

An Analysis of Hole Detection Schemes

E. R. Davies & S. P. Barker

Machine Vision Group, Department of Physics
Royal Holloway & Bedford New College
Egham Hill, Egham, Surrey TW20 0EX, UK

This paper studies the intricacies of hole detection, with an emphasis on the efficient location of small holes - the ultimate goal being real-time recognition and location of objects from their features. First, methods of locating circles are examined, and then the study is extended to hole location. A new form of chord-bisection scheme is found to be faster than the standard Hough transform for circle location, but the lateral histogram approach turns out to provide the most efficient available means for hole location.

Object recognition and location perform a central role in image analysis, and are particularly important in applications such as automated visual inspection and automated assembly. The most obvious means of performing object location is by template matching (correlation), and it is unfortunate that this process normally involves very considerable amounts of computation, thereby eliminating all possibility of real time object recognition.

Feature detection opens the possibility of significantly reducing computational load. Broadly, feature detection is useful because it reduces the size and number of the templates required for recognition, it being clear from this point of view that the most useful features are the smallest and least complex ones. However, there are so many feature recognition processes and schemes that it is somewhat difficult to generalise further. We therefore point to a number of types of feature (or composite feature) that have been used to initiate object recognition. They include straight lines and edges, circular and elliptic arcs, round holes, corners, and simple polygonal shapes. In addition, methods exist for detecting and analysing more arbitrary shapes that might appear around the boundaries of objects: such methods include the generalised Hough transform [1] and 1-D boundary pattern analysis techniques such as the Fourier descriptor method [2]. It is not our purpose in this paper to enter a discussion of the variety of features and feature detectors. Instead we concentrate on hole detection schemes since holes are frequently used as a starting point for unlocking image data - particularly when industrial components are being inspected and assembled.

In many ways holes are paradigm features, being compact and often quite small - hence apparently being ideal from the computational standpoint outlined above. Interestingly, their high spatial localisation should also make them ideal from the point of view of achieving high accuracy in the location and dimensional measurement of objects containing them: this will be especially true if

holes are perfectly round and hence isotropic in shape.

It has perhaps been implied above that holes are essentially different from circular objects from the point of view of feature detection. However, this need not be so, since in many cases holes are merely circular features that have negative contrast. This means that circular holes should be detectable by the same methods that apply for circular objects. While this is indeed the case, two factors should be noted. The first is that holes are frequently rather small, and this can lead to some difficulty when using normal circle detection schemes. The second is that holes often contain substantial shadow, and may well be poorly or irregularly lit. These factors require special attention when considering hole detection schemes.

This paper analyses the problems of hole detection. After a brief review of methods for detecting circles, hole detection methods are studied with particular reference to the lateral histogram approach. Next, a theoretical analysis of execution speeds for circle and hole detection schemes is carried out, and the results are compared with experimental measurements. Finally, the situation regarding the special factors in hole detection is presented as we now see it.

CIRCLE DETECTION SCHEMES

Much work has been carried out on circle detection, particularly using the Hough transform (HT) approach. In this context we mention the early work by Duda and Hart [3] and Kimme et al [4], and more recently the work of Gerig and Klein [5], Illingworth and Kittler [6], Davies [7,8], Conker [9], and the recent review by Yuen et al [10]. A good proportion of this work is orientated to the detection of circles of arbitrary size [5-7,9,10]. However, we here concentrate on circles of known size, since in many applications specific objects are to be located by their holes or circular features, and they are situated at known distance from the camera - e.g. on a worktable or conveyor; in addition, in real time applications it is important to ensure that speed is optimised for the prevailing situation. However, we note that if circles of known size can be found, then repeated application of the basic algorithm will permit circles of any size to be located [1].

The most widely used circle detection scheme is the HT. This involves scanning the input image to find edge points - e.g. using the Sobel edge detector [11] - and then moving inwards along the edge normal direction by a distance equal to the radius r of the circles being sought,

and plotting a point in a separate image space. This latter space is called a parameter space, and the process of plotting candidate centre points is in fact one of accumulation - since this is an evidence building scheme in which votes are accumulated at likely centre locations. When all the edge points in the image have been found, and all candidate centre points have been accumulated, parameter space is examined for significant peaks: when found, these are taken to indicate circle centres. This approach has the advantage of being robust in the sense of being able to locate circles that are fragmented, e.g. by noise, breakage, or occlusion, and is a major reason for its wide use.

