Comparative study of Hough Transform methods for circle finding

H K Yuen, J Princen, J Illingworth and J Kittler

A variety of circle detection methods which are based on variations of the Hough Transform are investigated. The five methods considered are the standard Hough Transform, the Fast Hough Transform of Li et al.[1], *a two stage Hough method, and two space saving approaches based on the method devised by Gerig and Klein*[2]. *The performance of each of the methods has been compared on synthetic imagery and real images from a metallurgical application. Figures and comments are presented concerning the accuracy, reliability, computational efficiency and storage requirements of each of the methods.*

Keywords: Hough Transform methods, circle detection, performance

In recent years, several methods of circle finding based on the Hough Transform (HT) have been proposed[2,3], as well as some general techniques for fast implementation of the HT[1,4,5]. Invariably these methods claim to improve efficiency, storage or reliability, though in most cases the comparison made with other techniques is superficial. This paper considers a number of Hough-based, circle finding algorithms, and examines their properties in detail. The study is not exhaustive, but it covers a number of proposed approaches including the Standard HT, the method of Gerig and Klein[2] (with and without edge direction information), the 2-stage method (discussed in References 3 and 4), and a method which uses the Fast Hough Transform[1]. In some cases these algorithms are implemented in their original form, though in others it has been necessary to make some extensions and improvements. The study is experimental, and presents results illustrating computational efficiency, storage requirements, reliability and accuracy of the methods for both real and synthetic images. A general conclusion of our work is that more complicated variations of the HT method do not necessarily outperform straightforward approaches.

The circle finding problem and the basic idea underlying the HT are introduced. This is followed by a brief description of each of the five HT-based methods considered in our study. The experimental results for each method are given, and several points arising from the study are discussed. Brief conclusions of the work are then presented.

Department of Electronic and Electrical Engineering, University of Surrey, Guildford GU2 5XH, UK

CIRCLE FINDING USING THE HT

The HT method of shape analysis uses a constraint equation relating points in a feature space to possible parameter values of the searched for shape. For each feature point, invariably edge points, votes are accumulated for all parameter combinations which satisfy the constraint. The votes are collected in an array of counters which is called the accumulator array. The accumulator array is a discrete representation of the continuous multidimensional space which spans all feasible parameter values. Edge points from a single instance of a shape vote coherently into the accumulator counter which is closest to the parameters of the shape. At the end of the voting or accumulation process those array elements containing large numbers of votes indicate strong evidence for the presence of the shape with corresponding parameters. Shapes are detected by using some method to identify peaks in the accumulator array.

The use of the HT to detect circles was first outlined by Duda and Hart[5]. If a circle is parameterized by its centre coordinates (a, b) and its radius r, then these are related to the position of edge points (x, y), which form the circle via the constraint:

$$(x-a)^2+(y-b)^2=r^2 \tag{1}$$

This equation also indicates that any given edge point (x_i, y_i) could be a point on any circle whose parameters lie on the surface of a right circular cone in the (a, b, r) parameter space (see Figure 1). If the cones corresponding to many edge points intersect at a single point, then all the image points lie on the circle defined by those three parameters. Kimme *et al.*[6] give probably

0262-8856/90/010071-07

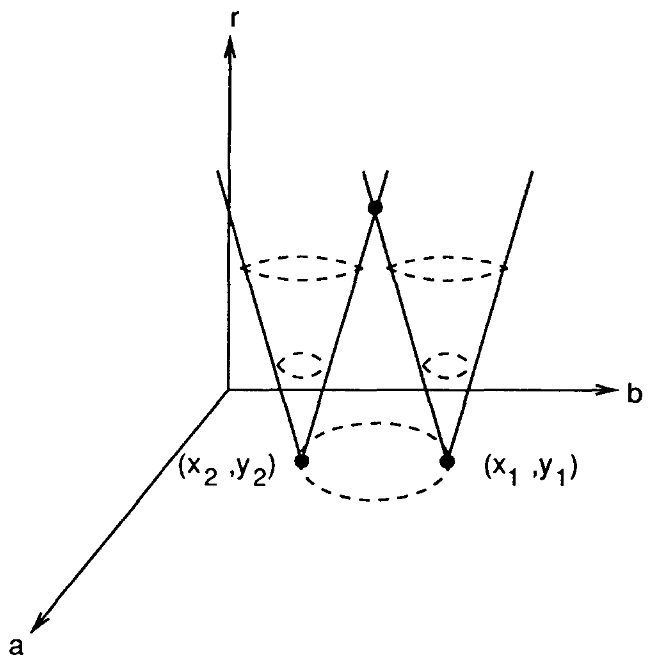


Figure 1. Edge points (x, y) *generate votes on a conic surface in 3D* (a, b, r) *parameter space. Points where three or more cones intersect define parameters of circles*

