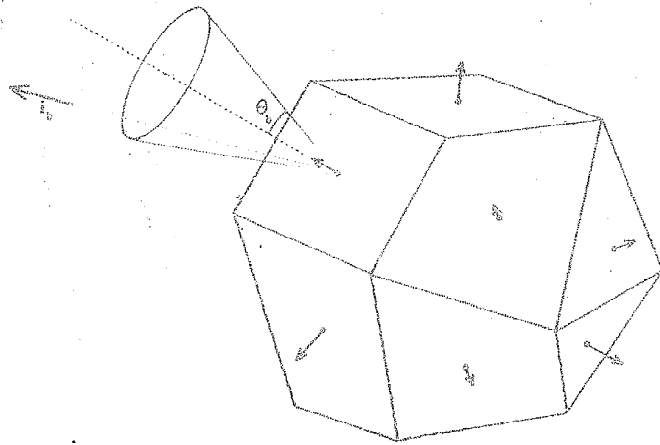


Exhibit L



Machine Vision

Theory
Algorithms
Practicalities

2nd Edition

E. R. Davies

ACADEMIC PRESS



SAMNDCA00031977

ts Applications

'K

rocessing, Signal Processing Research Institute,
1

gineering,

tems Engineering,

gineering,
m, UK

gineering,
ughborough, UK

atment,
A

Machine Vision: Theory, Algorithms, Practicalities

2nd Edition

E.R. DAVIES

*Department of Physics
Royal Holloway
University of London
Egham, Surrey, UK*



ACADEMIC PRESS

San Diego London Boston
New York Sydney Tokyo Toronto

SAMNDCA00031978

This book is printed on acid-free paper

Copyright © 1997 by ACADEMIC PRESS

All Rights Reserved

No part of this publication may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Academic Press, Inc.
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
<http://www.apnet.com>

Academic Press Limited
24-28 Oval Road, London NW1 7DX, UK
<http://www.hbuk.co.uk/ap/>

ISBN 0-12-206092-X

Library of Congress Cataloging-in-Publication Data

A catalogue record for this book is available from the British Library

Typeset by Mathematical Composition Setters Ltd, Salisbury
Printed in Great Britain by The University Press, Cambridge

96 97 98 99 00 01 EB 9 8 7 6 5 4 3 2 1

Series Preface

Signal processing applications are consumer products through to the systems extensively exploit this technology since the 1960s by the introduction of cheap processing algorithms in real-time semiconductor technology has developed in parallel, an ever increasing body of signal processing algorithms have been known and well developed.

Signal Processing and its Applications covers the depth of the subject with texts and applications of signal processing in the composition of the Editorial Board, v

- (i) Theory – The physics of the system and how to model the system;
- (ii) Implementation – VLSI/ASIC, numerical methods, systems;
- (iii) Applications – Speech, sonar, communications (both audio and video), sensing, imaging, survey, intrusion testing, and personal communications.

Signal Processing and its Applications is suitable for postgraduate students, academics, and

SAMNDCA00031979

Ellipse Detection

11.1 Introduction

It was seen in the previous chapter that very many manufactured goods are circular or else possess distinctive circular features: this makes it imperative to have an efficient set of algorithms for detecting circular shapes in digital images. However, in many situations such objects are viewed obliquely and so efficient ellipse detectors are also required. Furthermore, some objects are actually elliptical and this also gives motivation for finding effective ellipse detection algorithms.

This chapter describes some of the algorithms that have been developed for this purpose. We concentrate on parallel algorithms, since sequential methods have already been covered in Chapter 7. In addition, parallel algorithms such as the Hough transform tend to be more robust: in fact, all the algorithms described here are based on one or other form of the Hough transform. Although Duda and Hart (1972) considered curve detection, the first method that made use of edge orientation information to reduce the amount of computation was that of Tsuji and Matsumoto (1978). This ingenious method is described in the next section.

11.2 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 centre points of the connecting lines for all such pairs are taken as voting positions in parameter space (Fig. 11.1). 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 centres.

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 centre 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 Hough transform that votes must be accumulated in parameter space at all points which *could in principle* lead to correct object centre location: it is left to the peak finder to find the voting positions that are most likely to correspond to object centres.