Although the HT technique is considerably faster than template matching, it can still take significant time to run on a conventional processor. For this reason Davies [12] attempted to speed it up by a substantial factor. The basic idea was to increase the speed by a factor ~ 3 by using a more economical edge detection scheme, and to gain another factor 3 or so by sampling, i.e. by examining only every n th line in the image, n typically being in the range 3 to 8. Both strategies were found to work well in practical situations, and overall gains in speed of up to ~ 25 were recorded. The limitation was found to be a loss in robustness when small circular objects or holes are being sought, since insufficient sampling of the feature then takes place, sensitivity (in a signal-to-noise sense) suffers, and as a result the feature may not be detected. This makes the second (sampling) strategem rather less suitable for hole detection, and sampling will therefore not be considered further in this paper.

The economical edge detector used in this method uses a two-pixel differencing scheme, with mask $[-1 \ 1]$ acting in the x -direction. Bisection of the chord between each pair of edge points on the same horizontal image line gives the x -coordinate of a possible vertical diameter of a circular feature. All such x -coordinates are recorded in an x -histogram, which is later analysed for significant peaks indicating x -coordinates of circle centres. A similar scheme applies for finding the y -coordinates, and if necessary a simple check is made to find which pairs of coordinates correspond to actual circle centres. This alternative approach to centre location is used since the standard HT requires that edge orientation be known accurately, and the economical edge detector was taken in [12] to be unable to provide accurate enough orientation information.

It will be clear that if there are a number of circles in an image, then a considerable number of checks will need to be made to determine which horizontal and vertical diameter lines correspond to the same circles (though this was not a problem with the particular image data presented in [12]). One way around this problem is to revert to using a 2-D parameter space, each pair of edge points then leading to the need to accumulate a whole (horizontal or vertical) line of points in parameter space - this line being of length equal to a diameter of the circles being sought. Unfortunately this strategy requires computation to be increased considerably, and the overall speed will be slower than for the standard HT. Here

we report a modified approach in which account is taken of the distance apart of the pairs of edge points: if these are a distance d apart along the same horizontal line, then the centre will have x -coordinate given by the midpoint of the line joining the edge points, and y -coordinate given by applying Pythagoras' theorem:

$$x_c = (x_1 + x_2)/2$$

$$y_c = y \pm [r^2 - (d/2)^2]^{\frac{1}{2}} = y \pm [r^2 - (x_2 - x_1)^2/4]^{\frac{1}{2}}.$$

(Similar equations apply when edge points are scanned along a vertical direction.) It turns out that the sign ambiguity can conveniently be removed by taking note of the y -component of gradient at the edge points. Hence the overall result will be similar to that for a standard HT with 2-D parameter space, and there will be only a small saving in computation. This latter problem is overcome by use of a lookup table to determine y_c when d is known:

$$d = (x_2 - x_1)$$

$$y_c = y \pm \text{lookup}[d],$$

the sign ambiguity being resolved as before. The whole process is particularly efficient since the lookup table is one-dimensional. Note that finding peaks in the 2-D parameter space need not involve significantly more computation than finding the peaks in the two 1-D parameter spaces for the earlier version of the method, if (for example) a tally is kept during vote accumulation of where peaks are rising above certain critical levels. Our tests on this modified chord-bisection approach show it to be both effective and efficient (see also the results section below).

HOLE DETECTION SCHEMES

In this section we outline a number of schemes that have been used specifically for hole detection. First we note that it is common to employ a Laplacian type of operator for cases where very small holes (~ 1 -3 pixels diameter) are to be detected. This method is a form of the template matching technique, which would amount to a brute-force approach when larger holes are to be detected. More will be said about this in a later section.

Second, as indicated in the Introduction, the standard HT approach is frequently used for detecting large holes which are effectively indistinguishable from circular objects of negative contrast.