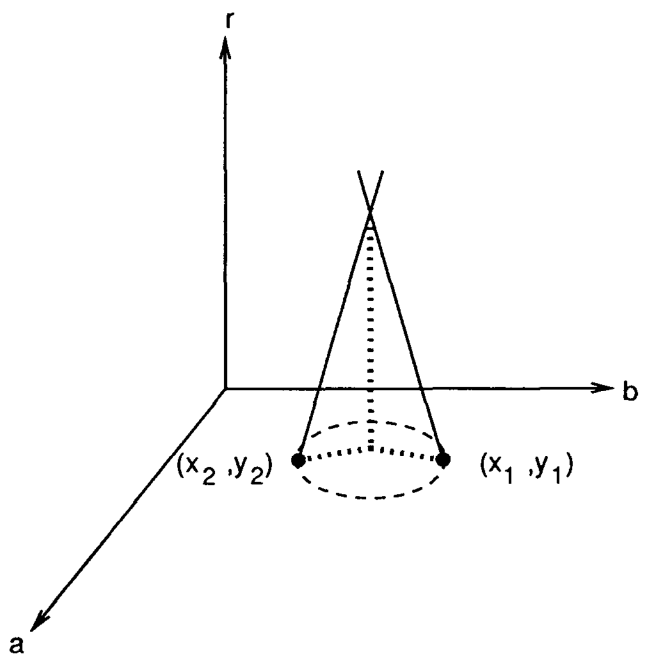


Figure 2. Using edge direction restricts edge points to voting along a single line of the conic surface

the first known application of the HT to detecting circles in real images. In this work, the direction of the gradient at each edge point is used as an additional piece of data which can further constrain the possible parameter values consistent with a given edge point. The centre of a circle must lie on the line passing through the edge point along the maximum grey level gradient direction. As a result, instead of incrementing the whole circular cone, only segments of the cone need be incremented.

An important part of the complete HT process is peak detection. A number of methods have been used in the past to identify peaks in the accumulator array, including local maximum detection and thresholding. However, many of these methods are unsatisfactory[7] because in most cases peaks in the accumulator are rather broad, and are not generally convex. Hence, identifying a single peak location is difficult. An extremely useful technique which has been adopted in all the work of this paper is the peak finding method proposed by Gerig and Klein[2]. It consists of taking each edge point and identifying the maximum accumulator bin which was voted for by the point. The edge point is then labelled with the parameter values of this bin. Following labelling of all edge points the accumulator array is reinitialized, and the HT is reaccumulated with each edge point voting only for the single parameter values that label it. The resultant one-to-one mapping of edge point to parameter point produces a very simple accumulator structure in which peaks are separated and typically consist of only a few neighbouring bins with high counts. A low threshold can then be used along with a simple clustering technique to group neighbouring bins into a single solution.

Standard HT

The Standard Hough Transform (SHT) in this study follows the basic ideas outlined above. A 3D accumulator array is employed, and edge direction information is used to limit voting to a section of the cone. For perfect edge direction accuracy only a single line of votes is incremented (see Figure 2). The centre of the circle should lie along the direction of maximum grey level gradient for points which form the circle boundary. Planes of (a, b) values, at constant values of r, are congruent to the image space, and therefore the bins which have to be incremented in the 3D (a, b, r) accumulator can be efficiently determined by using the edge point position and direction to define an (a, b) value and constraint equation (1) to calculate the remaining parameter r. The choice of quantization along each of the parameter axes is an important design choice which reflects both the image space quantization and the required accuracy of parameter determination. However, a natural choice for this study is to have each accumulator bin represent a 1 pixel unit for each parameter. Therefore, to search for a circle whose centre is within an image of size N, and whose radius is less than or equal to N, would require a 3D accumulator of size N^3.

