CHAPTER

Circle and Ellipse Detection

12

As in the case of straight lines, circle features occur very widely in digital images of manufactured objects. In fact, they can conveniently be located using the Hough transform approach. This also applies for ellipses, which may appear in their own right or as oblique views of circular objects. However, locating ellipses is more complicated than locating circles because of the greater number of parameters that are involved.

Look out for:

- the basic Hough transform technique for locating circular objects in images.
- how the method can be adapted when circle radius is unknown.
- how accuracy of center location can be improved.
- how speed of processing can be increased.
- the basic diameter bisection method for ellipse detection.
- the chord−tangent method for ellipse detection.
- means of testing a shape to confirm that it is an ellipse.
- how small holes may be detected.
- how the human iris may be located.

This chapter shows how the Hough transform is able to provide a useful means for detecting objects of selected shape in digital images. Again the method is seen to rely on the accumulation of votes in a parameter space and the robustness of the technique to result from concentration on *positive* evidence for the objects.

Computer and Machine Vision.

12.1 INTRODUCTION

Location of round objects is important in many areas of image analysis but it is especially important in industrial applications such as automatic inspection and assembly. In the food industry alone, a very sizable number of products are round—biscuits, cakes, pizzas, pies, jellies, oranges, and so on (Davies, 1984c). In the automotive industry, many circular components are used—washers, wheels, pistons, heads of bolts, and so on, while round holes of various sizes appear in such items as casings and cylinder blocks. In addition, buttons and many other everyday objects are round. Of course, when round objects are viewed obliquely, they appear elliptical; furthermore, certain other objects are *actually* elliptical. This makes it clear that we need algorithms that are capable of finding both circles and ellipses. Finally, objects can frequently be located by their holes, so finding round holes or features is part of the larger problem: this chapter addresses various aspects of this problem.

An important facet of this work is how well object location algorithms cope in the presence of artifacts such as shadows and noise. In particular, the paradigm represented by Table 12.1 has been shown in Chapter 10 to be insufficiently robust to cope in such situations. This chapter shows that the Hough transform (HT) technique is able to overcome many of these problems. Indeed, it is found to be particularly good at dealing with all sorts of difficulties, including quite severe occlusions. It achieves this not by adding robustness but by having robustness built in as an integral part of the technique.

The application of the HT to circle detection is one of the most straightforward uses of the technique. However, there are several enhancements and adaptations that can be applied in order to improve accuracy and speed of operation, and in addition to make the method work efficiently when detecting circles with a range of sizes. These modifications are studied after covering the basic HT technique. Versions of the Hough transform that can perform ellipse detection are then considered. Finally, after a short section on an important application—that of human iris location—the topic of hole detection is considered.

Table 12.1 Procedure for Finding Objects Using (r, θ) Boundary Graphs

1. Locate edges within the image
2. Link broken edges
3. Thin thick edges
4. Track around object outlines
5. Generate a set of (r, θ) plots
6. Match (r, θ) plots to standard templates

This procedure is not sufficiently robust with many types of real data, e.g., in the presence of noise, distortions in product shape, and so on: in fact, it is quite common to find the tracking procedure veering off and tracking around shadows or other artifacts.

12.2 HOUGH-BASED SCHEMES FOR CIRCULAR OBJECT DETECTION

In the original HT method for finding circles (Duda and Hart, 1972), the intensity gradient is first estimated at all locations in the image and then thresholded to give the positions of significant edges. Then the positions of all possible center locations—namely all points a distance R away from every edge pixel—are accumulated in parameter space, R being the anticipated circle radius. Parameter space can be a general storage area but when looking for circles it is convenient to make it congruent to image space: in that case, possible circle centers are accumulated in a new plane of image space. Finally, parameter space is searched for peaks that correspond to the centers of circular objects. Since edges have nonzero width and noise will always interfere with the process of peak location, accurate center location requires the use of suitable averaging procedures (Davies, 1984c; Brown, 1984).

This approach clearly requires a very large number of points to be accumulated in parameter space and so a revised form of the method has now become standard: in this approach, locally available edge orientation information at each edge pixel is used to enable the exact positions of circle centers to be estimated (Kimme et al., 1975). This is achieved by moving a distance R along the edge normal at each edge location. Thus, the number of points accumulated is equal to the number of edge pixels in the image:[1] this represents a significant saving in computational load. For this to be possible, the edge detection operator that is employed must be highly accurate. Fortunately, the Sobel operator is able to estimate edge orientation to 1° and is very simple to apply (Chapter 5). Thus, the revised form of the transform is viable in practice.

As was seen in Chapter 5, once the Sobel convolution masks have been applied, the local components of intensity gradient g_x and g_y are available, and the magnitude and orientation of the local intensity gradient vector can be computed using the formulae:

$$g = \left(g_x^2 + g_y^2\right)^{1/2} \tag{12.1}$$

and

$$\theta = \arctan\left(\frac{g_y}{g_x}\right) \tag{12.2}$$

However, use of the arctan operation is not necessary when estimating center location coordinates (x_c, y_c) since the trigonometric functions can be made to cancel out:

$$x_c = x - R\left(\frac{g_x}{g}\right) \tag{12.3}$$

[1] We assume that objects arc known to be *either* lighter *or* darker than the background, so that it is only necessary to move along the edge normal in one direction.

$$y_{\mathrm{c}} = y - R\left(\frac{g_y}{g}\right) \tag{12.4}$$

the values of cos θ and sin θ being given by:

$$\cos\theta = \frac{g_x}{g} \tag{12.5}$$

$$\sin\theta = \frac{g_y}{g} \tag{12.6}$$

In addition, the usual edge thinning and edge linking operations—which normally require considerable amounts of processing—can be avoided if a little extra smoothing of the cluster of candidate center points is performed (Davies, 1984c) (Table 12.2). Thus, this Hough-based approach can be a very efficient one for locating the centers of circular objects, virtually all the superfluous operations having been eliminated, leaving only edge detection, location of candidate center points, and center point averaging to be carried out. In addition, the method is highly robust, so if part of the boundary of an object is obscured or distorted, the object center is still located accurately. In fact, the results are often quite impressive (see, e.g., Figs. 12.1 and 12.2). The reason for this useful property is clear from Fig. 12.3.

The efficiency of the above technique means that it takes slightly less time to perform the actual HT part of the calculation than to evaluate and threshold the intensity gradient over the whole image. Part of the reason for this is that the edge detector operates within a 3 × 3 neighborhood and necessitates some 12 pixel accesses, four multiplications, eight additions, two subtractions, and an operation for the evaluation of the square root of sum of squares (Eq. (12.1)). As seen in Chapter 5, the latter type of operation is commonly approximated by taking a sum or maximum of absolute magnitudes of the component intensity gradients in order to estimate the magnitude of the local intensity gradient vector.

Table 12.2 A Hough-Based Procedure for Locating Circular Objects

1. Locate edges within the image
2. Link broken edges
3. Thin thick edges
4. For every edge pixel, find a candidate center point
5. Locate all clusters of candidate centers
6. Average each cluster to find accurate center locations

This procedure is particularly robust. It is largely unaffected by shadows, image noise, shape distortions, and product defects. Note that stages 1–3 of the procedure are identical to stages 1–3 in Table 12.1. However, in the Hough-based method, computation can be saved, and accuracy actually increased, by omitting stages 2 and 3.

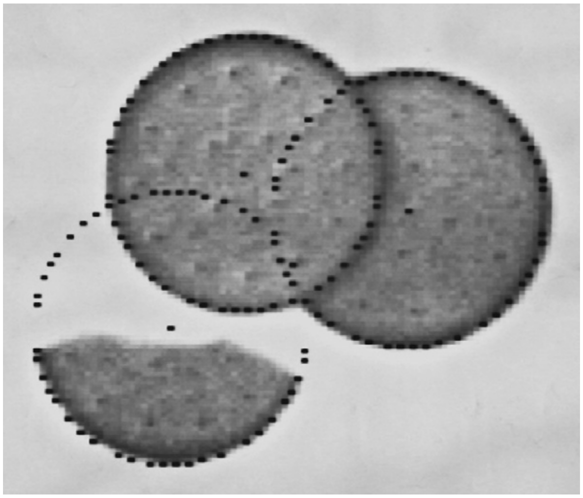


FIGURE 12.1

Location of broken and overlapping biscuits, showing the robustness of the center location technique. Accuracy is indicated by the black dots, which are each within 1/2 pixel of the radial distance from the center.