A third approach is the 'heuristic' hole finder [13]. This bears considerable similarities to the chord-bisection scheme outlined in the previous section. However, the actual procedure involves the following rather different pair of operations: (1) building two picture spaces in which the midpoints of horizontal and vertical lines joining pairs of edge points are plotted: this gives a set of short horizontal lines in the one space, and a set of short vertical lines in the other space; (2) seeking coincidences between the horizontal and vertical lines in the two spaces, and taking the resulting locations as hole-centres. This method seems to work reasonably well,

but the following observations are in order: (a) the final location depends on the position of just four edge points and no averaging occurs (beyond that in the rather large edge detector that is employed), so accuracy is bound to suffer; (b) if any of the critical four edge points is absent, then the centre will not be found: hence the robustness of the method is limited. Set against these limitations is the potential advantage that the method will produce a reasonable result even when the hole shape is distorted - or when the hole shape does not approximate to a circle, as for an ellipse or a square. (On the other hand, it will not work for all shapes - as in the case of a rectangular slot in certain orientations.)

The last method we describe here is the lateral histogram approach. This was first used for corner detection [14], and subsequently applied to hole detection [15]. This method involves computing two histograms representing the average x and y -variations in intensity over the image. These are then analysed to find whether there is any evidence for hole profiles in the laterally averaged intensity patterns. The result is a set of horizontal and vertical lines along which holes could exist in the input image. At this stage there is considerable ambiguity, since it is not known which horizontal and vertical lines correspond to which holes: this means that many tests may have to be made to determine exactly which crossing points correspond to hole centres.

The particular value of the lateral histogram approach is its inherent efficiency. Computation of the histograms takes some $2N^2$ operations for an $N \times N$ image, whereas location of hole profiles in the histograms takes of the order of $2nN$ operations (assuming hole templates have diameter n), and so is relatively insignificant. Disambiguation of possible hole centres is another matter, potentially taking a lot of computation. This point has been discussed in depth in [15], and will not be dwelt on further here. Suffice it to say that when the number of holes p is restricted, so that $np < N$, then the overall computation remains in the range $2N^2$ to $3N^2$, and the method is especially useful. It also has the advantage of incorporating considerable averaging, so that (a) accuracy of hole centre determination is enhanced, and (b) fuzzy objects and holes can be located reliably. Note that if noise levels are significant, or surface texture of the background objects is marked, then sensitivity will suffer, though this problem can be reduced by systematic breakdown of images into sub-images [15].

HOLE DETECTION ALGORITHMS STUDIED

In this section the execution speeds of a set of algorithms that have been used for hole detection are analysed. First we describe the specific algorithms studied.

The standard HT (SHT). The standard HT is that described in the section on circle detection and needs no more comment.

The economical HT (EHT). This version of the SHT is speeded up by use of economical 2-element edge detector masks [-1 1] etc in place of Sobel masks. Ordinarily

this would not give sufficient precision for edge orientation, but for small holes the error need not be excessive, and we have used it for benchmark tests. In addition, these small masks are more appropriate for detecting the edges of smaller holes, since larger masks could average out the small hole features being sought.

The high precision HT (HHT). This is a form of the EHT in which higher accuracy is achieved by a two-stage process in which final accumulation is performed in a local space with four times greater linear resolution: this process was found to give at least double the precision in each dimension, though the gain drops off for large radii. Again this provided a useful benchmark algorithm.

The chord-bisection HT (CHT). This is the new form of chord-bisection HT described at the end of the section on circle detection.

The heuristic hole finder (HEU). This is the method of [13] described in the section on hole detection.

The lateral histogram method (LAT). This is the method of [15] described in the section on hole detection.

The template matching method (TM). The brute-force method mentioned in the section on hole detection. (This was included as a reference benchmark rather than as a method to be considered seriously for practical use.)

