = \arctan[2B \tan \theta/(B+2)] \tag{5.17}$$

which immediately shows why the Sobel operator should give zero error for a specific range of θ and E. However, this is somewhat misleading, since considerable errors arise outside this region. Not only do they arise when $E = 0$, as assumed in the foregoing sections but also they vary strongly with E. Indeed, the maximum errors for the Sobel and Prewitt operators rise to 2.90° and 7.43°, respectively, in this more general case (the corresponding RMS errors are 1.20° and 4.50°, respectively). Hence a full analysis should be performed to determine how to reduce the maximum and average errors. Lyvers and Mitchell (1988) carried out an empirical analysis and constructed a lookup table with which to correct the orientations estimated by the Sobel operator, the maximum error being reduced to 2.06°.

Another scheme that reduces the error is the moment-based operator of Reeves et al. (1983). This leads to Sobel-like 3 × 3 masks that are essentially identical to the 3 × 3 masks of Davies (1984b), both having $B = 2.067$ (for $A = 1$). However, the moment method can also be used to estimate the edge position E if additional masks are used to compute second-order moments of intensity. Hence it is possible to make a very significant improvement in performance by using a 2-D lookup table to estimate orientation: the result is that the maximum error is reduced from 2.83° to 0.135° for 3 × 3 masks, and from 0.996° to 0.0042° for 5 × 5 masks.

However, Lyvers and Mitchell (1988) found that much of this additional accuracy is lost in the presence of noise, and RMS standard deviations of edge orientation estimates are already around 0.5° for 3 × 3 operators at 40 db signal-to-noise ratios. The reasons for this are quite simple. Each pixel intensity has a noise component that induces errors in its weighted mask components; the combined effects of these errors can be estimated assuming that they arise independently, so that their variances add (Davies, 1987b). Thus noise contributions to the x and y components of gradient can be computed. These provide estimates for the components of noise along and perpendicular to the edge gradient vector

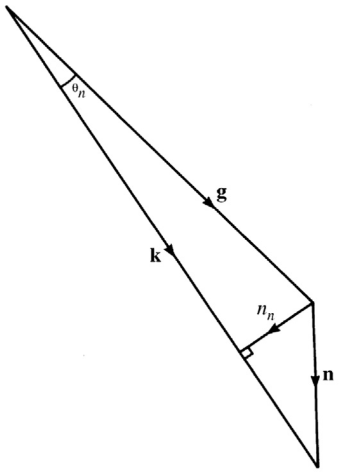


FIGURE 5.3

Calculating angular errors arising from noise: **g**, intensity gradient vector; **n**, noise vector; **k**, resultant of intensity gradient and noise vector; n_n, normal component of noise; θ_n, noise-induced orientation error.

(Fig. 5.3): the edge orientation for a Sobel operator turns out to be affected by an amount $\sqrt{12}\sigma/4h$ radians, where σ is the standard deviation on the pixel intensity values and h is the edge contrast. This explains the angular errors given by Lyvers and Mitchell, if Pratt's (2001) definition of signal-to-noise ratio (in decibels) is used:

$$\text{S/N} = 20 \log_{10}(h/\sigma) \tag{5.18}$$

A totally different approach to edge detection was developed by Canny (1986). He used functional analysis to derive an optimal function for edge detection, starting with three optimization criteria—good detection, good localization, and only one response per edge under white noise conditions. The analysis is too technical to be discussed in detail here. However, the 1-D function found by Canny is accurately approximated by the derivative of a Gaussian: this is then combined with a Gaussian of identical σ in the perpendicular direction, truncated at 0.001 of its peak value, and split into suitable masks. Underlying this method is the idea of locating edges at local maxima of gradient magnitude for a Gaussian-smoothed image. In addition, the Canny implementation employs a hysteresis operation (Section 5.10) on edge magnitude in order to make edges reasonably connected. Finally, a multiple scale method is employed to analyze the output of the edge detector. More details will be said about these points below. Lyvers and Mitchell (1988) tested the Canny operator and found it to be significantly less accurate for orientation estimation than the moment and integrated directional derivative (IDD) operators described above. In addition, it needed to be implemented using 180 masks and hence took enormous computation time, although many practical implementations of this operator are much faster than this early paper indicates. Indeed, it is nowadays necessary to ask "Which Canny?", as there are a great many implementations of it, and this leads to problems for any realistic comparison between operators. One such implementation is described in Section 5.11.

