CHAPTER

The Hough Transform and Its Nature 13

It has already been seen that the Hough transform can be used to locate straight line, circle, and ellipse features in digital images. It would be useful to know whether the method can be generalized to cover all shapes and whether it is always as robust as it is for the original three examples. This chapter discusses these questions, showing that the method can be generalized and is broadly able to retain its robustness properties.

Look out for:

- the generalized Hough transform technique.
- its relation to spatial matched filtering.
- how sensitivity is optimized by gradient rather than uniform weighting.
- use of the generalized HT for ellipse detection.
- how speed can be improved by the use of a universal lookup table.
- how the computational loads of the various HT techniques can be estimated.
- the value of the Gerig and Klein back-projection technique in cutting down the effects of extraneous clutter.

This chapter describes the generalized Hough transform and generalizes our view of the HT as a generic computer vision technique. It also makes order calculations of computational load for three HT-based methods of ellipse detection.

13.1 INTRODUCTION

In Chapters 11 and 12, it has been seen that the Hough transform (HT) is of great importance for the detection of features such as lines, circles, and ellipses, and for finding relevant image parameters. This makes it worthwhile to see the extent to which the method can be generalized so that it can detect arbitrary shapes. The

Computer and Machine Vision.

works of Merlin and Farber (1975) and Ballard (1981) were crucial historically and led to the development of the generalized Hough transform (GHT). The GHT is studied in this chapter, showing first how it is implemented and then examining how it is optimized and adapted to particular types of image data. This requires us to go back to first principles, taking spatial matched filtering as a starting point.

Having developed the relevant theory, it is applied to the important case of ellipse detection, showing in particular how computational load may be minimized. Finally, the computational problems of the GHT and HT are examined more generally.

13.2 THE GENERALIZED HOUGH TRANSFORM

This section shows how the standard Hough technique is generalized so that it can detect arbitrary shapes. In principle, it is trivial to achieve this. First, we need to select a localization point L within a template of the idealized shape. Then, we need to arrange such that, instead of moving from an edge point a *fixed* distance R directly along the local edge normal to arrive at the center, as for circles, we move an appropriate *variable* distance R in a variable direction φ so as to arrive at L: R and φ are now functions of the local edge normal direction θ (Fig. 13.1). Under these circumstances, votes will peak at the preselected object localization point L. The functions $R(\theta)$ and $\varphi(\theta)$ can be stored analytically in the computer algorithm, or for completely arbitrary shapes they may be stored as lookup tables. In either case, the scheme is beautifully simple in principle but two complications arise in practice. The first arises because some shapes have features such as concavities and holes, so several values of R and φ are required for certain values of θ (Fig. 13.2). The second arises because we are going from an isotropic shape (a circle) to an anisotropic shape, which may be in a completely arbitrary orientation.

To cope with the first of these complications, the lookup table (usually called the "R-table") must contain a list of the positions $\mathbf{r}$, relative to L, of all points on

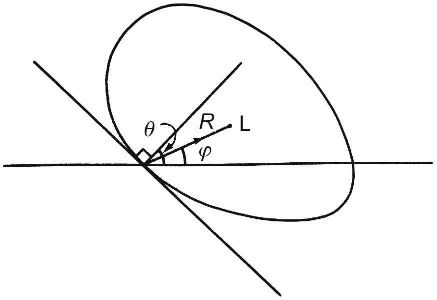

FIGURE 13.1

Computation of the generalized Hough transform.

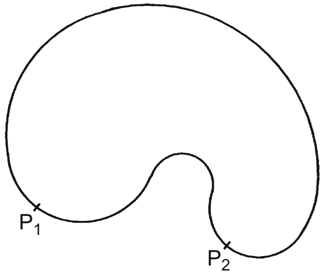

FIGURE 13.2

A shape exhibiting a concavity: certain values of θ correspond to several points on the boundary and hence require several values of R and φ—as for points: P_1 and P_2.

the boundary of the object for each possible value of edge orientation θ (or a similar effect must be achieved analytically). Then, on encountering an edge fragment in the image whose orientation is θ, estimates of the position of L may be obtained by moving a distance (or distances) $\mathbf{R} = -\mathbf{r}$ from the given edge fragment. Clearly, if the R-table has multivalued entries (i.e., several values of $\mathbf{r}$ for certain values of θ), only one of these entries (for given θ) can give a correct estimate of the position of L. However, at least the method is guaranteed to give optimum sensitivity, since all relevant edge fragments contribute to the peak at L in parameter space. This property of optimal sensitivity reflects the fact that the GHT is a form of spatial matched filter: it is analyzed in more detail below.

The second complication arises because any shape other than circle is anisotropic. Since in most applications (including industrial applications such as automated assembly) object orientations are initially unknown, the algorithm has to obtain its own information on object orientation. This means adding an extra dimension in parameter space (Ballard, 1981). Then each edge point contributes a vote in each plane in parameter space at a position given by that expected for an object of given shape and orientation. Finally, the whole of parameter space is searched for peaks, the highest points indicating both the locations of objects and their orientations. Clearly, if object size is also a parameter, the problem becomes far worse but this complication is ignored here (although the method described in Section 12.3 is clearly relevant).

The changes made in proceeding to the GHT leave it just as robust as the HT circle detector described previously. This gives an incentive to improve the GHT so as to limit the computational problems in practical situations. In particular, the size of the parameter space must be cut down drastically both to save storage and to curtail the associated search task. Considerable ingenuity has been devoted to devising alternative schemes and adaptations to achieve this. Important cases are those of ellipse detection and polygon detection, and in each of these, definite advances have been made: ellipse detection is covered in Chapter 12 and for polygon detection, see Davies (1989a). Here we proceed with some more basic studies of the GHT.

13.3 SETTING UP THE GENERALIZED HOUGH TRANSFORM—SOME RELEVANT QUESTIONS

The next few sections explore the theory underpinning the GHT, with the aim of clarifying how to optimize it systematically for specific circumstances. It is relevant to ask what is happening when a GHT is being computed. Although the HT has been shown to be equivalent to template matching (Stockman and Agrawala, 1977) and also to spatial matched filtering (Sklansky, 1978), further clarification is required. In particular, the following three problems (Davies, 1987a) need to be addressed:

1. *The parameter space weighting problem*: In introducing the GHT, Ballard mentioned the possibility of weighing points in parameter space according to the magnitudes of the intensity gradients at the various edge pixels. But when should gradient weighting be used in preference to uniform weighting?
2. *The threshold selection problem*: When using the GHT to locate an object, edge pixels are detected and used to compute candidate positions for the localization point L (see Section 13.2). To achieve this, it is necessary to threshold the edge gradient magnitude. How should the threshold be chosen?
3. *The sensitivity problem*: Optimum sensitivity in detecting objects does not automatically provide optimum sensitivity in locating objects, and vice versa. How should the GHT be optimized for these two criteria?

To understand the situation and solve these problems, it is necessary to go back to first principles. Section 13.4 starts discussion on this.

13.4 SPATIAL MATCHED FILTERING IN IMAGES

To discuss the questions posed in Section 13.3, it is necessary to analyze the process of spatial matched filtering. In principle, this is the ideal method of detecting objects, since it is well known (Rosie, 1966) that a filter that is matched to a given signal detects it with optimum signal-to-noise ratio under white noise[1] conditions (North, 1943; Turin, 1960). (For a more recent discussion of this topic, see Davies (1993).)