Source: © *IFS Publications Ltd 1984*

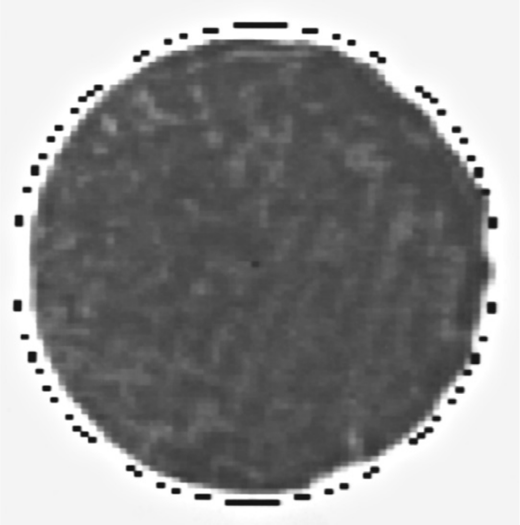


FIGURE 12.2

Location of a biscuit with a distortion, showing a chocolate-coated biscuit with excess chocolate on one edge. Note that the computed center has not been “pulled” sideways by the protuberances. For clarity, the black dots are marked 2 pixels outside the normal radial distance.

Source: © *IFS Publications Ltd 1984*

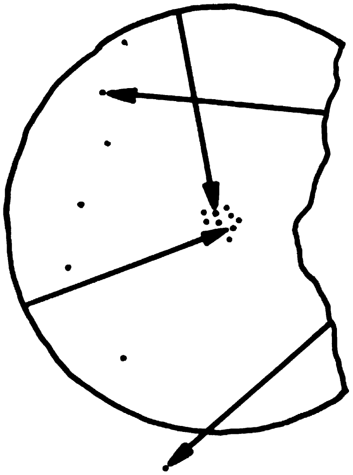


FIGURE 12.3

Robustness of the Hough transform when locating the center of a circular object. The circular part of the boundary gives candidate center points that focus on the true center, whereas the irregular broken boundary gives candidate center points at random positions.

Source: © *Unicom 1988*

However, this type of shortcut is not advisable in the present context, since an accurate value for the magnitude of this vector is required in order to compute the position of the corresponding candidate center location with sufficient precision.

Overall, the dictates of accuracy imply that candidate center location requires significant computation. However, substantial increases in speed are still possible by software means alone, as will be seen later in the chapter. Meanwhile, we consider the problems that arise when images contain circles of many different radii, or for one reason or another radii are not known in advance.

12.3 THE PROBLEM OF UNKNOWN CIRCLE RADIUS

There are a number of situations where circle radius is initially unknown. One such situation is where a number of circles of various sizes are being sought—as in the case of coins, or different types of washer. Another is where the circle size is variable—as for food products such as biscuits—so that some tolerance must be built into the system. In general, all circular objects have to be found and their radii measured. In such cases, the standard technique is to accumulate candidate center points simultaneously in a number of parameter planes in a suitably augmented parameter space, each plane corresponding to one possible radius value. The centers of the peaks detected in parameter space give not only the location of each circle in two dimensions but also its radius. Although this scheme is entirely viable in principle, there are several problems in practice:

1. Many more points have to be accumulated in parameter space.
2. Parameter space requires much more storage.
3. Significantly greater computational effort is involved in searching parameter space for peaks.

To some extent this is to be expected, since the augmented scheme enables more objects to be detected directly in the original image.

It is shown below that the last two problems may largely be eliminated. This is achieved by using just one parameter plane to store all the information for locating circles of different radii, i.e., accumulating not just one point per edge pixel but a whole line of points along the direction of the edge normal in this one plane. In practice, the line need not be extended indefinitely in either direction but only over the restricted range of radii over which circular objects or holes might be expected.

Even with this restriction, a large number of points are being accumulated in a single parameter plane, and it might be thought initially that this would lead to such a proliferation of points that almost any "blob" shape would lead to a peak in parameter space, which might be interpreted as a circle center. However, this is not so and significant peaks normally result only from genuine circles and some closely related shapes.

To understand the situation, consider how a sizeable peak can arise at a particular position in parameter space. This can happen only when a large number of radial vectors from this position meet the boundary of the object normally. In the absence of discontinuities in the boundary, a contiguous set of boundary points can only be normal to radius vectors if they lie on the arc of a circle (indeed, a circle could be *defined* as a locus of points that are normal to the radius vector and form a thin connected line). If a limited number of discontinuities are permitted in the boundary, it may be deduced that shapes like that of a fan[2] will also be detected using this scheme. Since it is in any case useful to have a scheme that can detect such shapes, the main problem is that there will be some ambiguity in interpretation—i.e., does peak P in parameter space arise from a circle or a fan? In practice, it is quite straightforward to distinguish between such shapes with relatively little additional computation, the really important problem being to cut down the amount of computation needed to key into the initially unstructured image data. Indeed, it is often a good strategy to prescreen the image to eliminate most of it from further detailed consideration and then to analyze the remaining data with tools having much greater discrimination: this two-stage template matching procedure frequently leads to significant savings in computation (VanderBrug and Rosenfeld, 1977; Davies, 1988g).

[2] A four-blade fan shape bounded by two concentric circles with eight equally spaced radial lines joining them.

A more significant problem arises because of errors in the measurement of local edge orientation. As stated earlier, edge detection operators such as the Sobel introduce an inherent inaccuracy of about 1°. Image noise typically adds a further 1° error, and for objects of radius 60 pixels, the result is an overall uncertainty of about 2 pixels in the estimation of center location. This makes the scheme slightly less specific in its capability for detecting objects.

Overall, the scheme is likely to accept the following object shapes during its prescreening stage:

1. Circles of various sizes
2. Shapes such as fans, which contain arcs of circles
3. Partly occluded or broken versions of these shapes

Sometimes, objects of type 2 will be known *not* to be present. Alternatively, they may readily be identified, e.g., with the aid of corner detectors. Hence, problems of ambiguity may be nonexistent or easy to eliminate in practice. As the situation is highly application-dependent, we will eschew further discussion of this point here.

Next, note that the information on radial distance has been lost by accumulating all votes in a single parameter plane. Hence, a further stage of analysis is needed to measure object radius. This extra stage of analysis normally involves negligible additional computation, because the search space has been narrowed down so severely by the initial circle location procedure. The radial histogram technique (Chapter 20) can be used to measure the radius: in addition, it can be used to perform a final object inspection function.

12.3.1 Some Practical Results

The method described above works much as expected, the main problems arising with small circular objects (of radii less than about 20 pixels) at low resolution (Davies, 1988b). Essentially, the problems are those of lack of discrimination in the precise shapes of small objects (see Figs. 12.4 and 12.5), as anticipated above. As suggested earlier, this can frequently be turned to advantage in that the method becomes a circular feature detector for small radii (see Fig. 12.5, where a wing nut is located).

As required, objects are detected reliably even when they are partly occluded. However, occlusions can result in the centers of small objects being "pulled" laterally (Fig. 12.4). More generally, it is clear from Figs. 12.4 and 12.5 that high accuracy of center location cannot be expected when a single parameter plane is used to detect objects over a large range of sizes: hence, it is best to cut down the voting range as far as possible.

Overall, there is a tradeoff between speed and accuracy with this approach. However, the results confirm that it is possible to locate objects of various radii within a significantly conflated parameter space, thereby making substantial

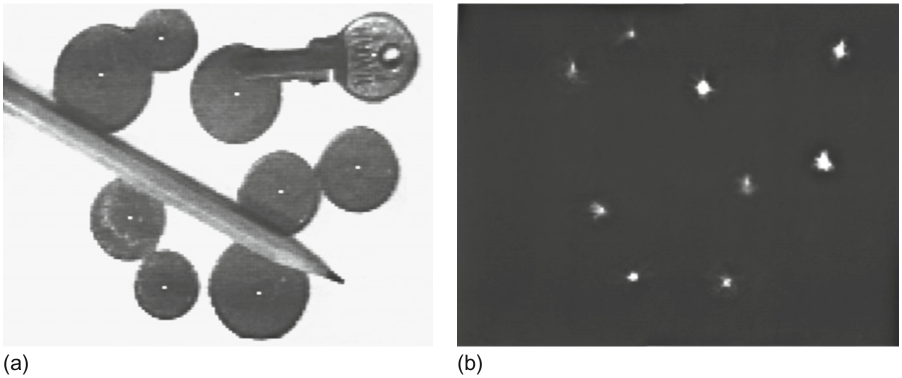


FIGURE 12.4

(a) Simultaneous location of coins and a key with the modified Hough scheme: the various radii range from 10 to 17 pixels and (b) transform used to compute the centers indicated in (a). Detection efficiency is unimpaired by partial occlusions, shape distortions, and glints. However, displacements of some centers are apparent; in particular, one coin (top left) has only two arcs showing the fact that one of these is distorted, giving a lower curvature, leads to a displacement of the computed center. The shape distortions are due to rather uneven illumination.

savings in storage and computation—even though the total number of votes that have to be accumulated in parameter space is not itself reduced.