An operator that has been of great historical importance is that of Marr and Hildreth (1980). The motivation for the design of this operator was the modeling of certain psychophysical processes in mammalian vision. The basic rationale is to find the Laplacian of the Gaussian-smoothed ($\nabla^2 G$) image and then to obtain a "raw primal sketch" as a set of zero-crossing lines. The Marr–Hildreth operator does not use any form of threshold since it merely assesses where the $\nabla^2 G$ image passes through zero. This feature is attractive, since working out threshold values is a difficult and unreliable task. However, the Gaussian smoothing procedure can be applied at a variety of scales, and in one sense the scale is a new parameter that substitutes for the threshold. In fact, a major feature of the Marr–Hildreth approach, which has been very influential in later work (Witkin, 1983; Bergholm, 1986), is the fact that zero-crossings can be obtained at several scales, giving the potential for more powerful semantic processing: clearly, this necessitates finding means for combining all the information in a systematic and meaningful way. This may be carried out by bottom-up or top-down approaches, and there has been much discussion in the literature about methods for carrying out these processes. However, it is worth remarking that in many (especially industrial inspection) applications, one is interested in working at a particular resolution and considerable savings in computation can then be made. It is also noteworthy that the Marr–Hildreth operator is reputed to require neighborhoods of at least 35×35 for proper implementation (Brady, 1982). Nevertheless, other workers have implemented the operator in much smaller neighborhoods, down to 5×5. Wiejak et al. (1985) showed how to implement the operator using linear smoothing operations to save computation. Lyvers and Mitchell (1988) reported orientation accuracies using the Marr–Hildreth operator that are not especially high (2.47° for a 5×5 operator and 0.912° for a 7×7 operator, in the absence of noise).

It was noted above that those edge detection operators that are applied at different scales lead to different edge maps at different scales. In such cases, certain edges that are present at lower scales disappear at larger scales; in addition, edges that are present at both low and high scales appear shifted or merged at higher scales. Bergholm (1986) demonstrated the occurrence of elimination, shifting, and merging, while Yuille and Poggio (1986) showed that edges that are present at high resolution should not disappear at some lower resolution. These aspects of edge location are by now well understood.

In what follows we first consider hysteresis thresholding, a process already mentioned with regard to the Canny operator. In Section 5.11 we give a fuller appraisal of the Canny operator and show detailed results on real images. Then in Section 5.12 we consider the Laplacian type of operator.

5.10 HYSTERESIS THRESHOLDING

The concept of hysteresis thresholding is a general one and can be applied in a range of applications, including both image and signal processing. In fact, the Schmitt trigger is a very widely used electronic circuit for converting a varying voltage into a pulsed (binary) waveform. In this latter case there are two thresholds, and the input has to rise above the upper threshold before the output is allowed to switch on and has to fall below the lower threshold before the output is allowed to switch off. This gives considerable immunity against noise in the input waveform—far more than where the difference between the upper and lower switching thresholds is zero (the case of zero hysteresis), since then a small amount of noise can cause an undue amount of switching between the upper and lower output levels.

When the concept is applied in image processing, it is usually with regard to edge detection, in which case there is an exactly analogous 1-D waveform to be negotiated around the boundary of an object, although as we shall see, some specifically 2-D complications arise. The basic rule is to threshold the edge at a high level and then to allow extension of the edge down to a lower level threshold but only adjacent to points that have already been assigned edge status.

Fig. 5.4 shows the results of making tests on the edge gradient image in Fig. 5.4E. Figs. 5.4A and B show the result of thresholding at the upper and lower hysteresis levels, respectively, and Fig. 5.4C shows the result of hysteresis thresholding using these two levels. For comparison, Fig. 5.4D shows the effect of thresholding at a suitably chosen intermediate level. Note that isolated edge points within the object boundaries are ignored by hysteresis thresholding, although noise spurs can occur and are retained. We can envision the process of hysteresis thresholding in an edge image as the location of points that

1. form a superset of the upper threshold edge image;
2. form a subset of the lower threshold edge image; and
3. form that subset of the lower threshold image that is connected to points in the upper threshold image via the usual rules of connectedness (see Chapter 8: Binary Shape Analysis).

Clearly, edge points only survive if they are seeded by points in the upper threshold image.