Mathematically, using a matched filter is identical to correlation with a signal (or "template") of the same shape as the one to be detected (Rosie, 1966). Here "shape" is a general term meaning the amplitude of the signal as a function of time or spatial location.

[1] White noise is noise that has equal power at all frequencies. In image science, white noise is understood to have equal power at all *spatial* frequencies. The significance of this is that noise at different pixels is completely uncorrelated but is subject to the same grayscale probability distribution, i.e., it has potentially the same range of amplitudes at all pixels.

When applying correlation in image analysis, changes in background illumination cause large changes in signal from one image to another and from one part of an image to another. The varying background level prevents straightforward peak detection in convolution space. The matched filter optimizes signal-to-noise ratio only in the presence of white noise. This is likely to be a good approximation in the case of radar signals, whereas this is not generally true in the case of images. For ideal detection, the signal should be passed through a "noise-whitening filter" (Turin, 1960), which in the case of objects in images is usually some form of high-pass filter: this must be applied prior to correlation analysis. However, this is likely to be a computationally expensive operation.

If we are to make correlation work with near optimal sensitivity but without introducing a lot of computation, other techniques must be employed. In the template matching context, the following possibilities suggest themselves:

1. Adjust templates so that they have a mean value of zero to suppress the effects of varying levels of illumination in first order.
2. Break up templates into a number of smaller templates each having a zero mean. Then as the sizes of subtemplates approach zero, the effects of varying levels of illumination will tend to zero in second order.
3. Apply a threshold to the signals arising from each of the subtemplates so as to suppress those that are less than the expected variation in signal level.

If these possibilities fail, only two further strategies appear to be available:

1. Readjust the lighting system—an important option in industrial inspection applications, although it may give little improvement when a number of objects can cast shadows or reflect light over each other.
2. Use a more "intelligent" (e.g., context sensitive) object detection algorithm, although this will almost certainly be computation intensive.

13.5 FROM SPATIAL MATCHED FILTERS TO GENERALIZED HOUGH TRANSFORMS

To proceed, we note that items 1–3 listed in Section 13.3 essentially amount to a specification of the GHT. First, breaking up the templates into small subtemplates each having a zero mean and then thresholding them are analogous, and in many cases identical, to a process of edge detection (see, e.g., the templates used in the Sobel and similar operators). Next, locating objects by peak detection in parameter space clearly corresponds to the process of reconstructing whole template information from the subtemplate (edge location) data. What is important here is that these ideas reveal how the GHT is related to the spatial matched filter. Basically, *the GHT can be described as a spatial matched filter that has been modified, with the effect of including integral noise whitening, by breaking down* *the main template into small zero-mean templates and thresholding the individual responses before object detection.*

Small templates do not permit edge orientation to be estimated as accurately as large ones. Although the Sobel edge detector is in principle accurate to about 1° (see Chapter 5), there is a deleterious effect if the edge of the object is fuzzy. In such a case, it is not possible to make the subtemplates very small, and an intermediate size should be chosen that gives a suitable compromise between accuracy and sensitivity.

Employing zero-mean templates results in the absolute signal level being reduced to zero and only local relative signal levels being retained. Thus, the GHT is not a true spatial matched filter: in particular, it suppresses the signal from the bulk of the object, retaining only that which is near its boundary. As a result, the GHT is highly sensitive to object position but is not optimized for object detection.

Thresholding of subtemplate responses has much the same effect as employing zero-mean templates, although it may remove a small proportion of the signal giving positional information. This makes the GHT even less like an ideal spatial matched filter and further reduces the sensitivity of object detection. The thresholding process is particularly important in the present context since it provides a means of saving computational effort *without losing significant positional information*. On its own this characteristic of the GHT would correspond to a type of perimeter template around the outside of an object (see Fig. 13.3). This must not be taken as excluding all of the interior of the object, since any high-contrast edges within the object will facilitate location.

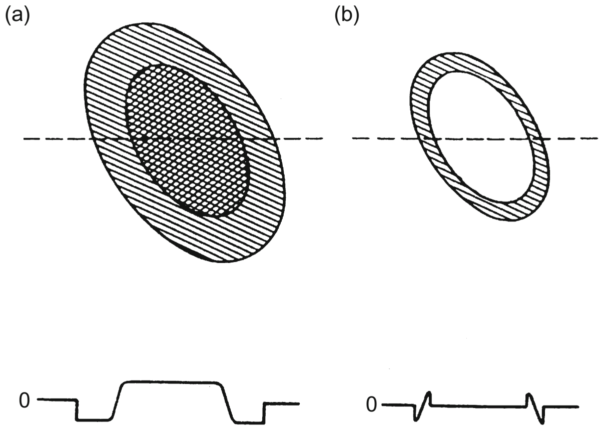

FIGURE 13.3

The idea of a perimeter template: both the original spatial matched filter template (a) and the corresponding "perimeter template" (b) have a zero mean (see text). The lower illustrations show the cross-sections along the dashed lines.

13.6 GRADIENT WEIGHTING VERSUS UNIFORM WEIGHTING

The first problem described in Section 13.3 is that of how best to weight plots in parameter space in relation to the respective edge gradient magnitudes. To find an answer to this problem, it should now only be necessary to go back to the spatial matched filter case to find the ideal solution, and then to determine the corresponding solution for the GHT in the light of the discussion in Section 13.4. First, note that the responses to the subtemplates (or to the perimeter template) are proportional to edge gradient magnitude. With a spatial matched filter, signals are detected optimally by templates of the same shape. Each contribution to the spatial matched filter response is then proportional to the local magnitude of the signal and to that of the template. In view of the correspondence between (a) using a spatial matched filter to locate objects by looking for peaks in convolution space and (b) using a GHT to locate objects by looking for peaks in parameter space, we should use weights proportional to the gradients of the edge points *and* the *a priori* edge gradients.

There are two ways in which the choice of weighting is important. First, the use of uniform weighting implies that all edge pixels whose gradient magnitudes are above threshold will effectively have them reduced to the threshold value, so that the signal will be curtailed: this can mean that the signal-to-noise ratio of high-contrast objects will be reduced significantly. Second, the widths of edges of high-contrast objects will be broadened in a crude way by uniform weighting (see Fig. 13.4) but under gradient weighting this broadening will be controlled, giving a roughly Gaussian edge profile. Thus, the peak in parameter space will be narrower and more rounded, and the object reference point L can be located more easily and with greater accuracy. This effect is visible in Fig. 13.5, which also shows the relatively increased noise level that results from uniform weighting.

Note also that low gradient magnitudes correspond to edges of poorly known location, whereas high values correspond to sharply defined edges. Thus, the accuracy of the information relevant to object location is proportional to the magnitude of the gradient at each of the edge pixels, and appropriate weighting should therefore be used.

13.6.1 Calculation of Sensitivity and Computational Load

The aim of this subsection is to underline the above-described ideas by working out formulae for sensitivity and computational load. It is assumed that p objects of size around $n \times n$ are being sought in an image of size $N \times N$.