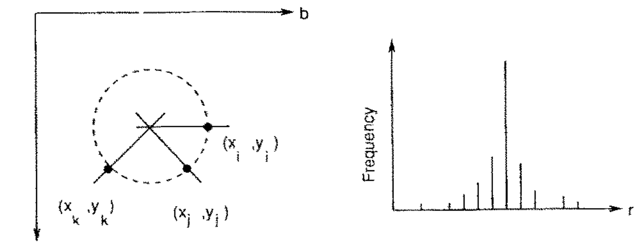
Gerig Hough method

One of the problems with the SHT method is the large storage space required if the range of circle radii is large. To overcome this, Gerig and Klein[2] suggest reordering the HT calculation to replace the 3D accumulator of size N^3 by three 2D arrays of size N^2. This method is referred to here as the Gerig Hough Transform (GHT) method. In the SHT, each edge point votes for all its parameter cells in 3D space. Peak finding is performed only when the transform has been accumulated for the full parameter space. The GHT method performs the full HT as a series of HTs in which, at each stage, there is only a single value of radius. At each stage, a 2D array acts as working space for transform accumulation and local peak finding. Peaks are characterized by their position, their size and the radius for which the transform is accumulated. The peak size and radius information can be recorded in the appropriate bins of two 2D arrays which are congruent to the working array (see Figure 3). The working array is then reinitialized and used to calculate the transform for another value of radius r. This process is repeated for all possible distinct radius values. At the end of the process the two 2D arrays contain information about location, size and radius of transform peaks. A deficiency of the GHT method is that because the details of peaks are recorded at corresponding spatial positions in the two 2D arrays then values are overwritten if two circles share a common position. Therefore, in this simplest implementation it is not possible to store all details of concentric circles.

The GHT method as described in Reference 2 does not use edge direction, and therefore a complete circle is incremented for each edge point at every value of radius. However, the efficiency of the GHT method can be improved by incorporating edge direction information. This new method is referred to here as the Gerig Hough Transform with gradient (GHTG). If edge direction is used, then only the single bin on the circular parameter locus which lies in the direction of the gradient of an edge point is incremented.

Two stage Hough Transform

If edge direction information is available, then one way to reduce the storage and computational demands of circle finding is to decompose the problem into two stages consisting of a 2D HT to find circle centres, followed by a 1D HT to determine radii (this approach has been used in References 3 and 4). This method is referred to here as the 2–1 Hough Transform (21HT), and is illustrated in Figure 4. Since the centre of a circle must lie along the gradient direction of each edge point on the circle, then the common intersection point of these gradients identifies the centre of the circle. A 2D array is used to accumulate the centre finding transform, and candidate centre parameters are identified by local peak detection. This stage of the method can be viewed as an integration along the radius axis of all values of the HT at a single value of (a, b). The second



a **b**

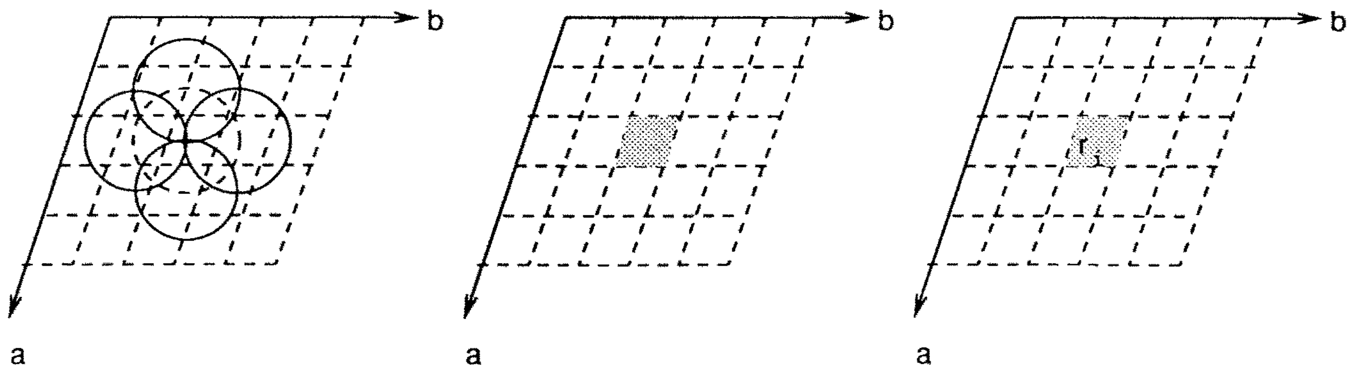
Figure 4. 21HT decomposes the circle detection problem into two stages. In (a) a transform is accumulated to find centre coordinates. In (b) a radius histogram is constructed for each candidate centre derived in (a)

stage of the method uses the centre parameters of the first stage together with constraint equation (1) to construct and analyse histograms of possible radius values. A radius histogram is made for each candidate centre, and peaks in the radius histogram indicate evidence for circles.

A potential problem with the 21HT method is that the 2D centre finding accumulator is likely to be more difficult to analyse than a 3D space, as it is an integral projection of that space. In addition, any false peaks or inaccuracy in parameter estimation in the first stage wili affect the performance of the second stage.