Execution Speed of Hole Detection Algorithms

Execution speed is analysed here in terms of the numbers of basic arithmetic operations to be performed and their execution times. For simplicity these operations were taken to be of the three types (1) addition and subtraction, (2) multiplication and division, and (3) square root. However, overall execution times also depend on characteristics of the image data: hole radii and numbers of holes are clearly important, but amounts of noise and image clutter are also highly relevant. In the calculations presented here the latter artefacts will at first be ignored, their effects finally being considered in the light of experimental data.

Table 1 gives formulae representing the approximate overall execution times of the various algorithms. These formulae reflect the programming detail of the particular algorithms, and it will not be beneficial to consider here the exact numerical coefficients. Instead we note the following points: (1) all methods have a scanning overhead proportional to the number of pixels in the image; (2) the HT-based methods have an additional overhead proportional to the number of edge points found in the image; (3) most methods involve a certain amount of computation proportional to the number of holes, which is required to obtain an accurate estimate of the position of each hole from data near appropriate peaks in parameter space; (4) for the lateral histogram method the latter problem gives way to the need to disambiguate p^2 potential hole positions.

SHT	$[15N^2 + 13e + (4 + 8r + 8r^2)p]t_1 + [2N^2 + 5e + p]t_2 + N^2t_3$
EHT	$[3N^2 + 13e + (4 + 8r + 8r^2)p]t_1 + [2N^2 + 5e + p]t_2 + N^2t_3$
HHT	$[3N^2 + 13e + (4 + 32r + 128r^2)p]t_1 + [2N^2 + 9e + p]t_2 + N^2t_3$
CHT	$[2N^2 + 11e + (3 + 10r + 2r^2)p]t_1 + [3e + 4pr]t_2$
HEU	$[36N^2 + 14pr]t_1 + 4prt_2$
LAT	$[2N^2 + (4 + 4r)N + 4p + (20 + 24r + 8r^2)p^2]t_1 + 2Nt_2$
LAT'	$[2N^2 + (4 + 4r)N + 4p + 18p^2 + (20 + 24r + 8r^2)q^2]t_1 + 2Nt_2$
TM	$[(19 + 24r + 8r^2)N^2 + 72p]t_1 + N^2t_2$

Table 1. Formulae modelling execution times of hole detection algorithms

In this table, N^2 is the number of pixels in the image, e is the number of edge points detected, p is the number of holes (including false alarms), q is the number of true holes, and t_1 , t_2 and t_3 are the times of the three categories of arithmetic operations (+, -, *, /; and $\sqrt{\quad}$ respectively).

We have tested these algorithms on a variety of image data, and for single small holes the results are much as indicated on the $r = 1$ axis of the graph in Figure 1. Indeed, the extrapolation to the $r = 0$ axis shows the overhead commonly obtained with these methods (see for example how closely the TM curve approaches the SHT curve for small r). The rest of Figure 1 shows how the situation develops as r increases in each case. It is seen that the ordering of the execution times is unchanged as r increases, and on the whole is as expected from the formulae of Table 1. On the other hand, the numerical values do not correspond exactly to the formulae, the discrepancy being most serious for the lateral histogram (LAT) technique. It turns out that this is because the particular image data is rather non-ideal (see Figure 2), with the lateral histograms throwing up a fair number of false alarms. For low values of r there are $\sim 6 \times 6$ candidate hole locations, but only one real hole. For larger values of r the boundary of the metal plate containing the hole disappears out of the image, and the number of false alarms drops to $\sim 3 \times 3$: this effect explains why the experimental (LAT) plots do not lie on a single monotonically increasing curve. The other discrepancies between the numerical results and the formulae seem to arise from variable pipelining and caching effects in the computer CPU: as they are specific to the computer, they will not be considered in detail here.

A final experiment with the lateral histogram method involved making simple tests to eliminate most of the false alarms, giving method LAT'. Thus the final template matching procedure for eliminating ambiguities was turned into a 2-stage template matching procedure - the first stage removing most of the false alarms and the second stage making a final check for the presence of a hole. This proved highly effective, and the resulting

execution times matched the expected variation closely (see Figure 1). Note, however, that the effectiveness of such 2-stage matching procedures is bound to be somewhat data-dependent: noise, texture or other structure within or just outside a hole necessitating a first stage that performs slightly more averaging or analysis, and this could increase computational requirements again.