Correlation requires N^2n^2 operations to compute the convolutions for all possible positions of the object in the image. Using the perimeter template, the number of basic operations is reduced to $\sim N^2n$, corresponding to the reduced number of pixels in the template. The GHT requires $\sim N^2$ operations to locate the edge pixels, plus a further $\sim pn$ operations to accumulate the points in parameter space.

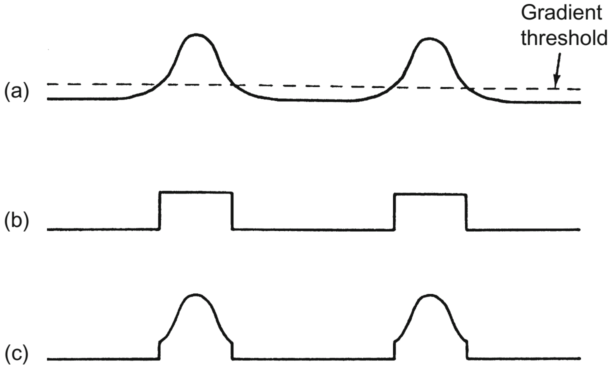

FIGURE 13.4

Effective gradient magnitude as a function of position within a section across an object of moderate contrast, thresholded at a fairly low level: (a) gradient magnitude for original image data and gradient thresholding level; (b) uniform weighting: the effective widths of edges are broadened rather crudely, adding significantly to the difficulty of locating the peak in parameter space; (c) gradient weighting: the position of the peak in parameter space can be estimated in a manner that is basically limited by the shape of the gradient profile for the original image data.

The situation for sensitivity is rather different. With correlation, the results for n^2 pixels are summed, giving a signal proportional to n^2, although the noise (assumed to be independent at every pixel) is proportional to n: this is because of the well-known result that the noise powers of various independent noise components are additive (Rosie, 1966). Overall, this results in the signal-to-noise ratio being proportional to n. The perimeter template possesses only $\sim n$ pixels, and here the overall result is that the signal-to-noise ratio is proportional to $\sqrt{n}$. The situation for the GHT is inherently identical to that for the perimeter template method, so long as plots in parameter space are weighted proportional to edge gradient g multiplied by *a priori* edge gradient G. It is now necessary to compute the constant of proportionality α. Take s as the average signal, equal to the intensity (assumed to be roughly uniform) over the body of the object, and S as the magnitude of a full matched filter template. In the same units, g (and G) is the magnitude of the signal within the perimeter template. Then $\alpha = 1/sS$. This means that the perimeter template method and the GHT method lose sensitivity for two reasons—first they look at less of the available signal and second they look where the signal is low. For a high value of gradient magnitude, which occurs for a step edge (where most of the variation in intensity takes place within the range of 1 pixel), the values of g and G saturate out, so that they are nearly equal to s and S (see Fig. 13.6). Under these conditions, the perimeter template method and the GHT have sensitivities that depend only on the value of n.

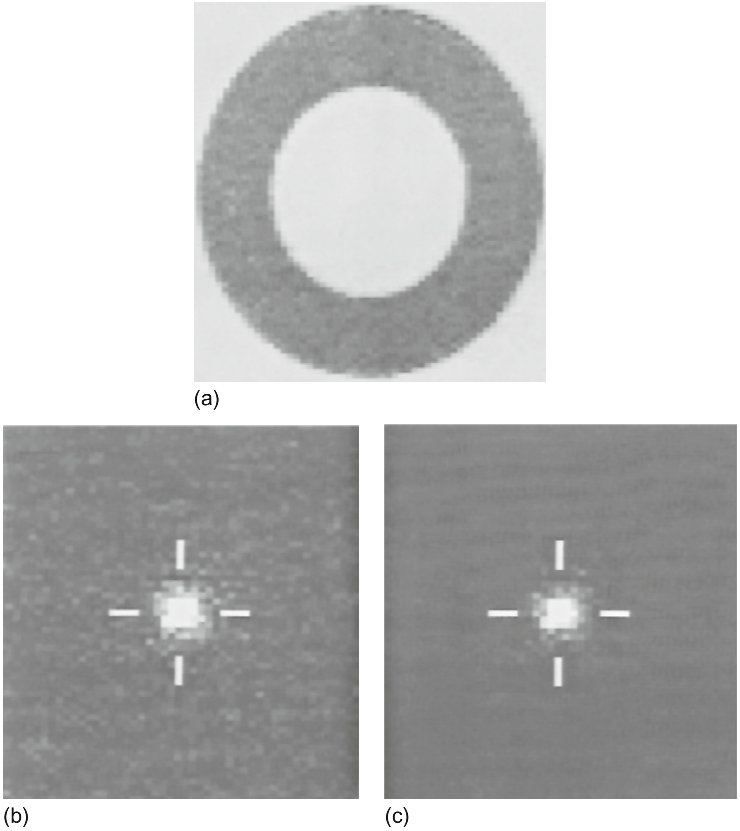

FIGURE 13.5

Results of applying the two types of weighting to a real image: (a) original image; (b) results in parameter space for uniform weighting; and (c) results for gradient weighting. The peaks (which arise from the outer edges of the washer) are normalized to the same levels in the two cases: the increased level of noise in (b) is readily apparent. In this example, the gradient threshold is set at a low level (around 10% of the maximum level) so that low-contrast objects can also be detected.

Table 13.1 summarizes the situation discussed above. The oft-quoted statement that the computational load of the GHT is proportional to the number of perimeter pixels, rather than to the much greater number of pixels within the body of an object, is only an approximation. In addition, this saving is not obtained without cost: in particular, the sensitivity (signal-to-noise ratio) is reduced (at best) as the square root of object area/perimeter (note that area and perimeter are measured in the same units, so it is valid to find their ratio).

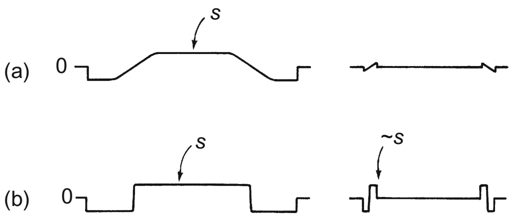

FIGURE 13.6

Effect of edge gradient on perimeter template signal: (a) low edge gradient: signal is proportional to gradient and (b) high edge gradient: signal saturates at value of *s*.

Table 13.1 Formulae for Computational Load and Sensitivity[a]

	Template Matching	Perimeter Template Matching	Generalized Hough Transform
Number of operations	$O(N^2n^2)$	$O(N^2n)$	$O(N^2)+O(pn)$
Sensitivity	$O(n)$	$\mathrm{O}\left(\frac{\sqrt{n}gG}{sS}\right)$	$\mathrm{O}\left(\frac{\sqrt{n}gG}{sS}\right)$
Maximum sensitivity[b]	$O(n)$	$O(\sqrt{n})$	$O(\sqrt{n})$

[a] *This table gives formulae for computational load and sensitivity when p objects of size n × n are sought in an image of size N × N. The intensity of the image within the whole object template is taken as s and the value for the ideal template is taken as S: corresponding values for intensity gradient within the perimeter template are g and G.*
[b] *Maximum sensitivity refers to the case of a step edge, for which g ≈ s and G ≈ S (see Fig. 13.6).*

Finally, the absolute sensitivity for the GHT varies as gG. As contrast changes so that $g \to g'$, we see that $gG \to g'G$, i.e., sensitivity changes by a factor of g'/g. Hence, theory predicts that sensitivity is proportional to contrast. Although this result might have been anticipated, we now see that it is valid only under conditions of gradient weighting.