Fast Hough Transform

Li *et al.*[1] suggest a method called the Fast Hough Transform (FHT), which uses a multidimensional quadtree structure to simultaneously accumulate and detect peaks in the HT. In their formulation shapes are parameterized so that they map into hyperplanes in parameter space. They suggest that the FHT requires less storage than the SHT. The algorithm can be seen as an example of hierarchic search. The original parameter volume is considered as a single hypercube. If the number of image points voting for parameters in a hypercube is higher than a threshold value, then the corresponding region of parameter space is assumed to include a parameter peak and the volume is investigated in greater detail by decomposing the hypercube into subhypercubes. This iterative process of accumula-



a **b** **c**

Figure 3. GHT method employs three 2D arrays. Array (a) acts as a working space for transform accumulation and analysis for a single radius value; arrays (b, c) record information concerning the position, size and radius value of candidate peaks

tion, analysis and division continues until each hypercube volume reaches a desired limit of parameter resolution, or a hypercube falls below the threshold which defines a peak structure. There are several possible control policies for the FHT tree search algorithm[1]. In this work, the simplest depth first policy has been adopted.

The original FHT was based on the use of a parameterization which mapped shapes into planar voting surfaces in parameter space: the hyperplane formulation. Such mappings are natural when considering the detection of lines or planes using gradient-intercept parameters, and are particularly advantageous as efficient methods can be devised for incremental testing of intersection between parameter space volumes and these hyperplanes[1]. Circle finding can be developed within the hyperplane formulation, but our experience[8] indicates that the resultant non-linearity along parameter axes causes severe problems. To avoid this problem, and increase the efficiency of the FHT, we use a different formulation which incorporates edge direction information. The modified approach is referred to here as the modified Fast Hough Transform (MFHT). The method uses the fact that an edge point with position (x_i, y_i) and gradient direction ϕ_i yields a locus in (a, b, r) space which consists of orthogonal straight lines pointing outward from the point $(x_i, y_i, 0)$. The orthogonal distance between these lines and the centre of each hypercube provides the basis for an easy, approximate test of whether the line intersects the hypercube. If the perpendicular distance is less than the diagonal distance across a hypercube, then the line intersects the hypercube in the radial sense as defined by Li *et al.*[1] and the hypercube receives a vote. Unlike the original FHT, this intersection test does not use a simple incremental updating formula.

An important aspect of the FHT algorithm is the choice of vote threshold which defines when a peak structure occurs and division continues. Too low a threshold will produce an inefficient search of the parameter space, while too high a threshold may lead to poor detection efficiency for small circles. For the work presented here we have developed an adaptive scheme. At first a high threshold is chosen, and large circles are searched for over a narrow range of radii. The edge points constituting these circles are then removed from consideration, and the MFHT is applied to the remaining points with a low threshold and small range of radii to find small circles.

EXPERIMENTAL RESULTS

The images considered in this study are of a class which commonly occur in medical or metallurgical applications. Such applications often involve liquid or metal droplets which project to approximately circular shapes in images. Typically, such images contain many overlapping circles of a wide range of sizes. An important problem concerns counting the number of distinct circles and determining distributions of quantities such as droplet size and/or spatial distribution. Many shape detection methods have great difficulty coping with the problem of occlusion, and the fragmentary nature of outlines extracted by methods such as edge finding. However, the HT is particularly well suited to these aspects of the imagery.

There are many criteria which can be considered in any comparison of algorithms, but in our study the most important points relate to the accuracy, reliability of detection, computational complexity and storage. In the following subsections, these are discussed in relation to experiments with both synthetic and real imagery. Synthetic images are useful for assessing the accuracy of parameter determination, but real imagery often poses unexpected problems, therefore it is essential to test methods on realistic data. Figures 5 and 6 show edge data extracted from a typical synthetic and a typical real image, respectively. The edge points are identified using a method based on the Canny edge detector[9], followed by thresholding and binary edge thinning. To simulate noise effects the edge direction of