These results and considerations show that the effectiveness of the lateral histogram technique is limited ultimately by the actual data it is run on, hence making it difficult to draw general lessons. However, the particular set of images tried here are quite non-ideal, with plenty of noise, surface texture and image structure: yet still the technique proved effective and easily the fastest of all the algorithms tested. The formulae show how speed varies with the number of holes, and indicate that where the method is applicable it is likely to be the most efficient available technique for hole detection.

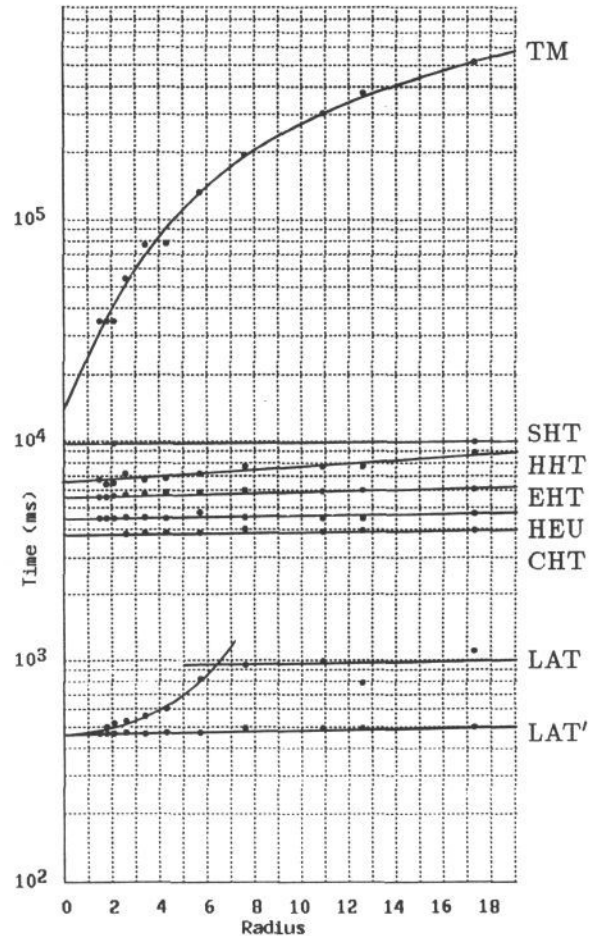


Figure 1. Graph of execution time v. radius for various hole detection algorithms

The experimental plots were obtained from algorithms coded in standard Pascal and run on a DEC MicroVax II computer. The curves are intended to show the underlying trend of the data (they are not theoretical curves): however, two separate curves are necessary to achieve this with the LAT data - see text.

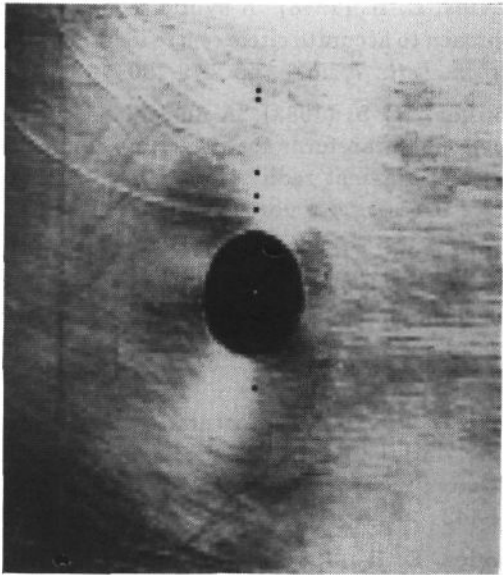
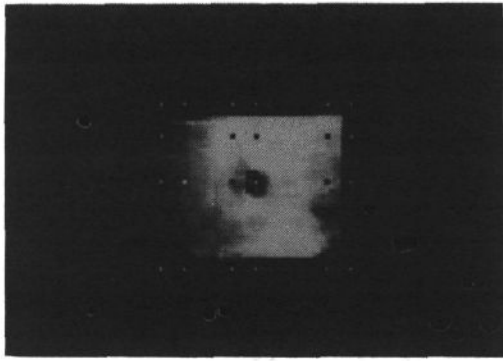