12.4 THE PROBLEM OF ACCURATE CENTER LOCATION

Section 12.3 analyzed the problem of how to cope efficiently with images containing circles of unknown or variable size. This section examines how centers of circles may be located with high (preferably subpixel) accuracy. This problem is relevant since the alternative of increased resolution would entail the processing of many more pixels. Hence, it will be of advantage if high accuracy can be attained with images of low or moderate resolution.

There are a number of causes of inaccuracy in locating the centers of circles using the HT:

1. Natural edge width may add errors to any estimates of the location of the center.
2. Image noise may cause the radial position of the edge to become modified.
3. Image noise may cause variation in the estimates of local edge orientation.
4. The object itself may be distorted, in which case the most accurate and robust estimate of circle center is what is required (i.e., the algorithm should be able to ignore the distortion and take a useful average from the remainder of the boundary).

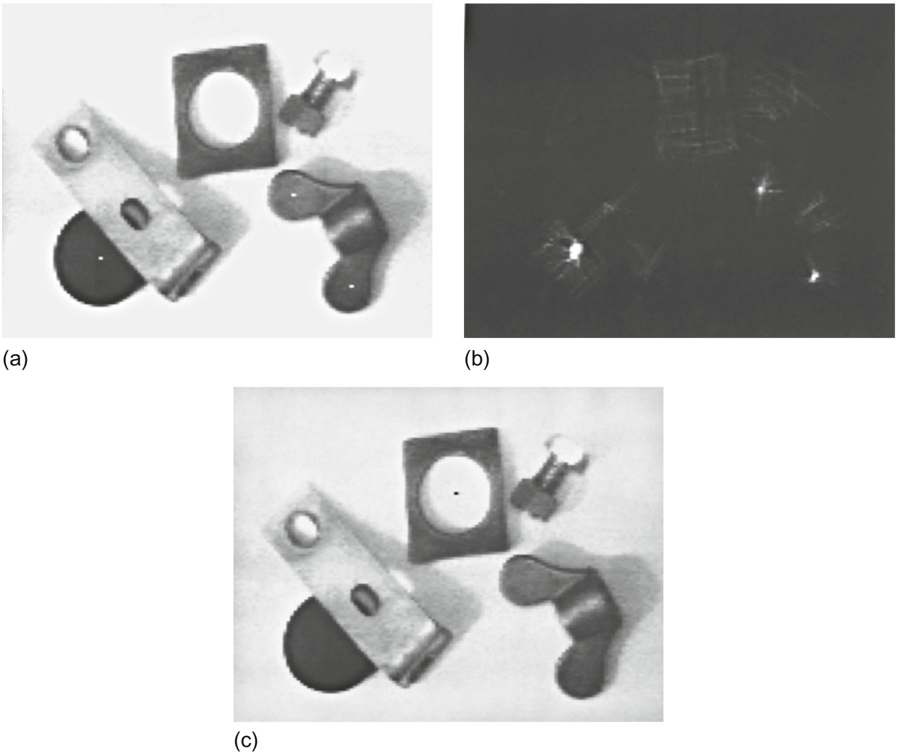


FIGURE 12.5

(a) Accurate simultaneous detection of a lens cap and a wing nut when radii are assumed to range from 4 to 17 pixels; (b) response in parameter space that arises with such a range of radii: note the overlap of the transforms from the lens cap and the bracket; (c) hole detection in the image of (a) when radii are assumed to fall in the range −26 to −9 pixels (negative radii are used since holes are taken to be objects of negative contrast): clearly, in *this* image a smaller range of negative radii could have been employed.

5. The object may appear distorted because of inadequacies in the method of illumination.
6. The edge orientation operator will have a significant inherent inaccuracy of its own, which contributes to inaccuracy in the estimation of center location.

Evidently, it is necessary to minimize the effects of all these potential sources of error. In applying the HT, it is usual to bear in mind that all possible center locations that could have given rise to the currently observed edge pixel should be accumulated in parameter space. Therefore, we should accumulate in parameter space not a single candidate center point corresponding to the given edge pixel, but a point spread function (PSF), which may be approximated by a Gaussian error function: this will generally have different radial and transverse

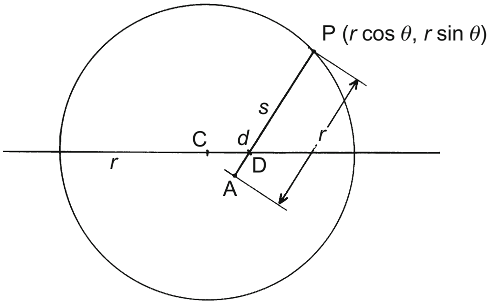


FIGURE 12.6

Arrangement for obtaining improved center approximation.

standard deviations. The radial standard deviation will commonly be 1 pixel—corresponding to the expected width of the edge—whereas the transverse standard deviation will frequently be at least 2 pixels, and for objects of moderate (60 pixels) radius the intrinsic orientation accuracy of the edge detection operator contributes at least 1 pixel to this standard deviation.

When the center need only be estimated to within 1 pixel, these considerations hardly matter. However, when it is desired to locate center coordinates to within 0.1 pixel, each PSF will have to contain 100 points, all of which will have to be accumulated in a parameter space of much increased spatial resolution, or its equivalent,[3] if the required accuracy is to be attained. However, the following section outlines a technique that can cut down the amount of computation from this sort of level, without significant loss of accuracy.

12.4.1 A Solution Requiring Minimal Computation

A potential key to increasing accuracy of center location arises from the observation that most of the inaccuracy in calculating the position of the center is due to transverse rather than radial errors. Hence, it is reasonable to concentrate on eliminating transverse errors.

This immediately leads to the following strategy: find a point D in the region of the center and use it to obtain a better approximation A to the center by moving from the current edge pixel P a distance equal to the expected radius r in the direction of D (Fig. 12.6). Then repeat the process 1 edge pixel at a time until all edge pixels have been taken into account. Although intuition indicates that this strategy will lead to a good final approximation to the center, the edge points need to be taken in random order to prevent the bias of the initial center approximation from

[3] For example, some abstract list structure might be employed, which effectively builds up to high resolution only where needed (see also the adaptive HT scheme described in Chapter 13).

permanently affecting the final result.[4] In addition, some averaging of the intermediate results is needed to minimize the effects of boundary noise, and edge points that give extreme predictions for the center approximation (e.g., deviations of more than ∼2 pixels) have to be discounted.

Overall, there are two main stages of calculation in the whole algorithm:

1. Find the position of the center to within ∼1 pixel by the usual Hough technique.
2. Use the iterative procedure to obtain the final result, with an accuracy of 0.1 pixels.

Tests of this algorithm on accurately made circular objects led to accuracies of center location in the region of 0.1 pixels (Davies, 1988e). It proved difficult to increase the accuracy further because of the problem of setting up lighting systems of sufficient uniformity, i.e., the limit was set by practicalities of illumination rather than by the algorithm itself.

Finally, a by-product of the approach is that radius r can be obtained with exceptionally high accuracy—generally within 0.05 pixels (Davies, 1988e).

12.5 OVERCOMING THE SPEED PROBLEM

Section 12.4 studied how the accuracy of the HT circle detection scheme could be improved substantially, with modest additional computational cost. This section examines how circle detection may be carried out with significant improvement in speed. To achieve this, two methods are tried: (1) sampling the image data and (2) using a simpler edge detector. The most appropriate strategy for (1) appears to be to look only at every nth line in the image, whereas that for (2) involves using a small two-element neighborhood while searching for edges (Davies, 1987f). Although this approach will lose the capability for estimating edge orientation, it will still permit horizontal and vertical chords of a circle to be bisected, thereby leading to values for the center coordinates x_c, y_c. It also involves much less computation, the multiplications and square root calculations, and most of the divisions being eliminated or replaced by two-element differencing operations, thereby giving a further gain in speed.