Not only does clutter lead to wasted computation but the method itself 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 would lead to $\binom{m}{2} \approx m^2/2$ pairs of edge points). Indeed, since there are likely to be at least 1000 edge points in a typical 256×256 image, the computational problems can be formidable. The situation is examined more carefully in Section 11.6.

It should be noted that 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

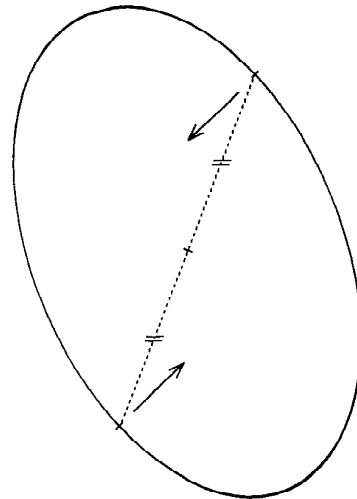


Fig. 11.1 Principle of the diameter bisection method. A pair of points is located for which the edge orientations are antiparallel. If such a pair of points lies on an ellipse, the midpoint of the line joining the points will be situated at the centre of the ellipse.

“ovals” such as ellipses with compressed ends. 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. 11.2). 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:

$$1/OP^2 + 1/OQ^2 = 1/R^2 = \text{constant} \quad (11.1)$$

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 equation (11.1)). 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, oval or other symmetrical shape may be present and each of these would need its own identifying test.

It will be clear from the above description that the method 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. 11.3). Finally, it should not go unnoticed that the method wastes the signal available from unmatched edge points.

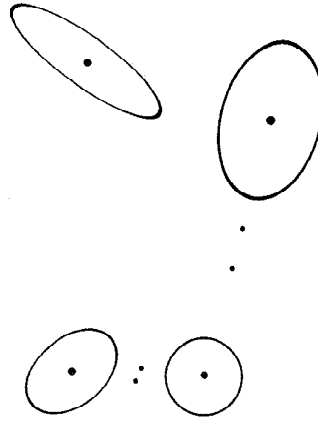


Fig. 11.2 Result of using the basic diameter bisection method. The large dots show true ellipse centres found by the method, while the smaller dots show positions at which false alarms commonly occur. Such false alarms are eliminated by applying the test described in the text.

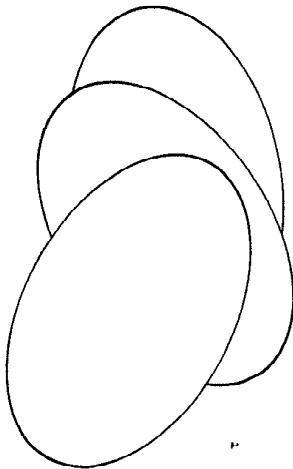


Fig. 11.3 Limitations of the diameter bisection method: of the three ellipses shown, only the lowest one cannot be located by the diameter bisection method.

These considerations have led to a search for further methods of ellipse detection.

11.3 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 ; then the equation of the line TM is found and all points that lie on the portion MK of this line are accumulated in parameter space (Fig. 11.4) (clearly, T and the centre 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 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: (i) cutting down the lengths of the lines of votes accumulated in parameter space by taking account of the expected sizes and spacings of ellipses; (ii) not pairing edge points initially if they are too close together or too far apart; and (iii)

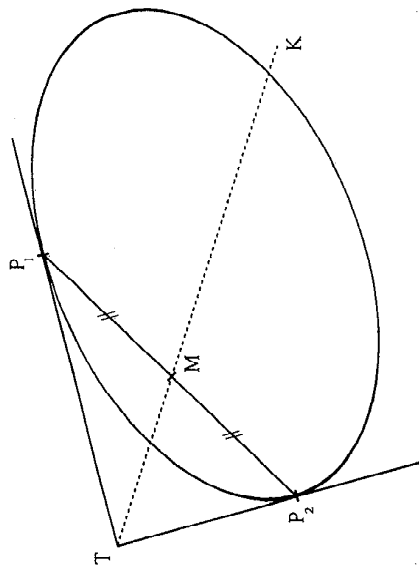


Fig. 11.4 Principle of the chord-tangent method. The tangents at P_1 and P_2 meet at T and the midpoint of P_1P_2 is M . The centre C of the ellipse lies on the line TM produced. Notice that M lies between C and T . Hence the transform for points P_1 and P_2 need only include the portion MK of this line.

eliminating edge points once they have been identified as belonging to a particular ellipse (clearly, the latter two techniques are also applicable to the diameter bisection method).