13.7 SUMMARY

The above sections examined the GHT and found a number of factors involved in optimizing it, as follows.

1. Each point in parameter space should be weighted in proportion to the intensity gradient at the edge pixel giving rise to it, *and* in proportion to the *a*

priori gradient, if sensitivity is to be optimized, particularly for objects of moderate-to-high contrast.

2. The ultimate reason for using the GHT is to save computation. The main means by which this is achieved is by ignoring pixels having low magnitudes of intensity gradient. If the threshold of gradient magnitude is set too high, fewer objects are in general detected; if it is set too low, computational savings are diminished. Suitable means are required for setting the threshold but little reduction in computation is possible if the highest sensitivity in a low-contrast image is to be retained.
3. The GHT is inherently optimized for the location of objects in an image but is not optimized for the detection of objects. This means that it may miss low-contrast objects, which are detectable by other methods that take the whole area of an object into account. However, this consideration is often unimportant in applications where signal-to-noise ratio is less of a problem than finding objects quickly in an uncluttered environment.

Overall, it is clear that the GHT is a spatial matched filter only in a particular sense, and as a result it has suboptimal sensitivity. The main advantage of the technique is that it is highly efficient, overall computational load in principle being proportional to the relatively few pixels on the perimeters of objects rather than to the much greater numbers of pixels within them. In addition, by concentrating on the boundaries of objects, the GHT retains its power to locate objects accurately. It is thus important to distinguish clearly between sensitivity in *detecting* objects and sensitivity in *locating* them.

13.8 USE OF THE GHT FOR ELLIPSE DETECTION

It has already been seen that when the GHT is used to detect anisotropic objects, there is an intrinsic need to employ a large number of planes in parameter space. However, it is shown below that by accumulating the votes for all possible orientations in a *single* plane in parameter space, significant savings in computation can sometimes be made. Basically, the idea is largely to reduce the considerable storage requirements of the GHT by using only one instead of 360 planes in parameter space while significantly reducing the computation involved in the final search for peaks. Such a scheme could have concomitant disadvantages such as the production of spurious peaks, and this aspect will have to be examined carefully.

To achieve these aims, it is necessary to analyze the shape of the point spread function (PSF) to be accumulated for each edge pixel. To demonstrate this, we take the case of ellipses of unknown orientation. We start by taking a general edge fragment at a position defined by ellipse parameter ψ and deducing the bearing of the center of the ellipse relative to the local edge normal (Fig. 13.7).

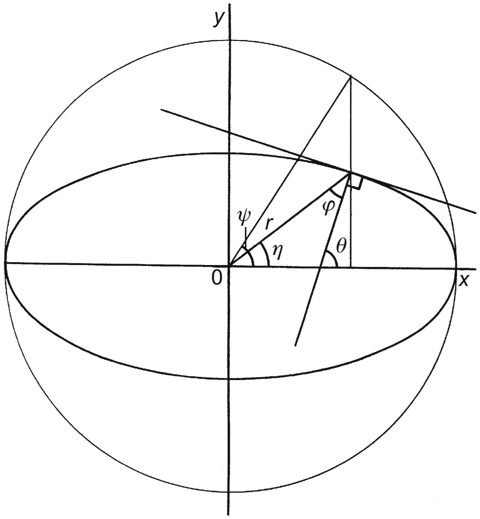

FIGURE 13.7

Geometry of an ellipse and its edge normal.

Working first in an ellipse-based axis system, for an ellipse with semimajor and semiminor axes a and b, respectively, it is clear that:

$$x = a\cos\psi \tag{13.1}$$

$$y = b\sin\psi \tag{13.2}$$

Hence

$$\frac{\mathrm{d}x}{\mathrm{d}\psi} = -a\sin\psi \tag{13.3}$$

$$\frac{\mathrm{d}y}{\mathrm{d}\psi} = b\cos\psi \tag{13.4}$$

giving

$$\frac{\mathrm{d}y}{\mathrm{d}x} = -\left(\frac{b}{a}\right)\cot\psi \tag{13.5}$$

Hence, the orientation of the edge normal is given by:

$$\tan\theta = \left(\frac{a}{b}\right)\tan\psi \tag{13.6}$$

At this point, we wish to deduce the bearing of the center of the ellipse relative to the local edge normal. From Fig. 13.7:

$$\varphi = \theta - \eta \tag{13.7}$$

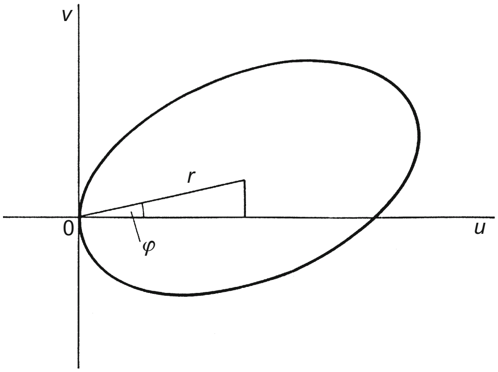

FIGURE 13.8

Geometry for finding the PSF for ellipse detection by forming the locus of the centers of ellipses touching a given edge fragment.

where

$$\tan \eta = \frac{y}{x} = \left(\frac{b}{a}\right) \tan \psi \tag{13.8}$$

and

$$\begin{aligned} \tan \varphi &= \tan(\theta - \eta) \\ &= \frac{\tan \theta - \tan \eta}{1 + \tan \theta \tan \eta} \end{aligned} \tag{13.9}$$

Substituting for tan θ and tan η, and then rearranging, gives:

$$\tan \varphi = \frac{(a^2 - b^2)}{2ab} \sin 2\psi \tag{13.10}$$

In addition:

$$r^2 = a^2 \cos^2 \psi + b^2 \sin^2 \psi \tag{13.11}$$

To obtain the PSF for an ellipse of unknown orientation, we now simplify matters by taking the current edge fragment to be at the origin and orientated with its normal along the u-axis (Fig. 13.8). The PSF is then the locus of all possible positions of the center of the ellipse. To find its form, it is merely required to eliminate ψ between Eqs. (13.10) and (13.11). This is facilitated by re-expressing r^2 in double angles (the significance of double angles lies in the 180° rotation symmetry of an ellipse):

$$r^2 = \frac{a^2 + b^2}{2} + \frac{a^2 - b^2}{2} \cos 2\psi \tag{13.12}$$

After some manipulation, the locus is obtained as:

$$r^4 - r^2(a^2 + b^2) + a^2b^2 \sec^2 \varphi = 0 \tag{13.13}$$

which can, in the edge-based coordinate system, also be expressed in the form:

$$v^2 = (a^2 + b^2) - u^2 - a^2b^2/u^2 \tag{13.14}$$

In fact, this is a complex and variable shape, as shown in Fig. 13.9, although for ellipses of low eccentricity, the PSF approximates to an ellipse. This is seen by defining two new parameters:

$$c = \frac{(a + b)}{2} \tag{13.15}$$

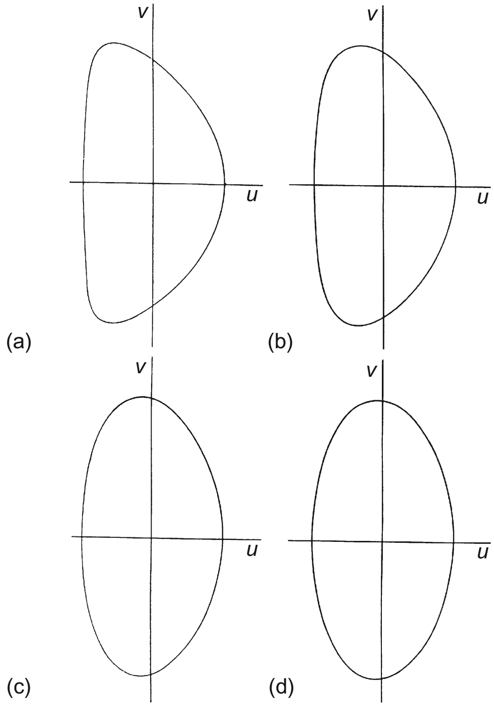

FIGURE 13.9

Typical PSF shapes for detection of ellipses with various eccentricities: (a) ellipse with $a/b = 21.0$ and $c/d = 1.1$, (b) ellipse with $a/b = 5.0$, $c/d = 1.5$, (c) ellipse with $a/b = 2.0$ and $c/d = 3.0$, (d) ellipse with $a/b = 1.4$ and $c/d = 6.0$. Note how the PSF shape approaches a small ellipse of aspect ratio 2.00 as eccentricity tends to zero. The semimajor and semiminor axes of the PSF are $2d$ and d, respectively.

$$d = \frac{(a-b)}{2} \tag{13.16}$$

and taking an approximation of small d, the locus is obtained in the form:

$$\frac{(u+c)^2}{d^2} + \frac{v^2}{4d^2} = 1 \tag{13.17}$$

As stated above, this approximates to an ellipse, and has semimajor axis $2d$ and semiminor axis d. However, this approximation has restricted validity, applying only when $d \gtrsim 0.1c$, so that $a \gtrsim 1.2b$. However, in practice, where ellipses are small and the PSF is only a few pixels across, it is a reasonable approximation to insist only that $a < 2b$.

Implementation is simplified since a universal lookup table (ULUT) for ellipse detection can be compiled, which is independent of the size and eccentricity of the ellipse to be detected so long as the eccentricity is not excessive. Since ellipses are common features in industrial and other applications—arising both from elliptical objects and from oblique views of circles—this factor should be an important consideration in many applications. Thus, a single ULUT is compiled and stored and it then needs only to be scaled and positioned to produce the PSF in a given instance of ellipse detection.

13.8.1 Practical Details

Having constructed a ULUT for ellipse detection, the detection algorithm has to scale it, position it, and rotate it so that points can be accumulated in parameter space. *A priori*, it would be imagined that a considerable amount of trigonometric computation is involved in this process. However, it is possible to avoid calculating angles directly (e.g., using the arctan function) by always working with sines and cosines; this is rendered possible partly because such edge orientation operators as the Sobel give two components (g_x, g_y) for the intensity gradient vector (this has already been seen to happen in several line and circle detection schemes—see Chapters 11 and 12). Hence, a lot of computation can be saved.

Figure 13.10 shows the result of testing the above scheme on an image of some O-rings lying on a slope of arbitrary direction, whereas Fig. 13.11 shows the result obtained for an elliptical object; the two cases used PSFs containing 50 and 100 votes, respectively. In Fig. 13.10, the O-rings are found accurately and with a fair degree of robustness, i.e., despite overlapping and partial occlusion (up to 40% in one case). In several cases, incidental transforms from points on the inner edges of the O-rings overlap other transforms from points on the outer edges, although only the latter are actually employed usefully here for peak finding. Hence, the scheme is able to overcome problems resulting from additional clutter in parameter space.

Figure 13.10 also shows the arrangement of points in parameter space that results from applying the PSF to every edge point on the boundary of an

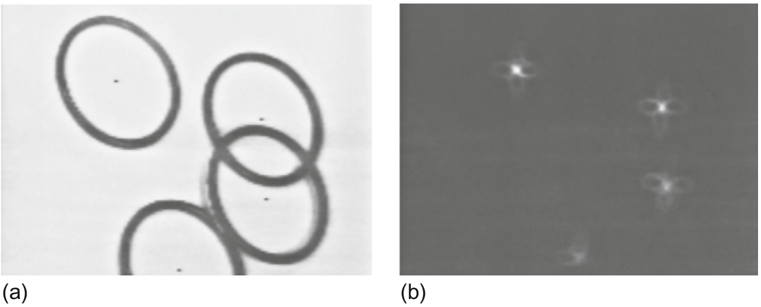

FIGURE 13.10

Applying the PSF to detection of tilted circles: (a) off-camera 128 × 128 image of a set of circular O-rings on a 45° slope of arbitrary direction; (b) transform in parameter space: note the peculiar shape of the ellipse transform, which is close to a "four-leaf clover" pattern. (a) also indicates the positions of the centers of the O-rings as located from (b): accuracy is limited by the presence of noise, shadows, clutter, and available resolution, to an overall standard deviation of about 0.6 pixels.

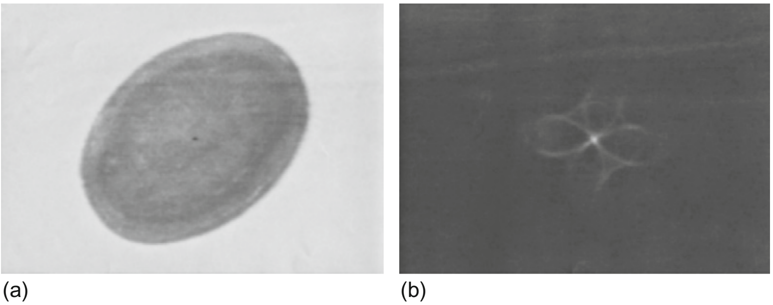

FIGURE 13.11

Applying the PSF to detection of elliptical objects: (a) off-camera 128 × 128 image of an elliptical bar of soap of arbitrary orientation; (b) transform in parameter space: in this case, the clover-leaf pattern is better resolved. Accuracy of location is limited partly by distortions in the shape of the object but the peak location procedure results in an overall standard deviation of the order of 0.5 pixels.

ellipse: the pattern is somewhat clearer in Fig. 13.11. In either case, it is seen to contain a high degree of structure (curiously, the votes seem to form approximate "four-leaved clover" patterns). For an ideal transform, there would be no structure apart from the main peak, and all points on the PSF *not* falling on the peak at the center of the ellipse would be randomly distributed nearby. Nevertheless, the peak at the center is very well defined and shows that this compressed form of GHT represents a viable solution.