12.5.1 More Detailed Estimates of Speed

To help understand the situation, this section estimates the gain in speed that should result by applying the strategy described above. First, the amount of computation involved in the original Hough-based approach is modeled by:

$$T_0 = N^2 s + S t_0 \tag{12.7}$$

[4] This problem arises because the method is intrinsically sequential rather than parallel, and it is necessary to compensate for this.

where T_0 is the total time taken to run the algorithm on an $N \times N$ pixel image; s is the time taken per pixel to compute the Sobel operator and to threshold the intensity gradient g; S is the number of edge pixels that are detected; and t_0 is the time taken to compute the position of a candidate center point.

Next, the amount of computation in the basic chord bisection approach is modeled as:

$$T = 2(N^2 q + Qt) \tag{12.8}$$

where T is the total time taken to run the algorithm; q is the time taken per pixel to compute a two-element x or y edge detection operator and to perform thresholding; Q is the number of edge pixels that are detected in one or other scan direction; and t is the time taken to compute the position of a candidate center point coordinate (either x_c or y_c). In Eq. (12.8), the factor 2 results because scanning occurs in both horizontal and vertical directions. If the image data are now sampled by examining only a proportion α of all possible horizontal and vertical scan lines, the overall gain in speed from using the chord bisection scheme should be:

$$G = \frac{N^2 s + St_0}{2\alpha(N^2 q + Qt)} \tag{12.9}$$

Typical values of relevant parameters for (say) a biscuit of radius 32 pixels in a 128 × 128 pixel image are listed below:

$N^2 = 16{,}384$	$t_0/s \approx 1$
$S \approx Q \approx 200$	$s/q \approx 6$
$\alpha \approx 1/3$	$t_0/t \approx 5$

Hence

$$G \approx \frac{s}{2\alpha q} \approx 9 \tag{12.10}$$

Broadly, this corresponds to a gain ~3 from sampling and a further gain ~3 from applying a much simplified edge detector. However, if the sampling factor l/α could be increased further, greater gains in speed could be obtained—in principle without limit, but in practice the situation is governed by how robust the algorithm really is.

12.5.2 Robustness

Robustness can be considered relative to two factors. The first is the amount of noise in the image and the second is the amount of signal distortion that can be tolerated. Fortunately, both the original HT and the chord bisection approach lead to peak finding situations, and if there is any distortion of the object shape, then points are thrown into relatively random locations in parameter space and consequently do not have a significant direct impact on the accuracy of peak location.

However, they do have an indirect impact in that the signal-to-noise ratio is reduced, so that accuracy is impaired. In fact, if a fraction β of the original signal is removed, leaving a fraction $\gamma = 1 - \beta$, either due to such distortions or occlusions or else by the deliberate sampling procedures already outlined, then the number of independent measurements of the center location drops to a fraction γ of the optimum. This means that the accuracy of estimation of the center location drops to a fraction $\sqrt{\gamma}$ of the optimum. Since noise affects the optimum accuracy directly, we have shown the result of both major factors governing robustness.

What is important here is that the effect of sampling is substantially the same as that of signal distortion, so that the more distortion that must be tolerated, the higher the value α has to have. This principle applies both to the original Hough approach and to the chord bisection algorithm. However, the latter does its peak finding in a different way—*via* 1-D rather than 2-D averaging processes. As a result, it turns out to be somewhat more robust than the standard HT in its capability for accepting a high degree of sampling.

This gain in capability for accepting sampling carries with it a set of related disadvantages: highly textured objects may not easily be located by the method; high noise levels may confuse it via the production of too many false edges; and (since it operates by specific x and y scanning mechanisms) there must be a sufficiently small amount of occlusion and other gross distortion that a significant number of scans (both horizontally and vertically) pass through the object. Ultimately, this means that the method will not tolerate more than about one-quarter of the circumference of the object being absent.

12.5.3 Practical Results

Tests (Davies, 1987f) with the image in Fig. 12.7 show that gains in speed of more than 25 can be obtained, with values of α down to less than 0.1 (i.e., every 10th horizontal and vertical line scanned). The results for broken circular products (Figs. 12.8 and 12.9) are self-explanatory; they indicate the limits to which the method can be taken. An outline of the complete algorithm is given in Table 12.3 (note the relatively straightforward problem of disambiguating the results if there happen to be several peaks).

Figure 12.10 shows the effect of adjusting the threshold in the two-element edge detector. The result of setting it too low is seen in Fig. 12.10(a). Here the surface texture of the object has triggered the edge detector, and the chord midpoints give rise to a series of false estimates of the center coordinates. Figure 12.10(b) shows the result of setting the threshold at too high a level, so that the number of estimates of center coordinates is reduced and sensitivity suffers. Although the images in Fig. 12.7 were obtained with the threshold adjusted intuitively, a more rigorous approach can be taken by optimizing a suitable criterion function. There are two obvious functions: (1) the number of accurate center predictions n and (2) the speed–sensitivity product. The latter can be written in the form $\sqrt{n}/T$, where T is the execution time. The two methods of optimization make little difference in the

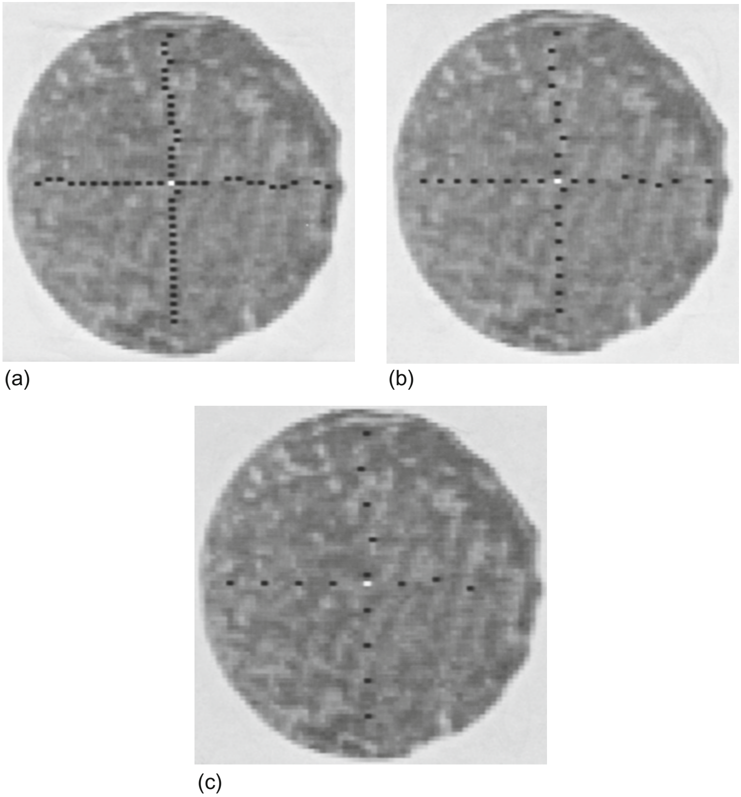


FIGURE 12.7

Successful object location using the chord bisection algorithm for the same initial image, using successive step sizes of 2, 4, and 8 pixels. The black dots show the positions of the horizontal and vertical chord bisectors, and the white dot shows the position found for the center.

example shown in Fig. 12.10. However, had there been a strong regular texture on the object, the situation would have been rather different.

12.5.4 Summary

The center location procedure described above is more than an order of magnitude faster than the standard Hough-based approach and often as much as 25 times faster. This could be quite important in exacting real-time applications. Robustness is so good that the method tolerates at least one-quarter of the circumference of an object being absent, making it adequate for many real applications. In addition, it is entirely clear what types of image data would be likely to confuse the algorithm.