Section 11.5 explores in more detail whether computation can be saved by reverting to the GHT approach. Meanwhile, it is necessary to consider how ellipse detection can be augmented by suitable procedures for finding ellipse orientation, size and shape.

11.4 Finding the remaining ellipse parameters

Although it has proved possible to locate ellipses in digital images by simple geometric constructions based on well-known properties of the ellipse, a more formal approach is required to determine other ellipse parameters. Accordingly, we write the equation of an ellipse in the form:

$$Ax^2 + 2Hxy + By^2 + 2Gx + 2Fy + C = 0 \quad (11.2)$$

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

$$AB > H^2 \quad (11.3)$$

This condition guarantees that A can never be zero and that the ellipse equation may without loss of generality be rewritten with $A = 1$; in any case, only ratios between the parameters have physical significance. 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 centre of the ellipse, we may select a new origin of coordinates at its centre (x_c, y_c) ; the equation then takes the form:

$$x'^2 + 2Hx'y' + By'^2 + C' = 0 \quad (11.4)$$

where

$$x' = x - x_c; \quad y' = y - y_c \quad (11.5)$$

It now remains to fit to equation (11.4) the edge points that gave evidence for the ellipse centre under consideration. The problem will in general 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. Since Hough parameter spaces require storage that increases exponentially with the number of unknown parameters, it is tempting to try to separate the determination of some of the parameters from that of others. For example, we can follow Tsukune and Goto (1983) by differentiating equation (11.4):

$$x' + By' \, dy'/dx' + H(y' + x' \, dy'/dx') = 0 \quad (11.6)$$

Then dy'/dx' 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 again be used with equation (11.4) to obtain a histogram of C' values, from which the final parameter for the ellipse can be obtained.

The following formulae must be used to determine the orientation and semi-axes values of an ellipse in terms of H, B and C' :

$$\theta = (1/2) \arctan [2H/(1 - B)] \quad (11.7)$$

$$a^2 = \frac{-2C'}{(B + 1) - [(B - 1)^2 + 4H^2]^{1/2}} \quad (11.8)$$

$$b^2 = \frac{-2C'}{(B + 1) + [(B - 1)^2 + 4H^2]^{1/2}} \quad (11.9)$$

Mathematically, θ is the angle of rotation that diagonalizes the second-order terms in equation (11.4); having performed this diagonalization, the ellipse is then essentially in the standard form $x'^2/a^2 + y'^2/b^2 = 1$, so a and b are determined.

It should be noted 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*. This three-stage calculation involves less computation but compounds any errors—in addition, edge orientation errors, though 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 equation (11.4): i.e. they aimed at a fast implementation of a thoroughgoing second stage which finds all the parameters of equation (11.4) in one 3-D parameter space. Their fast adaptive HT procedure has already been described in Chapter 10 and is not covered again here.

At this stage it is 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 computationally intensive, mainly because they examine all pairs of edge points; hence it seems worth considering again the GHT approach, which locates objects by taking edge points singly, to see whether any gains can be achieved by this means.