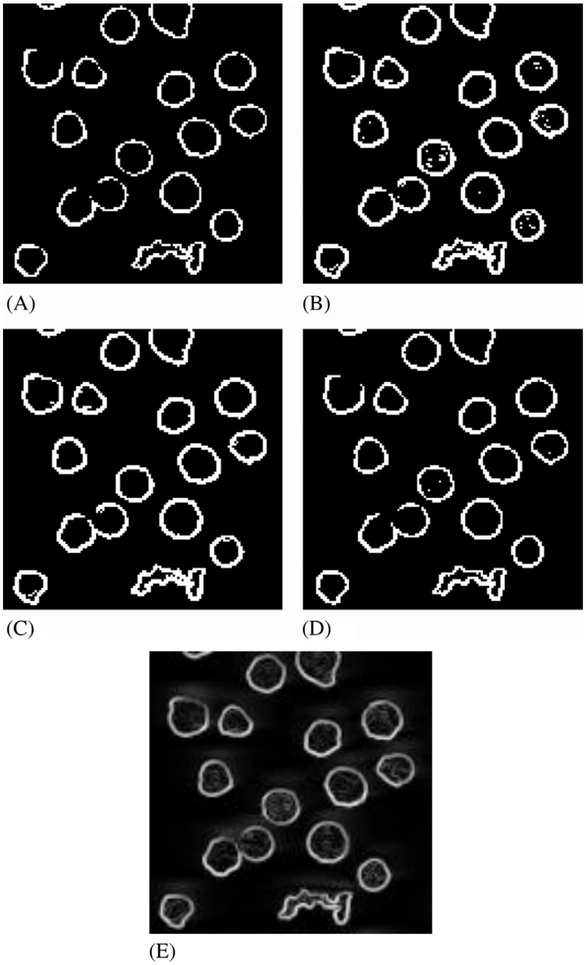


FIGURE 5.4

Effectiveness of hysteresis thresholding. (A) Effect of thresholding at the upper hysteresis level. (B) Effect of thresholding at the lower hysteresis level. (C) Effect of hysteresis thresholding. (D) Effect of thresholding at an intermediate level. This figure shows tests made on the edge gradient image in (E) (the latter was obtained by applying a Sobel operator to the image of Fig. 3.26A).

© World Scientific 2000.

While the result in Fig. 5.4C is better than in Fig. 5.4D, where gaps in the boundaries are eliminated or reduced in length, noise spurs are introduced in a few cases. Nevertheless, the aim of hysteresis thresholding is to obtain a better balance between false positives and false negatives by exploiting connectedness in the object boundaries. Indeed, if managed correctly, the additional parameter will normally lead to a net (average) reduction in boundary pixel classification error. However, there are few simple guidelines for selection of hysteresis thresholds, apart from the following:

1. use a pair of hysteresis thresholds, which provides immunity against the known range of noise levels;
2. choose the lower threshold to limit the possible extent of noise spurs (in principle, the lowest threshold subset that contains *all* true boundary points); and
3. select the upper threshold to guarantee as far as possible the seeding of important boundary points (in principle, the highest threshold subset that is connected to *all* true boundary points).

Unfortunately, in the limit of high signal variability, rules 2 and 3 appear to suggest eliminating hysteresis altogether! Ultimately, this means that the only rigorous way of treating the problem is to perform a complete statistical analysis of false positives and false negatives for a large number of images in any new application.

5.11 THE CANNY OPERATOR

Since its publication in 1986 the Canny operator (Canny, 1986) has become one of the most widely used edge detection operators: there are good reasons for this, as it was aimed at getting away from a tradition of mask-based operators, many of which can hardly be regarded as "designed," into one that is entirely principled and fully integrated. Intrinsic to the method is that of carefully specifying the spatial bandwidth within which it is expected to work, and also the exclusion of unnecessary thresholds, while at the same time permitting thin line structures to emerge, and ensuring that they are connected together as far as possible and indeed are meaningful at the particular scale and bandwidth. As a result of these considerations, the method involves a number of stages of processing:

1. low-pass spatial frequency filtering;
2. application of first-order differential masks;
3. nonmaximum suppression involving subpixel interpolation of pixel intensities; and
4. hysteresis thresholding.

In principle, low-pass filtering needs to be carried out by Gaussian convolution operators for which the standard deviation (or spatial bandwidth) σ is known and prespecified. Then first-order differential masks need to be applied: for this purpose the Sobel operator is acceptable. In this context note that the Sobel operator masks can be regarded as convolutions (⊗) of a basic $[-1 \quad 1]$ type of mask with a $[1 \quad 1]$ smoothing mask. Thus taking the Sobel x-derivative we have:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \tag{5.19}$$

where

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{5.20}$$

and

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{5.21}$$

These equations make it clear that the Sobel operator itself includes a considerable amount of low-pass filtering, so the amount of additional filtering needed in stage 1 can reasonably be reduced. Another thing to bear in mind is that low-pass filtering can itself be carried out by a smoothing mask of the type shown in Fig. 5.5B, and it is interesting how close this mask is to the full 2-D Gaussian shown in Fig. 5.5A. Note also that the bandwidth of the mask in Fig. 5.5B is exactly known (it is 0.707), and when combined with that of the Sobel, the overall bandwidth becomes almost exactly 1.0.