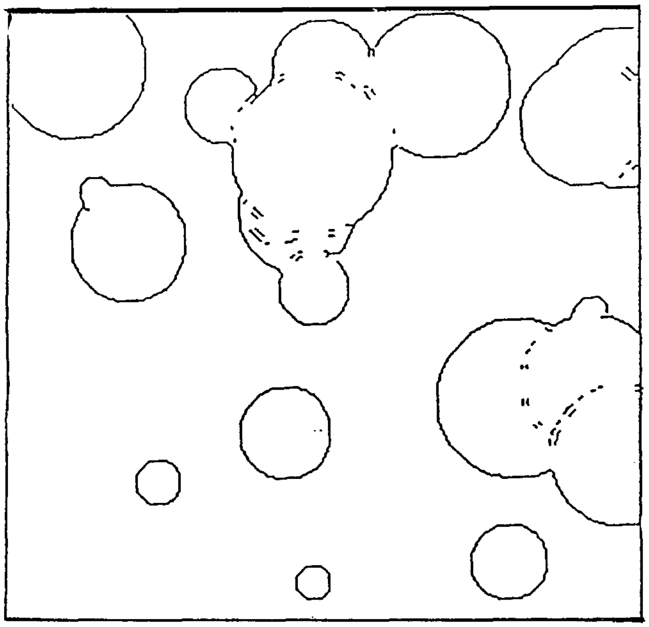


Figure 5. Edge data extracted from a synthetic image

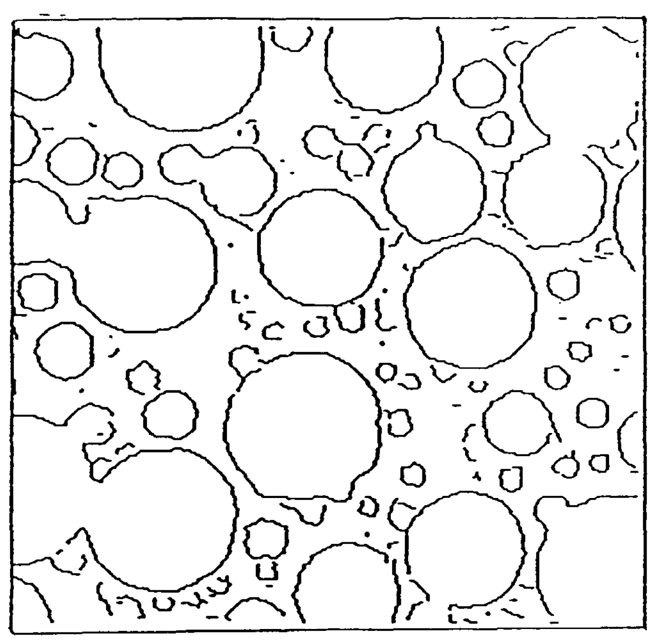


Figure 6. Edge data extracted from a real image

each edge pixel in the synthetic images was smeared by a value randomly chosen from a uniform distribution in the range $\pm 5°$. The synthetic image shown contains 1397 edge points and 19 circles generated in random positions with radii less than 30 pixels. The real data comes from a metallurgical experiment, and the image shown has 3750 edge points and approximately 76 circles of a wide range of sizes. All the algorithms have been implemented in Pascal on a μVax-2 computer.

Accuracy

The accuracy of each method is determined using synthetic images by measuring the absolute errors between estimated (a, b, r) values and the known parameters which were used to synthesize the images. Figure 7 shows these error measures for each of the methods averaged over all images. It is seen that all methods perform well, and on the average achieve sub-pixel accuracy for parameter estimation. In most cases the error was either 0 or 1 pixel.

Reliability

In this study the term reliability is used to encompass two effects: the probability of finding true circles, and the chance of detecting false positives, i.e. circle detection resulting from combining evidence from unrelated edge fragments. Figure 8 shows the number of circles correctly detected and the number of false detections as a fraction of the true number of circles in the image. Figure 8a shows the fraction of true circles found in synthetic images. All methods yield numbers greater than 0.8. Figure 8b shows the fraction of true circles for a typical real image. It can be seen that the number of circles correctly identified falls, quite dramatically in many cases. However, there is reasonable agreement concerning the relative ranking of each of the methods: the GHTG method is first followed by the SHT and MFHT methods. The GHT method is the worst method. The 21HT performs very well on the synthetic images, but deteriorates for complicated real images. This is related to small circles being lost as the accumulator space in the first stage is a projection of votes over all radius values. Small peaks corresponding to small circles are lost at the first stage of the 21HT method. Figure 9 shows a reconstruction of the real image of Figure 6 based on the parameters of circles detected by the GHTG method. It can be seen that the reconstruction is extremely good even for very small circles. Figure 8c shows the fraction of false positives found in real imagery. The values generally correspond to the false detection of small circles. In the case of synthetic images, false positives were almost zero in all cases.