11.5 Reducing computational load for the generalized Hough transform method

The complications described in Chapter 10 when the GHT is used to detect anisotropic objects clearly apply to ellipse detection: the various possible orientations of the object lead to the need to employ a large number of planes in parameter space. So far this chapter has avoided considering the GHT for these reasons. However, it will be seen below that by accumulating

* 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 .

the votes for all possible orientations in a *single* plane in parameter space, significant savings in computation can frequently be made. Basically, the idea is largely to eliminate the enormous storage requirements of the GHT by using only one instead of 360 planes in parameter space, while at the same time reducing by a large factor the computation involved in the final search for peaks. Such a scheme may 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 analyse the shape of the point spread function (PSF) to be accumulated for each edge pixel, in the case of an ellipse 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 centre of the ellipse relative to the local edge normal (Fig. 11.5). Working first in an ellipse-based axes system, for an ellipse with semimajor and semiminor axes a and b , respectively, it is clear that:

$$x = a \cos \psi \quad (11.10)$$

$$y = b \sin \psi \quad (11.11)$$

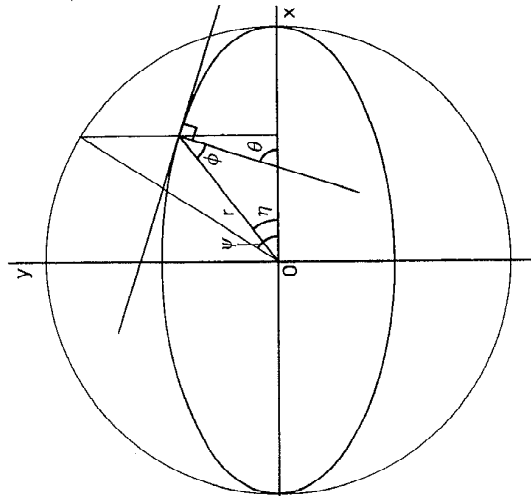


Fig. 11.5 Geometry of an ellipse and its edge normal.

Hence

$$dx/d\psi = -a \sin \psi \quad (11.12)$$

$$dy/d\psi = b \cos \psi \quad (11.13)$$

giving

$$dy/dx = -(b/a) \cot \psi \quad (11.14)$$

Hence the orientation of the edge normal is given by:

$$\tan \theta = (a/b) \tan \psi \quad (11.15)$$

At this point we wish to deduce the bearing φ of the centre of the ellipse relative to the local edge normal. From Fig. 11.5:

$$\varphi = \theta - \eta \quad (11.16)$$

where

$$\tan \eta = y/x = (b/a) \tan \psi \quad (11.17)$$

and

$$\begin{aligned} \tan \varphi &= \tan (\theta - \eta) \\ &= \frac{\tan \theta - \tan \eta}{1 + \tan \theta \tan \eta} \end{aligned} \quad (11.18)$$

Substituting for $\tan \theta$ and $\tan \eta$, and then rearranging, gives:

$$\tan \varphi = \frac{(a^2 - b^2)}{2ab} \sin 2\psi \quad (11.19)$$

In addition:

$$r^2 = a^2 \cos^2 \psi + b^2 \sin^2 \psi \quad (11.20)$$

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. 11.6). The PSF is then the locus of all possible positions of the centre of the ellipse. To find its form it is merely required to eliminate ψ between equations (11.19) and (11.20). 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):

$$r^2 = \frac{a^2 + b^2}{2} + \frac{a^2 - b^2}{2} \cos 2\psi \tag{11.21}$$

After some manipulation the locus is obtained as:

$$r^4 - r^2(a^2 + b^2) + a^2b^2 \sec^2 \varphi = 0 \tag{11.22}$$

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

$$v^2 = (a^2 + b^2) - u^2 - a^2b^2/u^2 \tag{11.23}$$

Unfortunately this is not a circle, and is not even symmetrical about axes through its centre. The shape of the PSF for various values of a and b is shown in Fig. 11.7. For very highly eccentric ellipses, the PSF is approximated by the arcs of two circles, one of radius a and the other of radius a^2/b . It will be seen that in the other extreme, of ellipses of low eccentricity, the PSF is approximated by an ellipse. In general, however, it is easy to see

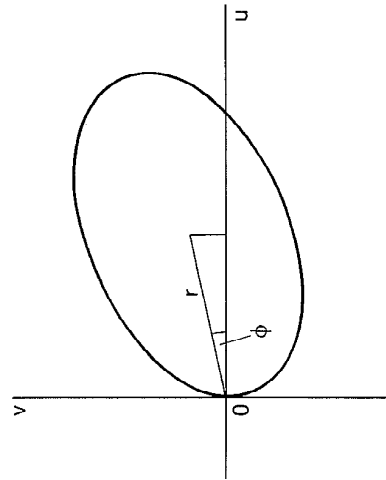


Fig. 11.6 Geometry for finding the PSF for ellipse detection by forming the locus of the centres of ellipses touching a given edge fragment.