Next, we turn our attention to stage 3, that of nonmaximum suppression. For this purpose, we need to determine the local edge normal direction using Eq. (5.5) and move either way along the normal to determine whether the current location is or is not a local maximum along it. If it is not, we suppress the edge output at the current location, only retaining edge points that are proven local maxima along the edge normal. Since only one point along this direction should be a local maximum, this procedure will necessarily thin the grayscale edges to

(A)

0.000	0.000	0.004	0.008	0.004	0.000	0.000
0.000	0.016	0.125	0.250	0.125	0.016	0.000
0.004	0.125	1.000	2.000	1.000	0.125	0.004
0.008	0.250	2.000	4.000	2.000	0.250	0.008
0.004	0.125	1.000	2.000	1.000	0.125	0.004
0.000	0.016	0.125	0.250	0.125	0.016	0.000
0.000	0.000	0.004	0.008	0.004	0.000	0.000

(B)

1	2	1
2	4	2
1	2	1

FIGURE 5.5

Exactness of the well-known 3 × 3 smoothing kernel. This figure shows the Gaussian based smoothing kernel (A) that is closest to the well-known 3 × 3 smoothing kernel (B) over the central (3 × 3) region. For clarity neither is normalized by the factor 1/16. The larger Gaussian envelope drops to 0.000 outside the region shown and integrates to 18.128 rather than 16. Hence the kernel in (B) can be said to approximate a Gaussian within ~13%. Its actual standard deviation is 0.707 compared with 0.849 for the Gaussian.

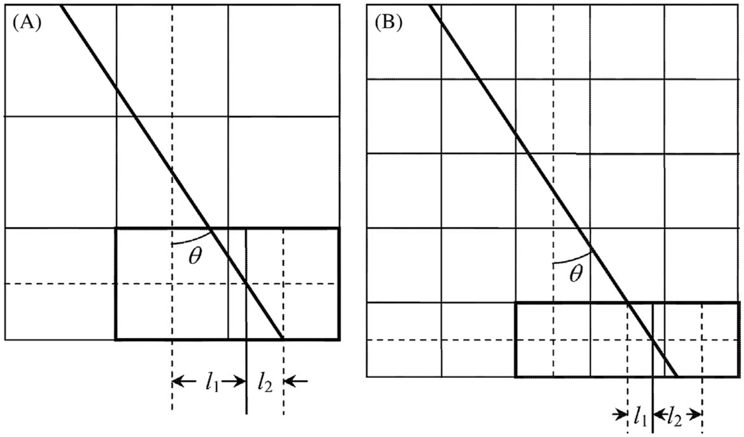


FIGURE 5.6

Pixel interpolation in the Canny operator. (A) Interpolation between the two highlighted pixels at the bottom right in a 3 × 3 neighborhood. (B) Interpolation in a 5 × 5 neighborhood: note that two possibilities exist for interpolating between pairs of adjacent pixels, the relevant distances being marked for the one on the right.

unit width. Here a slight problem arises in that the edge normal direction will in general not pass through the centers of the adjacent pixels, and the Canny method requires the intensities along the normal to be estimated by interpolation. In a 3 × 3 neighborhood this is simply achieved, as the edge normal in any octant will have to lie between a given pair of pixels, as shown in Fig. 5.6A. In a larger neighborhood, interpolation can take place between several pairs of pixels. For example, in a 5 × 5 neighborhood, it will have to be determined which of two pairs is relevant (Fig. 5.6B), and an appropriate interpolation formula applied. However, it could be construed that there is no need to use larger neighborhoods, as a 3 × 3 neighborhood will contain all the relevant information and given enough presmoothing in stage 1—negligible loss of accuracy will result. Of course, if impulse noise is present, this could lead to serious error, but low-pass filtering is in any case has not guaranteed to eliminate impulse noise so there are no special loss results from using the smaller neighborhood for nonmaximum suppression. Such considerations need to be examined carefully in the light of the particular image data and the noise it contains. Fig. 5.6 shows the two distances l_1 and l_2 that have to be determined. The pixel intensity along the edge normal is given by weighting the corresponding pixel intensities in *inverse* proportion to the distances:

$$I = (l_2 I_1 + l_1 I_2)/(l_1 + l_2) = (1 - l_1)I_1 + l_1 I_2 \tag{5.22}$$

where

$$l_1 = \tan\theta \tag{5.23}$$

This brings us to the final stage, that of hysteresis thresholding. By this point, as much as possible has been achieved without applying thresholds and it becomes necessary to take this final step. However, by applying the two hysteresis thresholds, it is intended to limit the damage that can be caused by a single threshold and repair it with another: that is to say, select the upper threshold to ensure capturing edges that are reliable and then select other points that have high likelihood of being viable edge points because they are adjacent to edge points of known reliability. In fact, this is still somewhat ad hoc but in practice it gives quite good results. A simple rule for choice of the lower threshold is that it should be about half the upper threshold. Again, this is only a rule of thumb, and it has to be examined carefully in the light of the particular image data.

Figs. 5.7 and 5.8 show results for the Canny operator at various stages and also comparisons for various thresholds—viz. (E) using hysteresis thresholding, (F) single thresholding at the lower level, and (G) single thresholding at the upper level. The evidence is that hysteresis thresholding is usually more reliable and more coherent than single level thresholding, in the sense of giving fewer false or misleading results.

5.12 THE LAPLACIAN OPERATOR

Whereas an edge detector such as the Sobel is a first derivative operator, the Laplacian is a second-derivative operator, and as such it is sensitive only to changes in intensity gradient. In 2-D its standard (mathematical) definition is given by:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{5.24}$$

Localized masks for computing Laplacians can be derived by taking difference of Gaussian (DoG) kernels using two Gaussians of different bandwidth (for details of this procedure, see Section 6.7.4). This gives them an isotropic 2-D profile, with a positive center and a negative surround. This shape can be approximated in 3 × 3 windows by masks such as the following:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{5.25}$$

Clearly, this mask is far from isotropic: nevertheless it exhibits many of the properties of larger masks, such as DoG kernels, which are much more accurately isotropic.

Here we present only an outline of the properties of this type of operator. These can be seen from Fig. 5.9. First, note that the Laplacian output ranges from positive to negative: hence in Fig. 5.9C it is presented on a medium gray