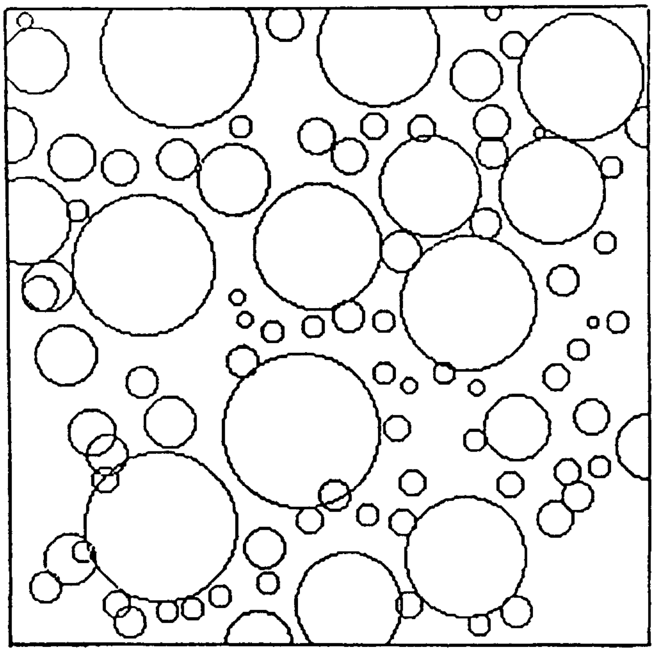


Figure 9. Reconstruction of the circles in the real image using parameters derived from the GHTG method

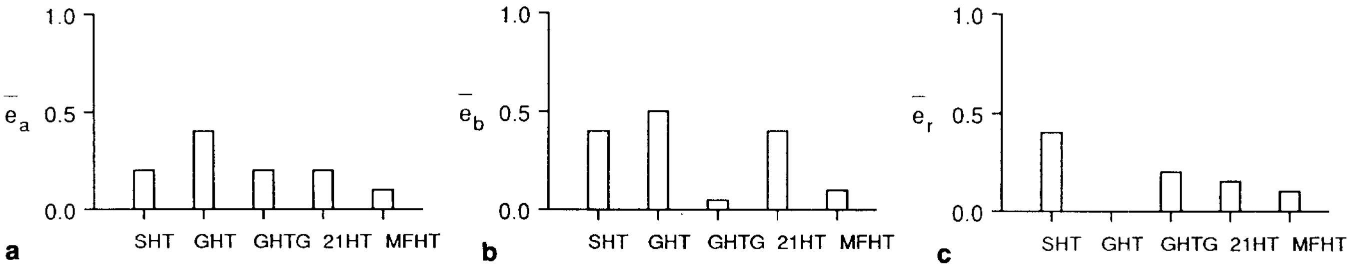


Figure 7. Mean of the absolute error for estimates of parameters (a, b, r) *as a function of method*

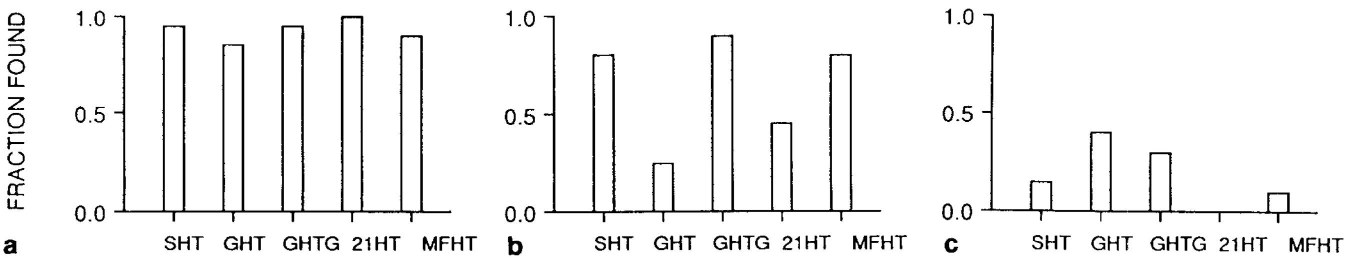


Figure 8. Detection rates as a fraction of the true number of circles in an image. (a) Correctly identified circles found in synthetic images; (b) correctly identified circles found in real images; (c) false positive rate for real images

Storage and computational efficiency

The storage and computation times of each of the methods is illustrated in Figure 10. All methods, except for the MFHT, used an accumulator size which allowed each parameter to be determined to an accuracy of 1 pixel. For a 256×256 image the centre coordinates could be anywhere in the image, but the radius of circles was assumed to be less than 35 pixels, i.e. the standard HT required storage of $256 \times 256 \times 35$ integer storage locations. In the MFHT the termination criterion was set to correspond to a parameter accuracy of 1 pixel. The storage used by the MFHT method consists of dynamically allocated space to store the nodes and attributes of nodes in the tree search. The storage per node can be quite large, as Li *et al.*[1] suggest maintaining flags which indicate which edge points contribute to each node.