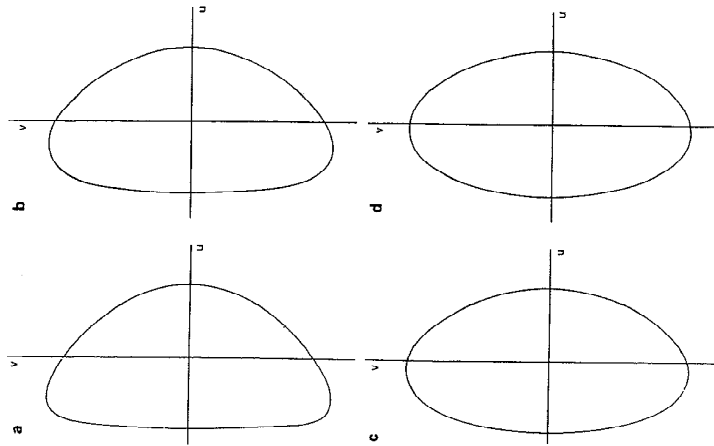


Fig. 11.7 Typical PSF shapes for detection of ellipses with various eccentricities: (a) ellipse with $a/b = 21.0$, $c/d = 1.1$; (b) ellipse with $a/b = 5.0$, $c/d = 1.5$; (c) ellipse with $a/b = 1.4$, $c/d = 5.0$; (d) ellipse with $a/b = 2.0$, $c/d = 3.0$. Notice how the PSF shape approaches a small ellipse of aspect ratio 2.00 as eccentricity tends to zero. The semimajor and semiminor axes of the PSF are $2d$ and d , respectively.

that, since the original ellipse possesses regions along the principal axes where the radii of curvature are approximately constant with values a^2/b and b^2/a , the PSF also possesses regions at either end of one diameter where the radii of curvature are $(a^2/b) - b$ and $a - (b^2/a)$; this makes it clear that the shape is not symmetrical, and also that it should become approximately symmetrical when $a \approx b$.

Defining two new parameters:

$$c = (a + b)/2 \tag{11.24}$$

$$d = (a - b)/2 \tag{11.25}$$

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

$$(u + c + d^2/2c)^2/d^2 + v^2/4d^2 = 1 \quad (11.26)$$

Neglecting the term $d^2/2c$ which provides a small correction to u , the PSF is itself approximately an ellipse of semimajor axis $2d$ and semiminor axis d . Making use of this fact simplifies implementation but it is practicable only for low-eccentricity ellipses where $d \leq 0.1c$, so that $a \leq 1.2b$. 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.

11.5.1 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 now been seen to happen in several line and circle detection schemes—see Chapters 8 and 9). Hence a very considerable amount of computation can be saved. This is clearly demonstrated by the simplicity of the final code for computing a given contribution to the PSF (Table 11.1).

Figure 11.8 shows the result of testing the above scheme on an image of some O-rings lying on a slope of arbitrary direction. 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

Table 11.1 Algorithm for ellipse detection.

```

for all pixels in image do (* scan over image *)
begin
  findgx; (* apply edge operator *)
  findgy;
  g := magnitude[gx, gy]; (* lookup magnitude *)
  if g > gthreshold then (* if edge point then *)
    for i := 1 to tablesize do (* run over PSF *)
      begin
        dx := c + d * xtable[i]; (* scale and position *)
        dy := d * ytable[i];
        xc := xedge - round((dx * gx - dy * gy)/g);
        yc := yedge - round((dx * gy + dy * gx)/g);
        (* rotate and position *)
        vote[xc, yc] := vote[xc, yc] + 1; (* accumulate vote *)
      end;
    end; (* end image scan *)
  end;

```

In this section of code, it is arbitrarily assumed (i) that the magnitude of the vector (g_x, g_y) is available from a lookup table, and (ii) that all calculations are carried out in floating-point form until the final "rounding" stage where voting positions are being estimated to the nearest pixel.