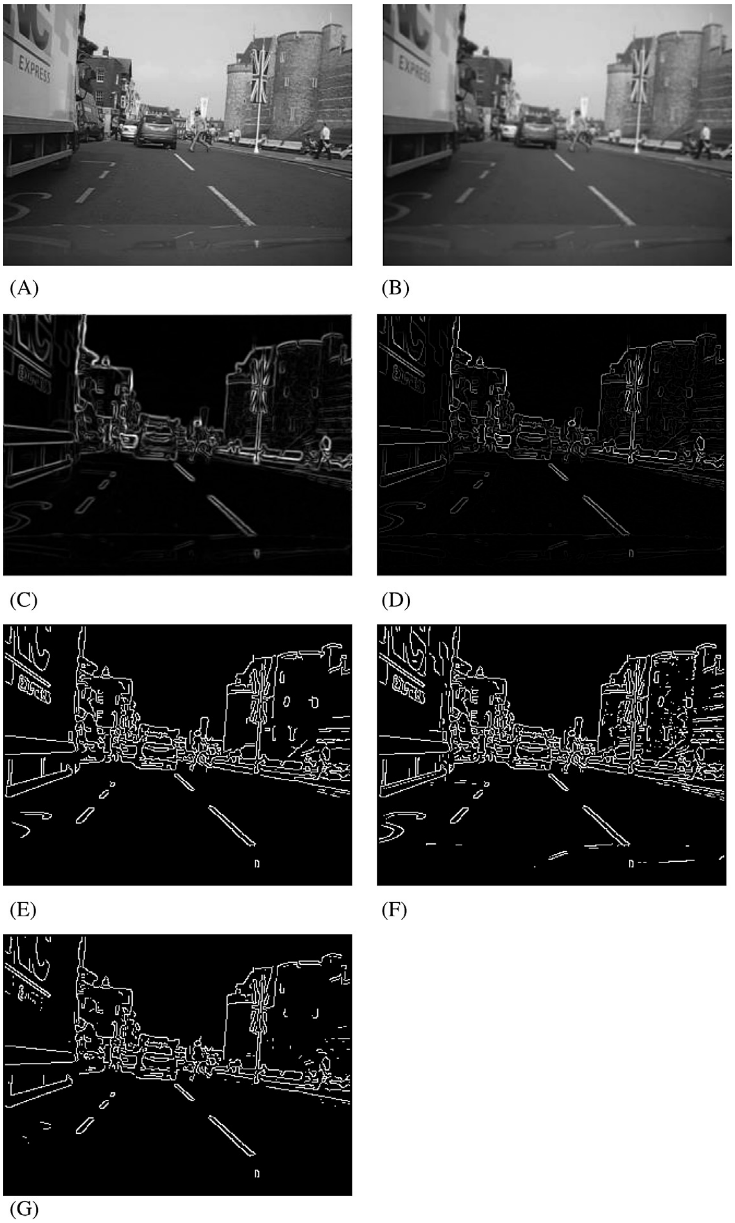


FIGURE 5.7

Application of the Canny edge detector. (A) Original image. (B) Smoothed image. (C) Result of applying Sobel operator. (D) Result of nonmaximum suppression. (E) Result of hysteresis thresholding. (F) Result of thresholding only at the lower threshold level. (G) Result of thresholding at the upper threshold level. Notice that there are fewer false or misleading outputs in (E) than would result from using a single threshold.

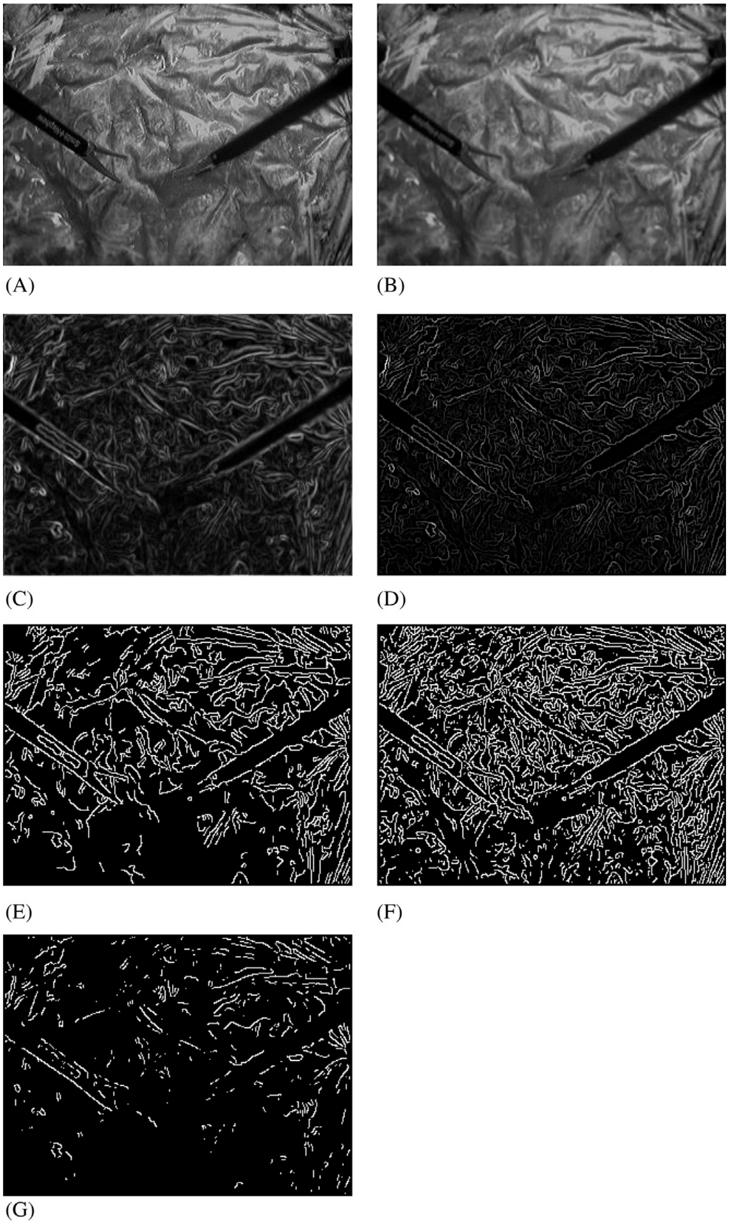


FIGURE 5.8

Another application of the Canny edge detector. (A) Original image. (B) Smoothed image. (C) Result of applying Sobel operator. (D) Result of nonmaximum suppression. (E) Result of hysteresis thresholding. (F) Result of thresholding only at the lower threshold level. (G) Result of thresholding at the upper threshold level. Again there are fewer false or misleading outputs in (E) than would result from using a single threshold.