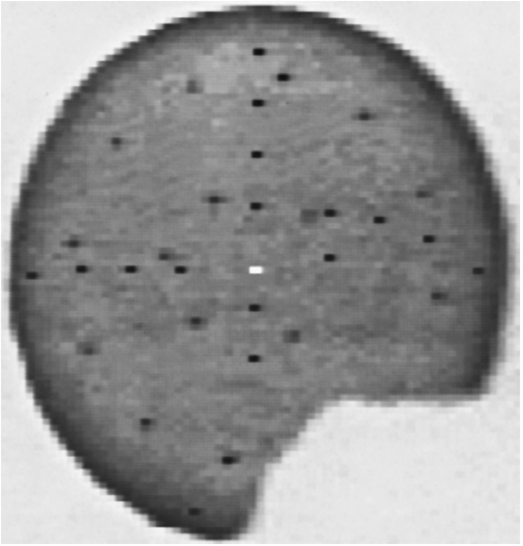


FIGURE 12.8

Successful location of a broken object using the chord bisection algorithm: only about one-quarter of the ideal boundary is missing.

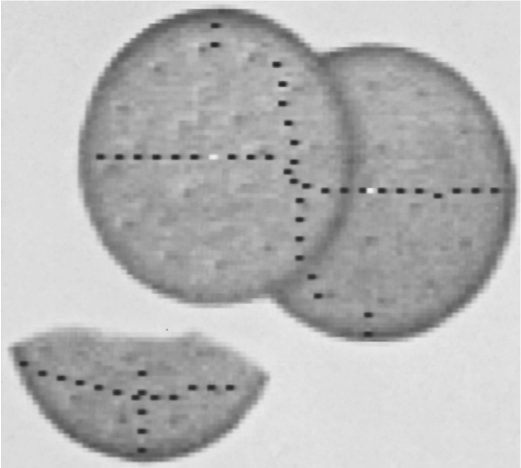


FIGURE 12.9

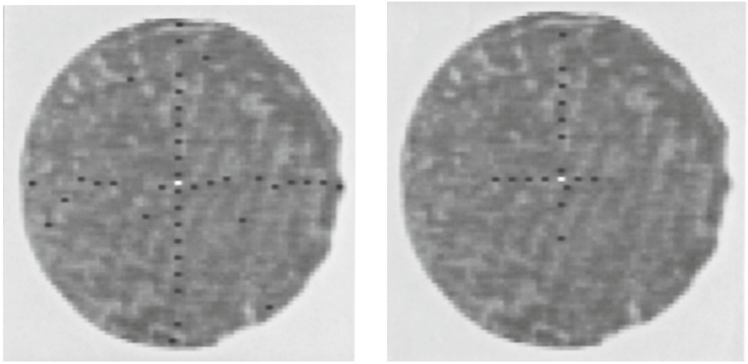
A test on overlapping and broken biscuits: the overlapping objects are successfully located, albeit with some difficulty, but there is no chance of finding the center of the broken biscuit since over one-half of the ideal boundary is missing.

Table 12.3 Outline of the Fast Center-Finding Algorithm

```
y = 0;
do {
    scan horizontal line y looking for start and end of each object;
    calculate midpoints of horizontal object segments;
    accumulate midpoints in 1-D parameter space (x space);
    // note that the same space, x space, is used for all lines y
    y = y + d;
} until y > ymax;

x = 0;
do {
    scan vertical line x looking for start and end of each object;
    calculate midpoints of vertical object segments;
    accumulate midpoints in 1-D parameter space (y space);
    // note that the same space, y space, is used for all lines x
    x = x + d;
} until x > xmax;

find peaks in x space;
find peaks in y space;
test all possible object centres arising from these peaks;
// the last step is necessary only if ∃ > 1 peak in each space
// d is the horizontal and vertical step-size ( = 1/α)
```



(a)

(b)

FIGURE 12.10

Effect of misadjustment of the gradient threshold: (a) effect of setting the threshold too low, so that surface texture confuses the algorithm and (b) loss of sensitivity on setting the threshold too high.

Not only is robustness built into the algorithm in such a way as to emulate the standard Hough-based approach, but it is possible to interpret the method as being a Hough-based approach, since the center coordinates are each accumulated in their own 1-D "parameter spaces." These considerations give further insight into the robustness of the standard Hough technique.

12.6 ELLIPSE DETECTION

The problem of detecting ellipses may seem only marginally more complex than that of detecting circles—as eccentricity is only a single parameter. However, eccentricity destroys the symmetry of the circle, so the direction of the major axis also has to be defined. As a result, five parameters are required to describe an ellipse, and ellipse detection has to take account of this, either explicitly or implicitly. In spite of this, one method for ellipse detection is especially simple and straightforward to implement, i.e., the diameter bisection method, which is described next.

12.6.1 The Diameter Bisection Method

The diameter bisection method of Tsuji and Matsumoto (1978) is very simple in concept. First, a list is compiled of all the edge points in the image. Then, the list is sorted to find those that are antiparallel, so that they could lie at opposite ends of ellipse diameters; next, the positions of the center points of the connecting lines for all such pairs are taken as voting positions in parameter space (Fig. 12.11). As for circle location, the parameter space that is used for this purpose is congruent to image space. Finally, the positions of significant peaks in parameter space are located to identify possible ellipse centers.

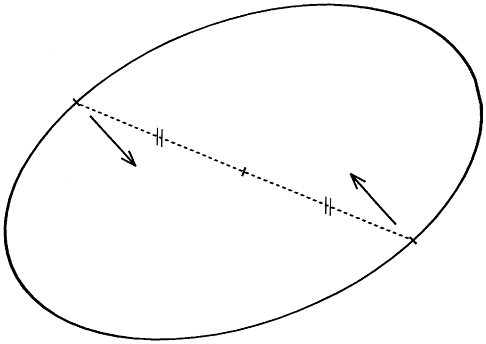


FIGURE 12.11

Principle of the diameter bisection method. A pair of points is located for which the edge orientations are antiparallel. If such a pair of points lies on an ellipse, the midpoint of the line joining the points will be situated at the center of the ellipse.

Naturally, in an image containing many ellipses and other shapes, there will be very many pairs of antiparallel edge points and for most of these the center points of the connecting lines will lead to nonuseful votes in parameter space. Clearly, such clutter leads to wasted computation. However, it is a principle of the HT that votes must be accumulated in parameter space at all points that *could in principle* lead to correct object center location: it is left to the peak finder to find the voting positions that are most likely to correspond to object centers.

Not only does clutter lead to wasted computation, but the method itself also is computationally expensive. This is because it examines all *pairs* of edge points, and there are many more such pairs than there are edge points (m edge points lead to ${}^{m}C_2 \approx m^2/2$ pairs of edge points). Indeed, since there are likely to be at least 1000 edge points in a typical image, the computational problems can be formidable.

Interestingly, the basic method is not particularly discriminating about ellipses. It picks out many symmetrical shapes—any indeed that possess 180° rotation symmetry, including rectangles, ellipses, circles, or superellipses (these have equations of the form $x^s/a^s + y^s/b^s = 1$, ellipses being a special case). In addition, the basic scheme sometimes gives rise to a number of false identifications even in an image in which only ellipses are present (Fig. 12.12). However, Tsuji and Matsumoto (1978) also proposed a technique by which true ellipses can be distinguished. The basis of the technique is the property of an ellipse that the lengths of perpendicular semidiameters OP, OQ obey the relation:

$$\frac{1}{\mathrm{OP}^2} + \frac{1}{\mathrm{OQ}^2} = \frac{1}{R^2} = \text{constant} \tag{12.11}$$

To proceed, the set of edge points that contribute to a given peak in parameter space is used to construct a histogram of R values (the latter being obtained from

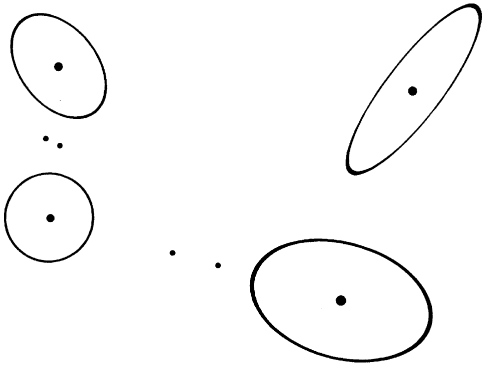


FIGURE 12.12

Result of using the basic diameter bisection method. The larger dots show true ellipse centers found by the method, whereas the smaller dots show positions at which false alarms commonly occur. Such false alarms are eliminated by applying the test described in the text.