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 11.8 also shows the arrangement of points in parameter space that results from applying the PSF to every edge point on the boundary of an ellipse: the pattern is somewhat clearer in Fig. 11.9. In either case it is seen to contain a high degree of structure. For an ideal transform, there would be no structure apart from the main peak. Looking at the situation in another way, in deducing the locus of points forming the PSF we found all points that could possibly be at the centre of an ellipse when we had no knowledge about its orientation: all points on the PSF *not* falling on the peak at the centre of the ellipse would hopefully be randomly (and fairly uniformly) distributed nearby. However, for a shape such as an ellipse—which has a particular structure and symmetry of its own—this is not the situation. In fact, a full computation of the positions of votes in parameter space yields very many candidate positions, and to find strong structure in the transform it is necessary to look for *envelopes* to the system of PSFs. This is a sufficiently complex task in this case to be left to experiment, but one or two observations on the results are useful and relevant. First, where there is a lot

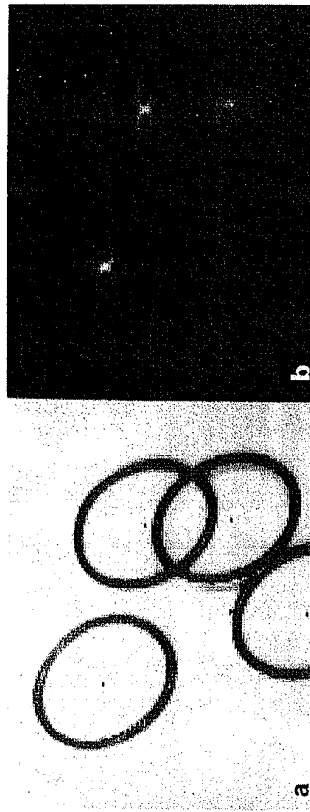


Fig. 11.8 Applying the PSF to detection of tilted circles: (a) off-camera 128×128 image of a set of circular O-rings on a 45° slope of arbitrary direction; (b) transform in parameter space: notice the peculiar shape of the ellipse transform, which is close to a "four-leaf clover" pattern. Part (a) also shows the positions of the centres 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.



Fig. 11.9 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.

of local symmetry in the starting shape (the ellipse)—and specifically where the curvature on the boundary is approximately constant—this leads automatically to strong envelope lines in the transform, as indicated in Fig. 11.10, where the on-axes regions of the original ellipse have essentially been moved radially (and are hence also compressed laterally) until they pass through its centre. These lines seem practically to form a "four-leaved

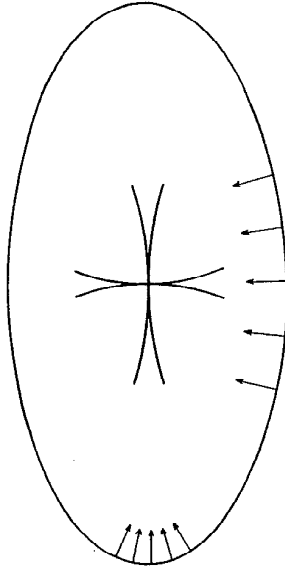


Fig. 11.10 Formation of the ellipse transform by moving the regions of the ellipse boundary lying roughly along the principal axes radially inwards until they pass through the centre of the ellipse. The reason for this peculiar construction lies in the fact that the prominent lines in the transform are the envelopes of all possible PSFs: the necessary coherence is provided by the nearly constant radii of curvature roughly along the directions of the principal axes.

clover" pattern, although the loops become more diffuse on moving away from the centre, in accordance with the envelope idea expressed above.