The computational efficiency of the methods was measured by the CPU times of the algorithms. Although this is not a measure of general significance, such as would be provided by a complexity analysis, its use can be justified in a coarse sense because all the algorithms, except the MGHT, have much in common. The numbers shown in Figure 10 are CPU time divided by the number of input edge pixels, averaged over all images. For most of the algorithms the major time consumption is split equally between transform accumulation and the Gerig and Klein peak finding algorithm. However, in the 21HT method the radius histogram construction and analysis in the second stage becomes a significant time factor if there are many candidate centres found in the first stage. In the case of the MFHT the range of values of CPU time and storage is much greater than for the other methods, as its efficiency depends on the spatial complexity of the image. As the complexity of the image increases, the computation and storage demands of the MFHT increase non-linearly. This means that it is difficult to accurately predict the storage requirements and run time of the algorithm on images, and in some cases these can be prohibitively large.

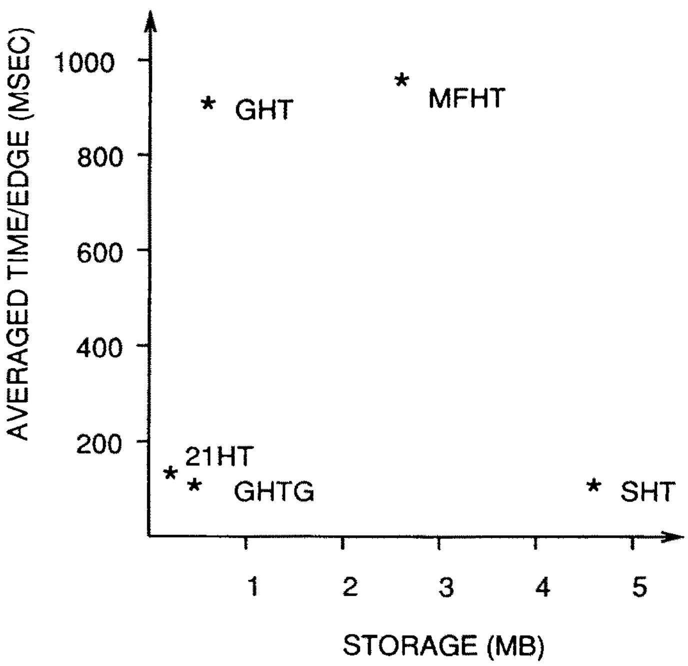


Figure 10. CPU time per edge pixel versus storage in MB for each method

Figure 10 shows a clear division between the set of proposed algorithms. The SHT requires a lot of storage while the GHT takes a long time. Both the 21HT and the GHTG provide a good compromise between storage requirement and computation time. The point shown for the MFHT is representative of its performance, but the range of values for the MFHT is much greater than the other methods as the number of nodes explored (i.e. storage and computation time) is highly dependent on the spatial complexity of the input image. The comparison of the GHT and GHTG forcefully illustrates the computational advantage of the use of edge direction information.

DISCUSSION

The experiments featured above provide a coarse picture of the relative performances of each of the five methods studied. In this section, several other points derived from these, and additional experiments are discussed and explained.

It has been shown that several methods benefit from the use of edge direction information as this permits a restriction on the number of accumulator bins which are incremented by each image point. However, the number of bins which have to be addressed is a function of the accuracy with which edge direction is determined. If edge direction is poorly measured, then several accumulator bins should be incremented. Figure 11 illustrates the region of parameter space which would be incremented in the first stage of the 21HT if the edge direction is measured with an angular accuracy of $\pm\Delta\phi$. In our studies we have experimented with different approximate incrementation strategies for different edge angle accuracy assumptions on the synthetic data. However, these sophisticated incrementation strategies generally yield only a slight gain in accuracy and reliability. These benefits have to be weighed against the requirement to reliably know

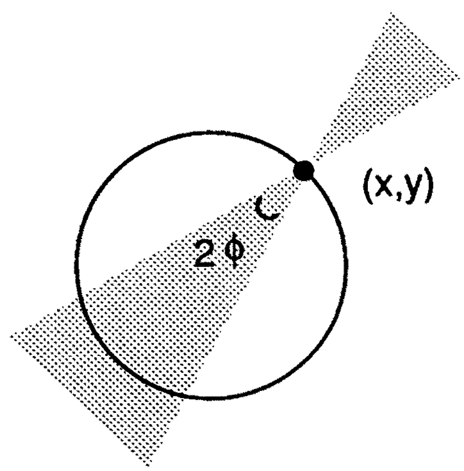


Figure 11. For the 21HT method the shaded patch indicates the region of parameter space which is incremented if edge direction is measured to an angular accuracy of $\pm\Delta\phi$

the edge direction accuracy of real data and the extra computation required in these methods.