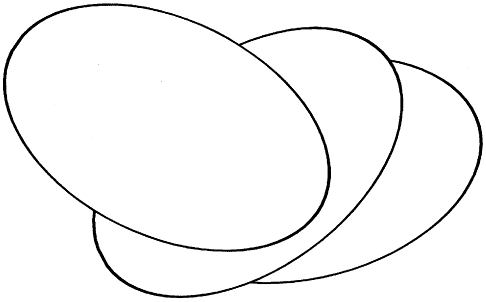


FIGURE 12.13

Limitations of the diameter bisection method: of the three ellipses shown, only the lowest one cannot be located by the diameter bisection method.

Source: © *Unicom 1988*

Eq. (12.11)). If a significant peak is found in this histogram, then there is clear evidence of an ellipse at the specified location in the image. If two or more such peaks are found, then there is evidence of a corresponding number of concentric ellipses in the image. If, however, no such peaks are found, then a rectangle, superellipse, or other symmetrical shape may be present and each of these would need its own identifying test.

The method obviously relies on there being an appreciable number of pairs of edge points on an ellipse lying at opposite ends of diameters: hence, there are strict limits on the amount of the boundary that must be visible (Fig. 12.13). Finally, it should not go unnoticed that the method wastes the signal available from unmatched edge points. These considerations have led to a search for further methods of ellipse detection.

12.6.2 The Chord–Tangent Method

The chord–tangent method was devised by Yuen et al. (1988) and makes use of another simple geometric property of the ellipse. Again pairs of edge points are taken in turn, and for each point of the pair, tangents to the ellipse are constructed and found to cross at T, the midpoint of the connecting line is found at M, and then the equation of line TM is calculated and all points that lie on the portion MK of this line are accumulated in parameter space (Fig. 12.14) (clearly, T and the center of the ellipse lie on the opposite sides of M). Finally, peak location proceeds as before.

The proof that this method is correct is trivial. Symmetry ensures that the method works for circles, and projective properties then ensure that it also works for ellipses: under projection, straight lines project into straight lines, midpoints into midpoints, tangents into tangents, and circles into ellipses; in addition, it is

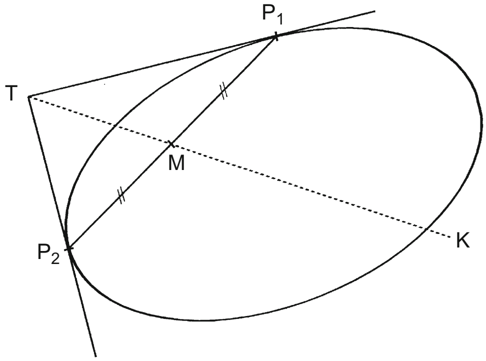


FIGURE 12.14

Principle of the chord–tangent method. The tangents at P_1 and P_2 meet at T and the midpoint of P_1P_2 is M. The center C of the ellipse lies on the line TM produced. Note that M lies between C and T. Hence, the transform for points P_1 and P_2 need only include the portion MK of this line.

always possible to find a viewpoint such that a circle can be projected into a given ellipse.

Unfortunately, this method suffers from significantly increased computation, since so many points have to be accumulated in parameter space. This is obviously the price to be paid for greater applicability. However, computation can be minimized in at least three ways: (1) cutting down the lengths of the lines of votes accumulated in parameter space by taking account of the expected sizes and spacings of ellipses; (2) not pairing edge points initially if they are too close together or too far apart; and (3) eliminating edge points once they have been identified as belonging to a particular ellipse.

12.6.3 Finding the Remaining Ellipse Parameters

Although the methods described above are designed to locate the center coordinates of ellipses, a more formal approach is required to determine other ellipse parameters. Accordingly, we write the equation of an ellipse in the form:

$$Ax^2 + 2Hxy + By^2 + 2Gx + 2Fy + C = 0 \qquad (12.12)$$

an ellipse being distinguished from a hyperbola by the additional condition:

$$AB > H^2 \qquad (12.13)$$

This condition guarantees that A can never be zero and that the ellipse equation may without loss of generality be rewritten with $A = 1$. This leaves five parameters, which can be related to the position of the ellipse, its orientation, and its size and shape (or eccentricity).

Having located the center of the ellipse, we may select a new origin of coordinates at its center (x_c, y_c); the equation then takes the form:

$$x'^2 + 2Hx'y' + By'^2 + C' = 0 \tag{12.14}$$

where

$$x' = x - x_c; \quad y' = y - y_c \tag{12.15}$$

It now remains to fit to Eq. (12.14) the edge points that gave evidence for the ellipse center under consideration. The problem will normally be vastly overdetermined. Hence, an obvious approach is the method of least squares. Unfortunately, this technique tends to be very sensitive to outlier points and is therefore liable to be inaccurate. An alternative is to employ some form of Hough transformation. Here we follow Tsukune and Goto (1983) by differentiating Eq. (12.14):

$$x' + \frac{By'}{dx'} + H\left(y' + \frac{x'\mathrm{d}y'}{\mathrm{d}x'}\right) = 0 \tag{12.16}$$

Then $\mathrm{d}y'/\mathrm{d}x'$ can be determined from the local edge orientation at (x', y') and a set of points accumulated in the new (H, B) parameter space. When a peak is eventually located in (H, B) space, the relevant data (a subset of a subset of the original set of edge points) can be used with Eq. (12.14) to obtain a histogram of C' values, from which the final parameter for the ellipse can be obtained.

The following formulae are needed to determine the orientation θ and semi-axes a and b of an ellipse in terms of H, B, and C':

$$\theta = \frac{1}{2}\arctan\left(\frac{2H}{1-B}\right) \tag{12.17}$$

$$a^2 = \frac{-2C'}{(B+1) - [(B-1)^2 + 4H^2]^{1/2}} \tag{12.18}$$

$$b^2 = \frac{-2C'}{(B+1) + [(B-1)^2 + 4H^2]^{1/2}} \tag{12.19}$$

Mathematically, θ is the angle of rotation that diagonalizes the second-order terms in Eq. (12.14); having performed this diagonalization, the ellipse is then essentially in the standard form $\tilde{x}^2/a^2 + \tilde{y}^2/b^2 = 1$, so a and b are determined.

Note that the above method finds the five ellipse parameters in *three* stages: first the positional coordinates are obtained, then the orientation, and finally the size and eccentricity.[5] This three-stage calculation involves less computation but compounds any errors—in addition, edge orientation errors, although low, become a limiting factor. For this reason, Yuen et al. (1988) tackled the problem by speeding up the HT procedure itself rather than by avoiding a direct assault on

[5] Strictly, the eccentricity is $e = (1 - b^2/a^2)^{1/2}$, but in most cases we are more interested in the ratio of semiminor to semimajor axes, b/a.

Eq. (12.14): i.e., they aimed at a fast implementation of a thoroughgoing second stage, which finds all the parameters of Eq. (12.14) in one 3-D parameter space.

It is now clear that reasonably optimal means are available for finding the orientation and semi-axes of an ellipse once its position is known: the weak point in the process appears to be that of finding the ellipse initially. Indeed, the two approaches for achieving this that have been described above are particularly computation intensive, mainly because they examine all pairs of edge points; a possible alternative is to apply the generalized Hough transform (GHT), which locates objects by taking edge points singly: this possibility will be considered in Chapter 13.

12.7 HUMAN IRIS LOCATION

Human iris location is an important application of computer vision for three reasons: (1) it provides a useful cue for human face analysis; (2) it can be used for the determination of gaze direction; and (3) it is useful in its own right for biometric purposes, that is to say, for identifying individuals almost uniquely. The latter possibility has already been noted in Chapter 8 where textural methods for iris recognition are outlined and some key references are given. More details of human face location and analysis will be given in Chapter 17. Here we concentrate on iris location using the Hough transform.

In fact, we can tackle the iris location and recognition task reasonably straightforwardly. First, if the head has been located with reasonable accuracy, then it can form a region of interest, inside which the iris can be sought. In a front view with the eyes looking ahead, the iris will be seen as a round often high-contrast object, and can be located straightforwardly with the aid of a HT (Ma et al., 2003). In some cases, this will be less easy because the iris is relatively light and the color may not be distinctive—although a lot will depend on the quality of the illumination. Perhaps more important, in some human subjects, the eyelid and lower contour of the eye may partially overlap the iris (Fig. 12.15(a)), making it more difficult to identify, although, as confirmed below, HTs are capable of coping with a fair degree of occlusion.