The important problem here is whether the structure in the transform is misleading, i.e. whether it gives rise to sharply focused peaks at positions other than the main peak. In fact this appears not to happen, the main and most intense structure being the cross pattern at the centre. This is clearly strongest and most focused (both practically and theoretically) near the centre while, in addition, it goes up to double the height quite sharply as the two branches cross at the very centre. The overall conclusion is that the additional structure does no particular harm, although it gives rise to interesting complications if two ellipses overlap so that their centres are quite close. Hence it is more difficult to use the method described here to disentangle two such ellipses than it would be to disentangle two circular objects with a similar degree of overlap. The way around this problem is to check individual high points in the transform by interrogating the original image—see for example, Bolles and Cain (1982), Davies (1987g) and Chapters 12 and 14.

It is interesting that more help would be obtained in interpreting the transform in such cases if the PSFs were regarded not as mere loci of the centres of ellipses of different orientations touching a given edge fragment, but as loci with proper weights, unit weight corresponding to unit boundary distance on the original ellipse. This is too complex a problem to be pursued here, but it may be noted that curvature information immediately leads to the prediction of larger weights for the part of the PSF corresponding to the two regions of the ellipse along the minor axes.

Finally, note that accuracy will be lost if the PSF is not guaranteed to place a vote at the centre of the ellipse for every edge pixel. This means that the PSF should be a connected chain of pixels: hence the size of the LUT must vary with the size of the PSF—or, if a large ULUT is used, a small PSF should be able to skip over a proportion of the entries. In the examples of Figs 11.8 and 11.9, the PSF contained 50 and 100 votes, respectively.

11.6 Comparing the various methods

This section briefly compares the computational loads for the methods of ellipse detection discussed above. To make fair comparisons, we concentrate on ellipse location *per se* and ignore any additional procedures concerned with (i) finding other ellipse parameters, (ii) distinguishing ellipses from other shapes, or (iii) 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 , d defined in Section 11.3. By ignoring noise and general background clutter, we shall be favouring 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:

$$L_A \approx p \times 2\pi c \times 2\pi(2d + d)/2 = 6\pi^2 pcd \approx 60pcd \quad (11.27)$$

where the ellipse has been taken to have relatively low eccentricity so that the PSF itself approximates to an ellipse of semi-axes $2d$ 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 11.1). 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:

$$L_B \approx \left(\frac{p \times 2\pi c}{2} \right) \approx (2\pi pc)^2 / 2 \approx 20p^2 c^2 \quad (11.28)$$

Hence

$$L_B/L_A \approx pc/3d \quad (11.29)$$

When a is close to b , as for a circle, $L_A \rightarrow 0$ and then the diameter bisection method becomes a very poor option. However, in many cases it is found that a is close to $2b$, so that c is close to $3d$. The ratio of the loads then becomes:

$$L_B/L_A \approx p \quad (11.30)$$

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 of Fig. 11.8). The overall image “clutter factor” would normally be expected to be at least two.

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, there seems no doubt that the GHT method will 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 is strong. The only situation where the diameter bisection method might win appears to be where an image contains a minimal number of highly eccentric ellipses. 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 must be noted the different characteristics of the methods. 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.

11.7 Concluding remarks

This chapter has examined methods for ellipse detection in digital images. The simplest such scheme is the diameter bisection method, which is unfortunately not very robust because many pairs of diameter ends need to be visible before sufficient signal can be built up. One alternative is the chord-tangent method; this is highly robust yet at the penalty of considerable additional computation. Another alternative is the GHT, but when applied to anisotropic shapes this method requires that one plane in parameter space be used for each discriminable orientation of the shape. However, if a single plane parameter space is used for this purpose, the GHT approach frequently requires less computational effort than the other two methods. This is because they are based on using pairs of edge points to deduce candidate ellipse centres, whereas the GHT takes edge points one at a time. However, the three methods differ markedly in their shape discriminability characteristics and this should be noted in real applications.

In particular, the diameter bisection method locates ellipses of all eccentricities, size and orientations, and also other symmetrical shapes; the chord-tangent method locates ellipses of all eccentricities, sizes and orientations but discriminates against other shapes; and the GHT locates ellipses of all orientations but discriminates against ellipses of the wrong eccentricity and size and against other shapes. However, the GHT can be adapted to detect many other shapes, whereas the diameter bisection method is limited and the chord-tangent method is specific to ellipses.