Figure 2. The images used to obtain the results of Figure 1

These figures show two of the images used to obtain the results of Figure 1, the radii having values 2.1 and 12.6 pixels. Notice how zooming in towards the metal plate eventually obscures its boundaries and reduces the numbers of false alarms. The black and white dots show the holes and false alarms obtained immediately after analysing the lateral histograms: further checks are able to eliminate all the false alarms. Note that the surface of the plate has quite a lot of damage, texture and uneven illumination, making the problem of locating holes more difficult than might at first be imagined.

APPRAISAL OF THE LATERAL HISTOGRAM METHOD

Overall, the high efficiency of the lateral histogram method lies in its 1-D matching technique, coupled with the fact that computation of the 1-D histograms takes place in just twice the time needed to visit each pixel in the image once. It is also of interest that this particular computation consists merely of a sequence of blind additions, which will be coded highly efficiently on most processors - especially if they incorporate significant pipelining.

More particularly, the lateral histogram approach incorporates several features that make it well-adapted for hole detection. As remarked in the Introduction,

holes are often small and spatial quantisation noise is often prominent. In addition, holes tend to have internal structure due to the effects of shadows; sometimes glints appear on the hole boundary or within the hole (this depends on the depth of the hole and what is at the far end, but it also depends critically on the illumination). Hence the intensity profile of a hole may have any of a variety of shapes. To offset these effects, it is helpful if hole detection algorithms incorporate a high degree of averaging of intensities as an integral part of their structure. Such a feature has a further advantage in permitting holes with fuzzy boundaries to be detected straightforwardly. Again the lateral histogram approach answers these requirements. Finally, it may sometimes be of interest to locate holes that are not exactly circular. The HT-based schemes are likely to have problems in such cases. However, the heuristic and lateral hole detection schemes will cope with this situation with little difficulty. We list in Table 2 the special points that need to be borne in mind when designing efficient hole detectors.

CONCLUDING REMARKS

At first sight hole detection may be regarded as a special case of circular object detection, hence being a process that is readily tackled by the usual HT-based procedures. However, the present work has shown that hole detection has a number of intricacies of its own that demand the use of carefully selected hole detection schemes. These intricacies are listed in Table 2, and it has been found that they are answered well by the lateral histogram technique. In addition, this technique has been found, under certain well-defined conditions, to be exceptionally efficient at the task of hole location.

It is of relevance that an overall view of the lateral histogram technique in the final form (LAT') employed here is actually a 3-stage template matching procedure. The first stage can be regarded as an efficient false-negative eliminator; the second stage can be regarded as a false positive eliminator; and the final stage as providing a final check. It is worth asking whether false negative elimination must necessarily be carried out before false positive elimination, but of course the answer is clear - that most of any image is not composed of the sought-after features, so the first stage should be devoted to eliminating as much redundancy as possible.

Finally, as suggested in the Introduction, it will often be the case that finding holes is only a first stage in locating specific types of object: if methods did not exist for inferring the presence of objects from their hole (or other) features, there would be little point in the exercise. However, several means have been devised for achieving this: here we mention merely the well-known graph matching and maximal clique type of approach, and the structural HT [16-18].

1. Holes may be small with significant spatial quantisation noise giving irregular boundaries.
2. Holes may not be exactly or even nearly circular (more precisely, their *images* may not be circular).
3. Holes may have fuzzy boundaries (i.e. no sharp edges), at the scale needed to see them.
4. Holes may have significant internal structure because of shadows, glints, etc.
5. Holes may appear in surroundings with considerable surface texture.
6. The small size of holes may preclude use of the usual edge detectors (even those with $\sim 3 \times 3$ masks).
7. The small size of holes may mitigate against use of Hough-based detectors, because parameter space will not obtain enough votes to form well-defined peaks.