In the case of the 21HT it has been found that inaccuracy of centre estimates determined in the first stage can cause problems in the second stage histogram. In particular, it can be shown using a simple model[7] that if the magnitude of the error in centre coordinates exceeds the radius histogram cell width, then points from a single complete circle will yield two peaks in the radius histogram. These peaks are symmetrically located about the true radius, and the distance between them increases with the centre error. This property means that it may be difficult to distinguish between peaks from two nearly concentric circles and two peaks caused by errors in the centre estimates.

In several experiments the MFHT method required very large amount of resources. This problem relates to the large number of 'phantom' solutions which the method investigates if images are complicated. It can be explained by considering the relationship between template matching and the Hough Transform[10]. Large parameter cells which typify the early stages of hierarchic search correspond to large area image templates which have little power to discriminate against extraneous image detail. Hence, hierarchic methods are often inefficient in complicated images. The 'thin-tree' observation cited by Li *et al.*[1] does not hold for complicated images.

This experimental study is to be seen as a systematic attempt to compare several methods on a small number of images which are representative of a class of interest. It cannot claim to be a comprehensive study. There are several detailed questions which make the design of a comprehensive study difficult. It is not always easy to choose a standard implementation of a particular method. There are many methods and each may include a great number of parameters whose effects have to be investigated. Also, there is a large variety of image data which is difficult to characterize and simulate realistically. All of these factors mean that comparative studies are always difficult and long exercises.

A final remark is that this study has only considered implementation of the algorithms on a standard sequential computer. An often quoted feature of the HT is that it has a high potential for parallel implementation. These factors could have a significant impact on our conclusions regarding computational efficiency of the various methods.

CONCLUSIONS

Several conclusions have emerged from this limited comparative study. Of the methods studied, the GHTG method emerges as the best overall method. It requires modest storage and is reasonably fast, accurate and reliable. Its only drawback is its inability to detect concentric circles. The main deficiency of the SHT is that it requires a lot of storage, and this grows as the range of radii covered increases. The 21HT performs reasonably well and has modest storage requirements. However, some circles, especially small ones, may be lost because of accumulator interpretation problems. The GHT and MFHT are found to be relatively poor methods in complicated images of the type that we have studied. The GHT is unreliable and inefficient because it does not use edge direction information. The storage and computation demands of the MFHT are comparitively large, and dependent on details of the image structure.

ACKNOWLEDGEMENTS

This work was carried out as part of Alvey contract MMI/078, a collaborative project involving the University of Surrey, Heriot-Watt University, and Computer Recognition Systems of Wokingham, UK.

REFERENCES

1 **Li, H, Lavin, M A and LeMaster, R J** 'Fast Hough Transform: a hierarchical approach' *Comput. Vision, Graphics & Image Process.* Vol 36 (1986) pp 139–161

2 **Gerig, G and Klein, F** 'Fast contour identification through efficient Hough Transform and simplified interpretation strategy' *8th Int. Joint Conf. Pattern Recog.* Paris, France (1986) pp 498–500

3 **Davies, E R** 'A modified Hough scheme for general circle location' *Patt. Recog. Lett.* Vol 7 No 1 (1988) pp 37–44

4 **Illingworth, J and Kittler, J** 'The adaptive Hough Transform' *IEEE Trans. PAMI* Vol 9 No 5 (1987) pp 690–697

5 **Duda, R O and Hart, P E** 'Use of the Hough transformation to detect lines and curves in pictures' *Commun. ACM* Vol 15 No 1 (1972) pp 11–15

6 **Kimme, C, Ballard, D and Sklansky, J** 'Finding circles by an array of accumulators' *Commun. ACM* Vol 18 No 2 (1975) pp 120–122

7 **Princen, J, Yuen, H K, Illingworth, J and Kittler, J** *A comparative study of Hough Transform algorithms: Part 1 – line detection methods* Internal Report, Department of Electronic and Electrical Engineering, University of Surrey, UK (1989)

8 **Illingworth, J, Kittler, J and Princen, J** 'Shape detection using the Adaptive Hough Transorm' *Proc. NATO Advanced Res. Workshop on Real-Time Object & Environment Measurement & Classification* Maratea, Italy (September 1987)

9 **Canny, J** 'A computational approach to edge detection' *IEEE Trans. PAMI* Vol 8 No 6 (1986) pp 679–698

10 **Princen, J, Illingworth, J and Kittler, J** 'Templates and the Hough Transform' *Proc. NATO Advanced Res. Workshop on Active Perception & Robot Vision* Maretea, Italy (July 1989)