Note that the iris will appear elliptical if the eyes are not facing directly ahead; in addition, the shape of the eye is far from spherical and the horizontal diameter is larger than the vertical diameter—again making the iris appear elliptical (Wang and Sung, 2001). In either case, the iris can still be detected using a Hough transform. Furthermore, once this has been done, it should be possible to estimate the direction of gaze with a reasonable degree of accuracy (Gong et al., 2000), thereby taking us further than mere recognition. (The fact that measurement of ellipse eccentricity would lead to an ambiguity in the gaze direction can be offset by measuring the position of the ellipse on the eyeball.) Finally, Toennies et al. (2002) showed that the HT can be used to localize the irises for real-time applications, in spite of quite substantial partial occlusion by the eyelid and lower contour of the eye.

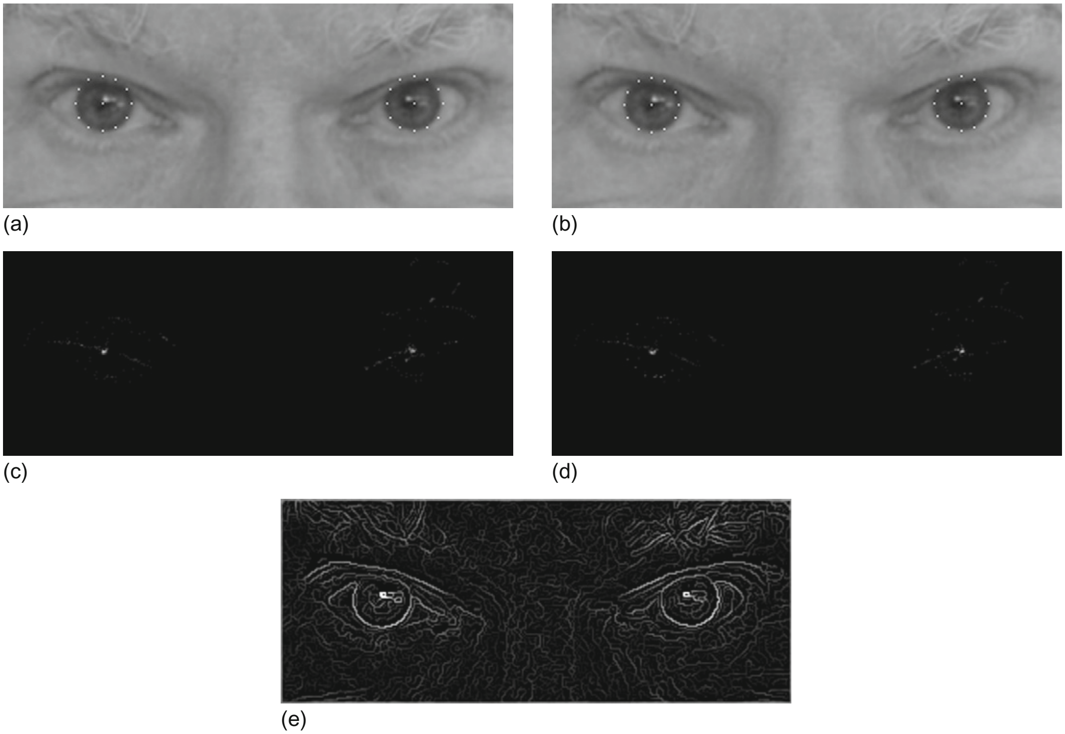


FIGURE 12.15

Iris location using the Hough transform. (a) Original image of the eye region of the face with irises located by a uniformly weighted Hough transform (HT). (b) Irises located using a gradient-weighted HT. (c) and (d) The respective HTs used to locate the irises in (a) and (b). (e) The Canny operator (incorporating smoothing, nonmaximum suppression, and hysteresis) is used to obtain the initial edge image in both cases. The sharper peaks obtained using the gradient weighting permit the irises to be located significantly more robustly and accurately. Note the number of additional edges in (e) that are able to produce substantive numbers of additional votes, which interfere with those from the irises.

A number of the points made above are illustrated by the example in Fig. 12.15. Far from being a trivial application of the HT, there are a surprisingly large number of edges in the eye region: these produce significant numbers of additional votes, which interfere with those from the irises. These arise from the upper and lower eyelids and the folds of skin around them. Hence, the accuracy of the method is not assured: this makes gradient weighting (see Section 13.6) especially valuable. The radii of the irises shown in Fig. 12.15 are about 17.5 pixels, and there is no particular evidence of ellipticity in their shape. However, more accurate location of the iris and measurement of ellipticity in order to estimate orientation (e.g., to determine the angle of gaze) require considerably greater resolution, with iris radii approaching 100 pixels, whereas pictures that analyze iris texture patterns for biometric purposes require even larger radii.

12.8 HOLE DETECTION

At this stage, an obvious question is whether the usual methods for circular object detection can be applied to hole detection. In principle, this is undoubtedly practicable, in many cases the only change arising because holes have negative contrast. This means that if the HT technique is used, votes will have to be accumulated on moving a distance $-R$ along the direction of the local edge normal. There is no difficulty with this for large holes, but as the holes become smaller, complications tend to occur because of poor lighting inside the holes. As a result, large regions of shadow are likely to occur, and the contrast is liable to be weak and variable. We shall not consider the general situation further as it is so dependent on size, illumination, and other factors, such as 3-D shape and the material within which the hole appears.

Nevertheless, for small holes a few pixels across, detection can be important, as they constitute point features, and these can be valuable for precise object location, as happens for biscuits, hinges, brackets, and a host of other manufactured parts (see Chapter 14). In fact, circular holes may even be preferable to corners for object location, as they are, ideally, isotropic and thus less liable to bias when used for measurement.

Very small holes can be detected using template matching. This involves applying a Laplacian type of operator, and in a 3 × 3 neighborhood this may take the form of one of the following convolution masks:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} -2 & -3 & -2 \\ -3 & 20 & -3 \\ -2 & -3 & -2 \end{bmatrix}$$

Having applied the convolution, the resulting image is thresholded to indicate regions where holes might be present. Note that the coefficients in each of the above masks sum to zero so that they are insensitive to varying levels of background illumination.

If the holes in question are larger than 1–2 pixels in diameter, the above masks will prove inadequate. Clearly, larger masks are required if larger holes are to be detected efficiently at their centers. For much larger holes, the masks that are required become impracticably large, and the HT approach becomes a more attractive option.

12.9 CONCLUDING REMARKS

This chapter has described techniques for circle and ellipse detection, starting with the HT approach. Although the HT is found to be effective and highly robust against occlusions, noise, and other artifacts, it incurs considerable storage and computation—especially if it is required to locate circles of unknown radius or if high accuracy is required. Techniques have been described whereby the latter two problems can be tackled much more efficiently; in addition, a method has been described for markedly reducing the computational load involved in circle detection. The general circle location scheme is a type of HT but although the other two methods are related to the HT, they are distinct methods, which draw on the HT for inspiration. As with the HT, these methods achieve robustness as an integral part of their design—i.e., robustness is not included as an afterthought—so they achieve known levels of robustness.

Two HT-based schemes for ellipse detection have also been described—the diameter bisection method and the chord−tangent method. A further approach to ellipse detection, based on the generalized HT, will be covered in Chapter 13. At that point, further lessons will be drawn on the efficacies of the various methods.

As in the case of line detection, a trend running through the design of circle and ellipse detection schemes is the deliberate splitting of algorithms into two or more stages. This is useful for keying into the important and relevant parts of an image prior to finely discriminating one type of object or feature from another, or prior to measuring dimensions or other characteristics accurately. Indeed, the concept can be taken further, in that the efficiencies of all the algorithms discussed in this chapter have been improved by searching first for edge features in the image. The concept of two-stage template matching is therefore deep seated in the methodology of the subject and is developed further in later chapters. Although two-stage template matching is a standard means of increasing efficiency (VanderBrug and Rosenfeld, 1977; Davies, 1988g), it is not obvious that efficiency can always be increased in this way. It appears to be in the nature of the subject that ingenuity is needed to discover means of achieving this.

> The Hough transform is very straightforwardly applied to circle detection, for which it achieves an impressive level of robustness. It is also easily applied to ellipse detection. Practical issues include accuracy, speed, and storage requirements—some of which can be improved by employing parameter spaces of reduced dimension, although this can affect the specificity that can be achieved.