With the GHT, the avoidance of many planes in parameter space is one aspect of a more general problem. For example, we might need to use many

planes in parameter space to allow for variation in object size. It is also possible that we might know the size and orientation of an ellipse but not its eccentricity, many planes being needed to cope with this situation. In Chapter 9 it was shown that circles of many sizes can be sought simultaneously in a single plane in parameter space, by a method related to that described here (namely using a PSF consisting of a radial line in place of a single point): it is clear that this method will carry over to ellipses and other shapes (i.e. replace the $R(\theta)$ in the basic transform in turn by $\alpha R(\theta)$, for all α in a given range, leaving $\varphi(\theta)$ unchanged). Similarly, it is evident from the derivation in Section 11.5 that the GHT approach carries over to other shapes when orientation is unknown. Finally, a similar extension can be made when eccentricity is unknown but size and orientation are known. Thus we hypothesize generally that *parameter space can be compressed by one degree of freedom from the ideal* in order to save computation. We also note that *it is not possible to compress parameter space by more than one degree of freedom from the ideal*: if we tried to do this, virtually any shape would be able to trigger the detector and there would be a great many false alarms. As an example, if we try to detect ellipses of unknown orientation and size in a single plane in parameter space, the PSF becomes an extended blob which is unable to discriminate between a very great variety of shapes (in general, the PSF becomes an *area* which is the result of convolving the two basic PSF curves in parameter space: indeed, it is not possible to find two such curves that do not produce an area when convolved). These observations appear both to extend and ultimately to limit the capabilities of the GHT approach for object location, and are by no means limited to ellipse detection.

11.8 Bibliographical and historical notes

This chapter is based on relatively few papers, all of which were written within the last 20 years: first, that of Tsuji and Matsumoto (1978), whose insight led to a totally new ellipse detector; second, that of Tsukune and Goto (1983), which aimed at more accurate determination of ellipse parameters; third, that of Yuen *et al.* (1988), whose detector made good certain shortcomings of the previous two methods; and fourth, that of the author (Davies, 1989a), which developed the GHT idea of Ballard (1981) in order to save computation. The contrasts between these methods are many and intricate, as the 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). Finally, the necessity for a multistage

approach to determination of ellipse parameters now seems proven, although somewhat surprisingly the optimum number of such stages appears to be just two.

The fact that as late as 1989 basic new ellipse detection schemes were 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 at a rather primitive stage. Nevertheless, there are moves to greater degrees of robustness with real data by explicit inclusion of errors and error propagation (Ellis *et al.*, 1992). In addition, increased attention has been given to the verification stage of the Hough approach (Ser and Siu, 1995).

Finally, some work has been carried out on the detection of superellipses, which are intermediate in shape between ellipses and rectangles, though the technique used (Rosin and West, 1995) was that of segmentation trees rather than Hough transforms (nonspecific detection of superellipses can of course be achieved by the diameter bisection method—see Section 11.2).

11.9 Problems

1. Derive equations (11.7)–(11.9) by the method suggested in the text.
2. Determine which of the methods described in this chapter will detect (a) hyperbolas, (b) curves of the form $Ax^3 + By^3 = 1$, (c) curves of the form $Ax^4 + Bx + Cy^4 = 1$.
3. Prove equation (11.1) for an ellipse. *Hint*: Write the coordinates of P and Q in suitable parametric forms, and then use the fact that $OP \perp OQ$ to eliminate one of the parameters from the left-hand side of the equation.

12

Hole Detection

12.1 Introduction

The last few chapters have been preoccupied with the location of objects from their shapes. This is a viable approach but in more complex scenes it is better located from their features. In the case of simple shapes, features may be components of the shape, but there are two important types of feature: small holes and corners. This chapter considers the latter. These features can be used to locate objects and to identify them. This chapter considers carefully how the process of hole detection is done once its hole features (or a subset of them) have been located.

We start by considering the template matching approach. *a priori* the most obvious means of searching for holes is to examine alternative means of searching for such features. It is noted that although the chapter is oriented towards *methods* of searching for such features, it also discusses convenient vehicles which allow the researcher to experiment.

12.2 The template matching approach

When considering features such as small holes, it is useful to have a mental model with which to work.