Table 2. Intricacies of hole detection

Overall, these points amount to a situation of poor signal-to-noise ratios, exacerbated by the small size of the hole features. As discussed in the text, this leads to the need for significant averaging in hole detection schemes. Likewise, point 7 should be combated by using 1-D histogram detectors because these concentrate the votes (see the circle detector of [12] described briefly in the section on circle detection, and the lateral histogram technique [15] described in the section on hole detection).

Note: Although holes of radius up to ~ 20 pixels have been considered in this paper (see especially Figure 1), the above points are targeted mainly at holes of radius up to ~ 10 pixels.

Acknowledgements

We are grateful to the ACME Directorate of the SERC for grant GR/D/88526 which supported this research. In addition, one of us (SPB) is indebted to the SERC for personal support through a Research Studentship.

REFERENCES

1. Ballard, D. H. (1981) "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recogn.*, **13**, no. 2, pp. 111-122
2. Persoon, E. and Fu, K.-S. (1977) "Shape discrimination using Fourier descriptors", *IEEE Trans. Systems Man Cybern.*, **7**, no. 2, pp. 170-179
3. Duda, R. O. and Hart, P. E. (1972) "Use of the Hough transformation to detect lines and curves in pictures", *Comm. ACM*, **15**, no. 1, pp. 11-15
4. Kimme, C., Ballard, D. and Sklansky, J. (1975) "Finding circles by an array of accumulators", *Comm. ACM*, **18**, no. 2, pp. 120-122
5. Gerig, G. and Klein, F. (1986) "Fast contour identification through efficient Hough transform and simplified interpretation strategy", *Proc. 8th Int. Conf. on Pattern Recogn.*, Paris, (27-31 Oct), pp. 498-500
6. Illingworth, J. and Kittler, J. (1987) "The adaptive Hough transform", *IEEE Trans. Pattern Anal. Mach. Intell.*, **9**, no. 5, pp. 690-698
7. Davies, E. R. (1988) "A modified Hough scheme for general circle location", *Pattern Recogn. Lett.*, **7**, no. 1, pp. 37-43
8. Davies, E. R. (1988) "A hybrid sequential-parallel approach to accurate circle centre location", *Pattern Recogn. Lett.*, **7**, no. 5, pp. 279-290
9. Conker, R. S. (1988) "A dual plane variation of the Hough transform for detecting non-concentric circles of different radii", *Computer Vision Graph. Image Process.*, **43**, no. 2, pp. 115-132
10. Yuen, H. K., Princen, J., Illingworth, J. and Kittler, J. (1989) "A comparative study of Hough transform methods for circle finding", *Proc. 5th Alvey Vision Conf.*, Reading (31 Aug - 2 Sept), pp. 169-174
11. Davies, E. R. (1984) "Circularity - a new principle underlying the design of accurate edge orientation operators", *Image and Vision Comput.*, **2**, no. 3, pp. 134-142
12. Davies, E. R. (1987) "A high speed algorithm for circular object location", *Pattern Recogn. Lett.*, **6**, no. 5, pp. 323-333
13. Kelley, R. B. and Gouin, P. (1984) "Heuristic vision hole finder", *Proc. 4th Int. Conf. on Robot Vision and Sensory Controls*, London (9-11 Oct), pp. 341-350
14. Wu, Z.-Q. and Rosenfeld, A. (1983) "Filtered projections as an aid in corner detection", *Pattern Recogn.*, **16**, no. 1, pp. 31-38
15. Davies, E. R. (1987) "Lateral histograms for efficient object location: speed versus ambiguity", *Pattern Recogn. Lett.*, **6**, no. 3, pp. 189-198
16. Bolles, R. C. (1979) "Robust feature matching via maximal cliques", *SPIE*, **182**, *Imaging Applications for Automated Industrial Inspection and Assembly*, pp. 140-149
17. Bolles, R. C. and Cain, R. A. (1982) "Recognizing and locating partially visible objects: the local-feature-focus method", *Int. J. Robotics Res.*, **1**, no. 3, pp. 57-82
18. Davies, E. R. (1988) "An alternative to graph matching for locating objects from their salient features", *Proc. 4th Alvey Vision Conf.*, Manchester (31 Aug - 2 Sept), pp. 281-286