12.10 BIBLIOGRAPHICAL AND HISTORICAL NOTES

The Hough transform was developed in 1962 and first applied to circle detection by Duda and Hart (1972). However, the now standard HT technique, which makes use of edge orientation information to reduce computation, only emerged 3 years later (Kimme et al., 1975). The author's work on circle detection for automated inspection required real-time implementation and also high accuracy. This spurred the development of the three main techniques described in Sections 12.3−12.5 (Davies, 1987f, 1988b, 1988e). In addition, the author has considered the effect of noise on edge orientation computations, showing in particular their effect in reducing the accuracy of center location (Davies, 1987e).

Yuen et al. (1989) reviewed various existing methods for circle detection using the HT. In general, their results confirmed the efficiency of the method described in Section 12.3 when circle radius is unknown, although they found that the two-stage process that was involved can sometimes lead to slight loss of robustness. It appears that this problem can be reduced in some cases by using a modified version of the algorithm of Gerig and Klein (1986). Note that the Gerig and Klein approach is itself a two-stage procedure: it is discussed in detail in Chapter 13. More recently, Pan et al. (1995) have increased the speed of computation of the HT by prior grouping of edge pixels into arcs, for an underground pipe inspection application.

The two-stage template matching technique and related approaches for increasing search efficiency in digital images were known by 1977 (Nagel and Rosenfeld, 1972; Rosenfeld and VanderBrug, 1977; VanderBrug and Rosenfeld, 1977) and have undergone further development since then—especially in relation to particular applications such as those described in this chapter (Davies, 1988g).

Later, Atherton and Kerbyson (1999) (see also Kerbyson and Atherton, 1995) showed how to find circles of arbitrary radius in a single parameter space using the novel procedure of coding radius as a phase parameter and then performing accumulation with the resulting phase-coded annular kernel. Using this approach, they attained higher accuracy with noisy images; Goulermas and Liatsis (1998) showed how the HT could be fine tuned for the detection of fuzzy circular objects such as overlapping bubbles by using genetic algorithms. In effect, the latter are able to sample the solution space with very high efficiency and hand over cleaner data to the following HT.

The ellipse detection sections are based particularly on the work of Tsuji and Matsumoto (1978), Tsukune and Goto (1983), and Yuen et al. (1988); for a fourth method (Davies, 1989a) using the GHT idea of Ballard (1981) in order to save computation, see Chapter 13. The contrasts between these methods are many and intricate, as this chapter has shown. In particular, the idea of saving dimensionality in the implementation of the GHT appears also in a general circle detector (Davies, 1988b). At that point in time, the necessity for a multistage approach to determination of ellipse parameters seemed proven, although somewhat surprisingly the optimum number of such stages was just two.

Later algorithms represented moves to greater degrees of robustness with real data by explicit inclusion of errors and error propagation (Ellis et al., 1992); increased attention was subsequently given to the verification stage of the Hough approach (Ser and Siu, 1995). In addition, work was carried out on the detection of superellipses, which are shapes intermediate in shape between ellipses and rectangles, although the technique used (Rosin and West, 1995) was that of segmentation trees rather than HTs (nonspecific detection of superellipses can of course be achieved by the diameter bisection method (see Section 12.6.1); see also Rosin (2000)).

For cereal grain inspection, with typical flow rates in excess of 300 grains per second, ultrafast algorithms were needed and the resulting algorithms were limiting cases of chord-based versions of the HT (Davies, 1999b, 1999d); a related approach was adopted by Xie and Ji (2002) for their efficient ellipse detection method; Lei and Wong (1999) employed a method that was based on symmetry, and this was found to be able to detect parabolas and hyperbolas as well as ellipses:[6] it was also reported as being more stable than other methods since it does not have to calculate tangents or curvatures; the latter advantage has also been reported by Sewisy and Leberl (2001). The fact that even in the 2000s, basic new ellipse detection schemes are being developed says something about the science of image analysis: even today the toolbox of algorithms is incomplete, and the science of how to choose between items in the toolbox, or how, *systematically*, to develop new items for the toolbox, is immature. Further, although all the parameters for specification of such a toolbox may be known, knowledge about the possible tradeoffs between them is still limited.

12.10.1 More Recent Developments

Much work has recently been carried out on iris detection using the HT. Jang et al. (2008) were particularly concerned with overlap of the iris region by the upper and lower eyelids, and used a parabolic version of the HT to accurately locate their boundaries, taking special care to limit the computational load. Li et al. (2010) used a circular HT to locate the iris and a RANSAC-like technique for locating the upper and lower eyelids, again using a parabolic model for the latter: their approach was designed to cope with very noisy iris images. Chen et al. (2010) used a circular HT to locate the iris and a straight line HT to locate up to two line segment approximations to the boundaries of each of the eyelids. Cauchie et al. (2008) produced a new version of the HT to accurately locate common circle centers from circular or partial circle segments, and demonstrated its value for iris location. Min and Park (2009) used a circular HT for iris detection, a parabolic HT for eyelid detection, and followed this with eyelash detection using thresholding.

Finally, we summarize the work carried out by Guo et al. (2009) to overcome the problems of dense sets of edges in textured regions. To reduce the impact of such edges, a measure of isotropic surround suppression was introduced: the resulting algorithm gave small weights to edges in texture regions and large weights to edges on strong and clear boundaries when accumulating votes in Hough space. The approach gave good results when locating straight lines in scenes containing man-made structures such as buildings.

[6] Note that while this is advantageous in some applications, the lack of discrimination could prove to be a disadvantage in other applications.

12.11 PROBLEMS

1. Prove the result of Section 12.4.1 that as D approaches C and d approaches zero, the shape of the locus becomes a circle on DC as diameter.
2. **a.** Describe the use of the Hough transform for circular object detection, assuming that object size is known in advance. Show also how a method for detecting ellipses could be adapted for detecting circles of unknown size.

   b. A new method is suggested for circle location that involves scanning the image both horizontally and vertically. In each case, the midpoints of chords are determined and their x or y coordinates are accumulated in separate 1-D histograms. Show that these can be regarded as simple types of Hough transform, from which the positions of circles can be deduced. Discuss whether any problems would arise with this approach; consider also whether it would lead to any advantages relative to the standard Hough transform for circle detection.

   c. A further method is suggested for circle location. This again involves scanning the image horizontally, but in this case, for every chord that is found, an estimate is immediately made of the *two* points at which the center could lie, and votes are placed at those locations. Work out the geometry of this method, and determine whether this method is faster than the method outlined in (b). Determine whether the method has any disadvantages compared with method described in (b)?
3. Determine which of the methods described in this chapter will detect (a) hyperbolas, (b) curves of the form $Ax^3 + By^3 = 1$, and (c) curves of the form $Ax^4 + Bx + Cy^4 = 1$.
4. Prove Eq. (12.11) for an ellipse. *Hint*: Write the coordinates of P and Q in suitable parametric forms, and then use the fact that OP⊥OQ to eliminate one of the parameters from the left-hand side of the equation.
5. Describe the *diameter bisection* and *chord–tangent* methods for the location of ellipses in images, and compare their properties. Justify the use of the chord–tangent method by proving its validity for circle detection and then extending the proof to cover ellipse detection.
6. Round coins of a variety of sizes are to be located, identified, and sorted in a vending machine. Discuss whether the chord–tangent method should be used for this purpose instead of the usual form of Hough transform circle location scheme operating within a 3D (x, y, r) parameter space.
7. Outline each of the following methods for locating ellipses using the Hough transform: (a) the *diameter bisection* method and (b) the *chord–tangent* method. Explain the principles on which these methods rely. Determine which is more robust and compare the amounts of computation each requires.
8. For the diameter bisection method, searching through lists of edge points with the right orientations can take excessive computation. It is suggested that a

two-stage approach might speed up the process: (a) load the edge points into a table, which may be addressed by orientation and (b) look up the right edge points by feeding appropriate orientations into the table. Estimate how much this would be likely to speed up the diameter bisection method.

9. It is found that the diameter bisection method sometimes becomes confused when several ellipses appear in the same image, and generates false “centers” that are not situated at the centers of any ellipses. It is also found that certain other shapes are detected by the diameter bisection method. Ascertain in each case quite what the method is sensitive to, and consider ways in which these problems might be overcome.