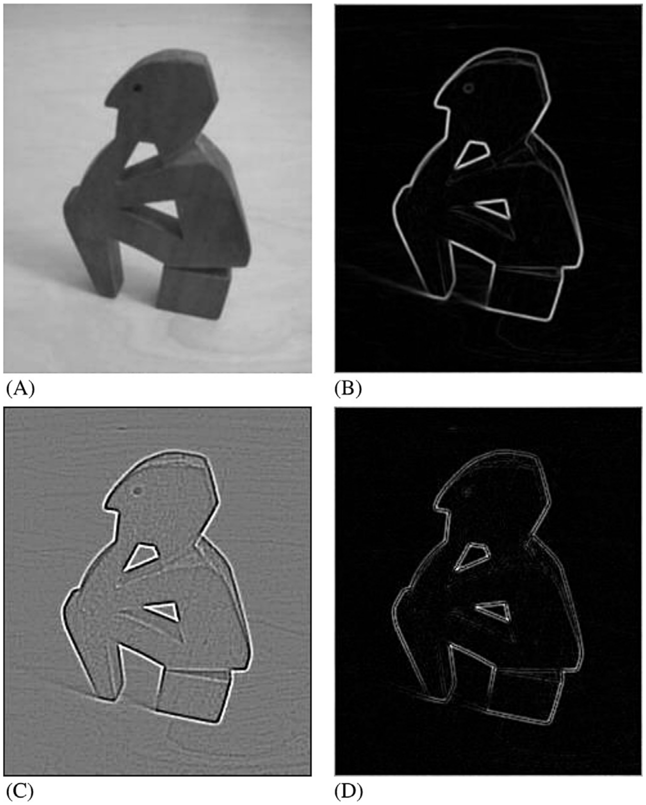


FIGURE 5.9

Comparison of Sobel and Laplacian outputs. (A) Presmoothed version of original image. (B) Result of applying the Sobel operator. (C) Result of applying the Laplacian operator. Because the Laplacian output can be positive or negative, the output in (C) is displayed relative to a medium (128) gray level background. (D) Absolute magnitude Laplacian output. For clarity, (C) and (D) have been presented at increased contrast. Notice that the Laplacian output in (D) gives double edges—one just inside and one just outside the edge position indicated by a Sobel or Canny operator (to find edges using a Laplacian, zero crossings have to be located). Both the Sobel and the Laplacian used here operate within a 3 × 3 window.

background, which indicates that on the exact edge of an object the Laplacian output is actually zero, as stated earlier. This is made clearer, in Fig. 5.9D, where the magnitude of the Laplacian output is shown. It is seen that edges are highlighted by strong signals just inside and just outside the edge locations that are located by a Sobel or Canny operator (Fig. 5.9B). Ideally this effect is symmetrical, and if the Laplacian is to be used for edge detection, zero crossings of the output will have to be located. However, in spite of preliminary smoothing of the image (Fig. 5.9A), the background in Fig. 5.9D has a great deal of noise in the background, and attempting to find zero crossings will therefore lead to a lot of noise being detected in addition to the edge points: in fact, it is well known that differentiation (especially double differentiation, as here) tends to accentuate noise. Nevertheless, this approach has been used highly successfully, usually with DoG operators working in much larger windows. Indeed, with much larger windows there will be a good number of pixels lying very near the zero crossings, and it will be possible to discriminate much more successfully between them and the pixels merely having low Laplacian output. A particular advantage of using Laplacian zero crossings is that theoretically they are bound to lead to closed contours around objects (albeit noise signals will also have their own separate closed contours).

5.13 CONCLUDING REMARKS

The above sections make it clear that the design of edge detection operators has by now been taken to quite an advanced stage, so that edges can be located to subpixel accuracy and orientated to fractions of a degree. In addition, edge maps may be made at several scales and the results correlated to aid image interpretation. Unfortunately, some of the schemes that have been devised to achieve these things are fairly complex and tend to consume considerable computation. In many applications this complexity may not be justified because the application requirements are, or can reasonably be made, quite restricted. Furthermore, there is often the need to save computation for real-time implementation. For these reasons, it will often be useful to explore what can be achieved using a single high-resolution detector such as the Sobel operator, which provides a good balance between computational load and orientation accuracy. Indeed, several of the examples in Part 2, Intermediate-Level Vision, of the book have been implemented using this type of operator, which is able to estimate edge orientation to within about 1°. This does not in any way invalidate the latest methods, particularly those involving studies of edges at various scales: such methods come into their own in applications such as general scene analysis, where vision systems are required to cope with largely unconstrained image data.