13.9 COMPARING THE VARIOUS METHODS

This section briefly compares the computational loads for the methods of ellipse detection discussed in Section 13.8 and in Chapter 12. To make fair comparisons, we concentrate on ellipse detection *per se* and ignore any additional procedures concerned with (a) finding other ellipse parameters, (b) distinguishing ellipses from other shapes, or (c) separating concentric ellipses. We start by examining the GHT method and the diameter bisection method.

First, suppose that an $N \times N$ pixel image contains p ellipses of identical size given by the parameters a, b, c, and d defined in Section 13.8. By ignoring noise and general background clutter, we shall be favoring the diameter bisection method, as will be seen below. Next, the discussion is simplified by supposing that the computational load resides mainly in the calculation of the positions at which votes should be accumulated in parameter space—the effort involved in locating edge pixels and in locating peaks in parameter space is much smaller.

Under these circumstances, the load for the GHT method may be approximated by the product of the number of edge pixels and the number of points per edge pixel that have to be accumulated in parameter space, the latter being equal to the number of points on the PSF. Hence, the load is proportional to:

$$\begin{aligned} L_{\mathrm{G}} &\approx \frac{p \times 2\pi c \times 2\pi(2d+d)}{2} = 6\pi^2 pcd \\ &\approx 60pcd \end{aligned} \tag{13.18}$$

where the ellipse has been taken to have relatively low eccentricity so that the PSF itself approximates to an ellipse of semiaxes $2d$ and d.

For the diameter bisection method, the actual voting is a minor part of the algorithm—as indeed it is in the GHT method (see the snippet of code listed in Table 13.1). In either case, most of the computational load concerns edge orientation calculations or comparisons. Assuming that these calculations and comparisons involve similar inherent effort, it is fair to assess the load for the diameter bisection method as:

$$L_{\mathrm{D}} \approx {}^{p \times 2\pi c}C_2 \approx \frac{(2\pi pc)^2}{2} \approx 20p^2c^2 \tag{13.19}$$

Hence

$$\frac{L_{\mathrm{D}}}{L_{\mathrm{G}}} \approx \frac{pc}{3d} \tag{13.20}$$

when a is close to b, as for a circle, $L_{\mathrm{G}} \rightarrow 0$ and then the diameter bisection method becomes a poor option. However, in some cases, it is found that a is close to $2b$, so that c is close to $3d$. The ratio of the loads then becomes:

$$\frac{L_{\mathrm{D}}}{L_{\mathrm{G}}} \approx p \tag{13.21}$$

It is possible that p will be as low as 1 in some cases: however, such cases are likely to be rare and are offset by applications where there is significant background image clutter and noise, or where all p ellipses have other edge detail giving irrelevant signals that can be considered as a type of self-induced clutter (see the O-ring example in Fig. 13.10).

It is also possible that some of the pairs of edge points in the diameter bisection method can be excluded before they are considered, e.g., by giving every edge point a range of interaction related to the size of the ellipses. This would tend to reduce the computational load by a factor of the order of (but not as small as) p. However, the computational overhead required for this would not be negligible.

Overall, the GHT method should be significantly faster than the diameter bisection method in most real applications, the diameter bisection method being at a definite disadvantage when image clutter and noise are strong. By comparison, the chord−tangent method always requires more computation than the diameter bisection method, since not only does it examine every pair of edge points but also it generates a *line* of votes in parameter space for each pair.

Against these computational limitations, the different characteristics of the methods must be noted. First, the diameter bisection method is not particularly discriminating, in that it locates many symmetrical shapes, as remarked earlier. The chord−tangent method is selective for ellipses but is not selective about their size or eccentricity. The GHT method is selective about all of these factors. These types of discriminability, or lack of it, can turn out to be advantageous or disadvantageous, depending on the application: hence, we do no more here than draw attention to these different properties. It is also relevant that the diameter bisection method is rather less robust than the other methods. This is so since if one edge point of an antiparallel pair is not detected, then the other point of the pair cannot contribute to detection of the ellipse—a factor that does not apply for the other two methods since they take all edge information into account.

13.10 FAST IMPLEMENTATIONS OF THE HOUGH TRANSFORM

The foregoing sections have shown that the GHT requires considerable computation. The problem arises particularly in respect of the number of planes needed in parameter space to accommodate transforms for different object orientations and sizes. Clearly, significant improvements in speed are needed before the GHT can achieve its potential in practical instances of arbitrary shapes. This section considers some important developments in this area.

The source of the computational problems in the HT is the huge size the parameter space can take. Typically, a single plane parameter space has much the same size as an image plane (this will normally be so in those instances where parameter space is congruent to image space), but when many planes are required to cope with various object orientations and sizes, the number of planes is likely to be multiplied by a factor of around 300 for each extra dimension. Hence, the total storage area will then involve some 10,000 million accumulator cells. Clearly, reducing the resolution might just make it possible to bring this down to 100 million cells, although accuracy will start to suffer. Further, when the HT is stored in such a large space, the search problem involved in locating significant peaks becomes formidable.

Fortunately, data are not stored at all uniformly in parameter space and this provides a key to solving both the storage problem and the subsequent search problem. Indeed, the fact that parameter space is to be searched for the most prominent peaks—in general, the highest and the sharpest ones—means that the process of detection can start during accumulation. Furthermore, initial accumulation can be carried out at relatively low resolution, and then resolution can be increased locally as necessary in order to provide both the required accuracy and the separation of nearby peaks. In this context, it is natural to employ a binary splitting procedure, i.e., to repeatedly double the resolution locally in each of the dimensions of parameter space (e.g., the *x* dimension, the *y* dimension, the orientation dimension, and the size dimension, where these exist) until resolution is sufficient: a tree structure may conveniently be used to keep track of the situation.

Such a method (the "fast" HT) was developed by Li and Lavin (1986) and Li et al. (1985). Illingworth and Kittler (1987) found this method to be insufficiently flexible in dealing with real data and produced a revised version (the "adaptive" HT), which permits each dimension in parameter space to change its resolution locally in tune with whatever the data demand, rather than insisting on some previously devised rigid structure. In addition, they employed a 9×9 accumulator array at each resolution rather than the theoretically most efficient 2×2 array, since this was found to permit better judgments to be made on the nature of the local data. This approach seemed to work well with fairly clean images but later, doubts were cast on its effectiveness with complex images (Illingworth and Kittler, 1988). The most serious problem to be overcome here is that at coarse resolutions, extended patterns of votes from several objects can overlap, giving rise to spurious peaks in parameter space. Since all of these peaks have to be checked at all relevant resolutions, the whole process can consume more computation than it saves. Clearly, optimization in multiresolution peak-finding schemes is complex[2] and data-dependent, and so discussion is curtailed here. The reader is

[2] Ultimately, the problem of system optimization for analysis of complex images is a difficult one, since in conditions of low signal-to-noise ratio even the eye may find it difficult to interpret an image and may "lock on" to an interpretation that is incorrect. Note that in general image interpretation work, there are many variables to be optimized—sensitivity, efficiency/speed, storage, accuracy, robustness, and so on—and it is seldom valid to consider any of these individually. Often tradeoffs between just two such variables can be examined and optimized but in real situations multivariable tradeoffs should be considered. This is a complex task and it is one of the purposes of this book to show clearly the serious nature of these types of optimization problem, although at the same time it can only guide the reader through a limited number of basic optimization processes.

referred to the original research papers (Li et al., 1985; Li and Lavin, 1986; Illingworth and Kittler, 1987) for implementation details.