This chapter has completed another facet of the task of low-level image segmentation. Later chapters move on to consider the shapes of objects that have been found by the thresholding and edge detection schemes discussed in Chapter 3, Image Filtering and Morphology, and Chapter 4, The Role of Thresholding. In particular, Chapter 8, Binary Shape Analysis, studies shapes by analysis of the regions over which objects extend, while Chapter 9, Boundary Pattern Analysis, studies shapes by considering their boundary patterns.

Edge detection is perhaps the most widely used means of locating and identifying objects in digital images. While different edge detection strategies vie with each other for acceptance, this chapter has shown that they obey fundamental laws—such as sensitivity, noise suppression capability and computation cost all increasing with footprint size.

5.14 BIBLIOGRAPHICAL AND HISTORICAL NOTES

As seen in the first few sections of this chapter, early attempts at edge detection tended to employ numbers of template masks that could locate edges at various orientations. Often these masks were ad hoc in nature, and after 1980 this approach finally gave way to the DG approach that had already existed in various forms for a considerable period (see the influential paper by Haralick, 1980).

The Frei−Chen approach is of interest in that it takes a set of nine 3 × 3 masks forming a complete set within this size of neighborhood—of which one tests for brightness, four test for edges, and four test for lines (Frei and Chen, 1977). Although interesting, the Frei−Chen edge masks do not correspond to those devised for optimal edge detection: Lacroix (1988) makes further useful remarks about the approach.

Meanwhile, psychophysical work by Marr (1976), Wilson and Giese (1977), and others provided another line of development for edge detection. This led to the well-known paper by Marr and Hildreth (1980), which was highly influential in the following few years. This spurred others to think of alternative schemes, and the Canny (1986) operator emerged from this vigorous milieu. In fact, the Marr−Hildreth operator was among the first to preprocess images in order to study them at different scales—a technique that has expanded considerably (see, for example, Yuille and Poggio, 1986), and which will be considered in more depth in Chapter 6, Corner, Interest Point, and Invariant Feature Detection. The computational problems of the Marr−Hildreth operator kept others thinking along more traditional lines, and the work by Reeves et al. (1983), Haralick (1984), and Zuniga and Haralick (1987) fell into this category. Lyvers and Mitchell (1988) reviewed many of these papers and made their own suggestions. Another study (Petrou and Kittler, 1988) carried out further work on operator optimization. The work of Sjöberg and Bergholm (1988), which found rules for discerning shadow edges from object edges, is also of interest.

More recently, there was a move to achieving greater robustness and confidence in edge detection by careful elimination of local outliers: in Meer and Georgescu's (2001) method, this was achieved by estimating the gradient vector, suppressing nonmaxima, performing hysteresis thresholding, and integrating with a confidence measure to produce a more general robust result; in fact, each pixel was assigned a confidence value *before* the final two steps of the algorithm. Kim et al. (2004) took this technique a step further and eliminated the need for setting a threshold by using a fuzzy reasoning approach. Similar sentiments were expressed by Yitzhaky and Peli (2003), and they aimed to find an optimal parameter set for edge detectors by receiver operating characteristic curve and chi-square measures, which actually gave very similar results. Prieto and Allen (2003) designed a similarity metric for edge images, which could be used to test the effectiveness of a variety of edge detectors. They pointed to the fact that metrics need to allow slight latitude in the positions of edges, in order to compare the similarity of edges reliably. They reported a new approach that took into account both displacement of edge positions and edge strengths in determining the similarity between edge images.

Not content with hand-crafted algorithms, Suzuki et al. (2003) devised a back-propagation neural edge enhancer that undergoes supervised learning on model data to permit it to cope well (in the sense of giving clear, continuous edges) with noisy images: it was found to give results superior to those of conventional algorithms (including Canny, Heuckel, Sobel, and Marr-Hildreth) in similarity tests relative to the desired edges. The disadvantage was a long learning time, although the final execution time was short.

5.14.1 MORE RECENT DEVELOPMENTS

Among the most recent developments, Shima et al. (2010) have described the design of more accurate gradient operators on hexagonal lattices. While the latter are not commonly used, there has long been a specialist interest in this area because of the greater number of nearest neighbors at equal distances from a given pixel in a hexagonal lattice: this makes certain types of window operation and algorithm more accurate and efficient, and is particularly useful for edge detection and thinning. Ren et al. (2010) have described an improved edge detection algorithm that operates via the fusion of intensity and chromatic difference, thereby making better use of intercomponent information in color images.

5.15 PROBLEMS

1. Prove Eqs. (5.12) and (5.13).
2. Check the results quoted in Section 5.9 giving the conditions under which the Sobel operator leads to zero error in the estimation of edge orientation. Proceed to prove Eq. (5.17).