An alternative scheme is the hierarchical HT (Princen et al., 1989a); so far it has been applied only to line detection. The scheme can most easily be envisaged by considering the foot-of-normal method of line detection described in Section 11.3. Rather small subimages of just 16×16 pixels are taken first of all, and the foot-of-normal positions determined. Then each foot-of-normal is tagged with its orientation and an identical HT procedure is instigated to generate the foot-of-normal positions for line segments in 32×32 subimages, this procedure being repeated as many times as necessary until the whole image is spanned at once. The paper by Princen et al. discusses the basic procedure in detail and also elaborates necessary schemes for systematically grouping separate line segments into full-length lines in a hierarchical context. An interesting detail is that successful operation of the method requires subimages with 50% overlap to be employed at each level. The overall scheme appears to be as accurate and reliable as the basic HT method but does appear to be faster and to require significantly less storage.

13.11 THE APPROACH OF GERIG AND KLEIN

The Gerig and Klein approach was first demonstrated in the context of circle detection but was only mentioned in passing in Chapter 12. This is because it is an important *approach* that has much wider application than merely to circle detection. The motivation for it has already been noted in Section 13.10—namely, the problem of extended patterns of votes from several objects giving rise to spurious peaks in parameter space.

Ultimately, the reason for the extended pattern of votes is that each edge point in the original image can give rise to a large number of votes in parameter space. The tidy case of detection of circles of known radius is somewhat unusual, as will be seen particularly in Chapters 12 and 17. Hence, in general *most* of the votes in parameter space are in the end unwanted and serve only to confuse. Ideally, we would like a scheme in which each edge point gives rise only to the single vote corresponding to the localization point of the particular boundary on which it is situated. Although this ideal is not initially realizable, it can be engineered by the "back-projection" technique of Gerig and Klein (1986). Here all peaks and other positions in parameter space to which a given edge point contributes are examined, and a new parameter space is built in which only the vote at the strongest of these peaks is retained (there is the greatest probability, but no certainty, that it belongs to the *largest* such peak). This second parameter space thus contains no extraneous clutter and weak peaks are hence found much more easily: this gives objects with highly fragmented or occluded boundaries much more chance of being detected. Overall, the method avoids many of the problems associated with setting arbitrary thresholds on peak height—in principle, no thresholds are required in this approach.

The scheme can be applied to any HT detector that throws multiple votes for each edge point. Thus, it appears to be widely applicable and is capable of improving robustness and reliability at an intrinsic expense of approximately doubling computational effort (however, set against this is the relative ease with which peaks can be located—a factor which is highly data-dependent). Note that the method is another example in which a two-stage process is used for effective recognition.

Other interesting features of the Gerig and Klein method must be omitted here for reasons of space, except to note that, rather oddly, the published scheme ignores edge orientation information as a means of reducing computation.

13.12 CONCLUDING REMARKS

The Hough transform was introduced in Chapter 11 as a line detection scheme and then used in Chapter 12 for detecting circles and ellipses. In both chapters, it appeared as a rather cunning method for aiding object detection; although it was seen to offer various advantages, particularly in its robustness in the face of noise and occlusion, there appeared to be no real significance in its rather novel voting scheme. The present chapter has shown that, far from being a trick method, the HT is much more general an approach than originally supposed: indeed, it embodies the properties of the spatial matched filter and is therefore capable of close-to-optimal sensitivity for object detection. However, this does not prevent its implementation from entailing considerable computational load, and significant effort and ingenuity have been devoted to overcoming this problem, both in general and in specific cases. The general case is tackled by the schemes discussed in Sections 13.9 and 13.10. It is important not to underestimate the value of specific solutions, both because such shapes as lines, circles, ellipses, and polygons cover a large proportion of (or approximations to) manufactured objects and because methods for coping with specific cases have a habit (as for the original HT) of becoming more general as workers see possibilities for developing the underlying techniques.

Finally, to further underline the generality of the GHT, it has also been used for optimal location of lines of known length, by emulating a spatial matched filter detector; this result has been applied to the optimal detection of polygons and of corners: for an example of the latter, see Fig. 13.12 (Davies, 1988a, 1989a). For further discussion and critique of the whole HT and GHT approach, see Chapter 27.

Although the Hough transform may appear to have a somewhat arbitrary design, this chapter has shown that it has solid roots in matched filtering, which in turn implies that votes should be gradient weighted for optimal sensitivity. The chapter also contrasts three methods for ellipse detection, showing how computational load may be estimated and minimized.

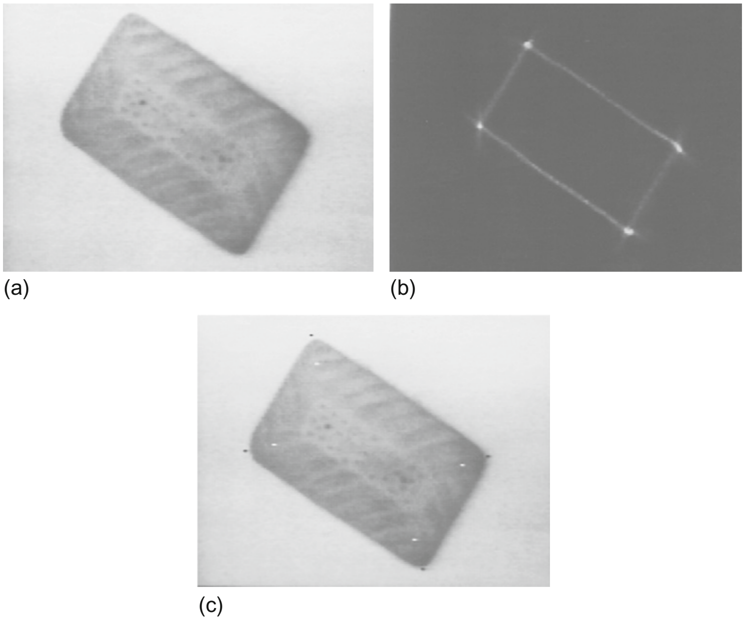

FIGURE 13.12

Example of the generalized Hough transform approach to corner detection. (a) Original image of a biscuit (128 × 128 pixels, 64 gray levels). (b) Transform with lateral displacement around 22% of the shorter side. (c) Image with transform peaks located (white dots) and idealized corner positions deduced (black dots). The lateral displacement employed here is close to the optimum for this type of object.

Source: (a,b) © IEE 1988, (c) © Unicom 1988

13.13 BIBLIOGRAPHICAL AND HISTORICAL NOTES

Although the HT was introduced as early as 1962, a number of developments—including especially those of Merlin and Farber (1975) and Kimme et al. (1975)—were required before the GHT was developed (Ballard, 1981). By that time, the HT was already known to be formally equivalent to template matching (Stockman and Agrawala, 1977) and to spatial matched filtering (Sklansky, 1978). However, the questions posed in Section 13.3 were only answered much later (Davies, 1987a), the necessary analysis being reproduced in Sections 13.4–13.7. The author's work (Davies, 1987b, 1987d, 1989b) on line detection by the GHT (not covered here) was aimed particularly at optimizing sensitivity of line detection, although deeper issues of tradeoffs between sensitivity, speed, and accuracy are also involved.

By 1985, the computational load of the HT became the critical factor preventing its more general use—particularly as it could be used for most types of arbitrary shape detection, with well-attested sensitivity and considerable robustness. Following preliminary work by Brown (1984), with the emphasis on hardware implementations of the HT, Li et al. (1985) and Li and Lavin (1986) showed the possibility of much faster peak location by using nonuniformly quantized parameter spaces. This work was developed further by Illingworth and Kittler (1987) and others (see, e.g., Illingworth and Kittler, 1988; Princen et al., 1989a, 1989b; Davies, 1992g). An important development has been the randomized Hough transform (RHT), pioneered by Xu and Oja (1993) among others: it involves casting votes until specific peaks in parameter space become evident, thereby saving unnecessary computation.

Accurate peak location remains an important aspect of the HT approach. Properly, this is the domain of robust statistics, which handles the elimination of outliers (see Appendix A). Davies (1992f) has shown a computationally efficient means of accurately locating HT peaks, and has found why peaks sometimes appear narrower than *a priori* considerations would indicate (Davies, 1992b). Kiryati and Bruckstein (1991) have tackled aliasing effects, which can arise with the HT, and which have the effect of cutting down accuracy.

Over time, the GHT approach has been broadened by geometric hashing, structural indexing, and other approaches (e.g., Lamdan and Wolfson, 1988; Gavrila and Groen, 1992; Califano and Mohan, 1994). At the same time, a probabilistic approach to the subject has been developed (Stephens, 1991), which puts it on a firmer footing. Grimson and Huttenlocher (1990) warn (possibly over-pessimistically) against the blithe use of the GHT for complex object recognition tasks, because of the false peaks that can appear in such cases. For further review of the state of the subject up to 1993, see Leavers (1993).

In various chapters of Part 2, the statement has been made that the HT[3] carries out a search leading to hypotheses that should be checked before a final decision about the presence of an object can be made. However, Princen et al. (1994) show that the performance of the HT can be improved if it is itself regarded as a hypothesis testing framework: this is in line with the concept that the HT is a model-based approach to object location. Other studies have been made about the nature of the HT. In particular, Aguado et al. (2000) consider the intimate relationship between the HT and the principle of duality in shape description: the existence of this relationship underlines the importance of the HT and provides a means for a more general definition of it. Kadyrov and Petrou (2001) have developed the trace transform, which can be regarded as a generalized form of the Radon transform—itself closely related to the Hough transform.

Other workers have used the HT for affine-invariant search: Montiel et al. (2001) made an improvement to reduce the incidence of erroneous evidence in the gathered data, whereas Kimura and Watanabe (2002) made an extension for 2-D shape detection that is less sensitive to the problems of occlusion and broken

[3] A similar statement can be made in the case of graph matching methods such as the maximal clique approach to object location (see Chapter 14).

boundaries. Kadyrov and Petrou (2002) have adapted the trace transform to cope with affine parameter estimation.

In a generalization of the work of Atherton and Kerbyson (1999) and of Davies (1987a) on gradient weighting (see Section 13.6), Anil Bharath and his colleagues have examined how to optimize the sensitivity of the HT (private communication, 2004). Their method is particularly valuable in solving the problems of early threshold setting that limit many HT techniques. Similar sentiments come out in a different way in the work of Kesidis and Papamarkos (2000), which maintains the grayscale information throughout the transform process, thereby leading to more exact representations of the original images.

Olson (1999) has shown that localization accuracy can be improved efficiently by transferring local error information into the HT and handling it rigorously. An important finding is that the HT can be subdivided into many sub-problems without decrease in performance. This finding is elaborated in a 3-D model-based vision application where it is shown to lead to reduced false positive rates (Olson, 1998). Wu et al. (2002) extend the 3-D possibilities further by using a 3-D HT to find glasses: first a set of features are located that lie on the same plane, and this is then interpreted as the glasses' rim plane. This approach allows the glasses to be separated from the face, and then they can be located in their entirety.

van Dijck and van der Heijden (2003) develop the geometric hashing method of Lamdan and Wolfson (1988) to perform 3-D correspondence matching using full 3-D hashing. This is found to have advantages in that knowledge of 3-D structure can be used to reduce the number of votes and spurious matches. Tuytelaars et al. (2003) describe how invariant-based matching and HTs can be used to identify regular repetitions in planes appearing within visual (3-D) scenes in spite of perspective skew. The overall system has the ability to reason about consistency and is able to cope with periodicities, mirror symmetries, and reflections about a point.

13.13.1 More Recent Developments

Among the most recent developments are the following. Aragon-Camarasa and Siebert (2010) considered using the GHT for clustering SIFT feature matches. However, it turned out that a *continuous* rather than discretized HT space was needed for this application. This meant that each matched point had to be stored at the full machine precision in a Hough space consisting of a list data structure. Therefore, peak location had to take the form of standard unsupervised clustering algorithms. This was an interesting case where the intended GHT could not follow the standard voting and accumulating procedure. Assheton and Hunter (2011) also deviated sharply from the standard GHT approach when performing pedestrian detection and tracking: they used a shape-based voting algorithm based on Gaussian mixture models. The algorithm was stated to be highly effective for detecting pedestrians based on the silhouette shape. Chung et al. (2010) studied the problem of information retrieval from databases. They produced a region-based solution for object retrieval using the GHT and adaptive image segmentation. A key aspect of the overall scheme was the location of affine-invariant maximally stable extremal regions (MSERs) (see Chapter 6) in the database and query images. Roy et al. (2011) applied the GHT to the detection and verification of seals (stamps) containing lettering and geometric patterns. This is a difficult problem because of the likely presence of noise, interfering text, and signatures as well as incompleteness due to the application of uneven pressure to the stamp. In practice, a seal has to be located using scale and rotation invariant features (particularly text characters); it is then detected as a GHT peak resulting from application of a spatial feature descriptor of neighboring connected component pairs, i.e., in this application, the text characters in the seal are used as basic features for seal detection instead of individual edge or feature points. Memory demands are limited by splitting the *R*-table into two different lookup tables—the character pair table and the distance table.

13.14 PROBLEMS

1. **a.** Describe the main stages in the application of the HT to locate objects in digital images. What are the particular advantages offered by the HT technique? Give reasons why they arise.
   b. It is said that the HT only leads to *hypotheses* about the presence of objects in images and that they should all be checked independently before making a final decision about the contents of any image. Comment on the accuracy of this statement.
2. Devise a GHT version of the spatial matched filter for detecting lines of known length L. Show that when used to detect an ideal line of length L, it gives a distributed response of length $2L$ that peaks at the center of the line, but when used to detect a partially occluded version of the line, it gives a response that is flat-topped over a range that includes the center of the line.
3. Show how a GHT version of the spatial matched filter can be devised to detect an equilateral triangle, leading to a star-shaped transform that peaks at the center of the triangle. How may this approach be adapted for (a) a general triangle and (b) a regular polygon having N sides?