A Dynamic Combinatorial Hough transform for Straight Lines and Circles.

V. F. Leavers**, D. Ben-Tzvi*,
and M. B. Sandler*

A new algorithm for the Hough transform is presented. It uses information available in the distribution of image points to calculate the parameters associated with combinations of the minimum number of points necessary to define an instance of the shape under detection. The method requires only one dimensional accumulation of evidence to determine the parameters associated with a given shape. Using the algorithm, the Hough transform of sparse images is more efficiently calculated. Dense images may be segmented and similarly processed. The method also provides a feedback mechanism between image and transform space whereby contiguity of feature points and endpoints of curves may be determined.

The Hough transform[1], [2] is a powerful tool in shape analysis. It is used to extract global features from shapes and gives good results even in the presence of noise or occlusion. While the theoretical potential of parametric transform methods has been demonstrated they have made little impact on large scale industrial applications because of supposed excessive storage requirements and computational complexity[3] The development of fast, efficient implementations of parametric transformation methods of shape detection has accordingly received much attention in the recent literature [4], [5], [6], [3], [7], [8]. An up-to-date and comprehensive review of the use of the Hough transform is given by Illingworth and Kittler[9].

Previous suggested approaches may be divided into two categories. The first seeks to reduce the computational load by using evidence from the image to reduce the generation of evidence groupings of points. Such methods are:

1. Using edge direction information as an indication of the parameter range to be considered[2], [10]. Using this approach requires an accurate estimation of the θ range from the edge direction data. A large mask needs to be used in the edge detection stage, which merely shifts the computational burden.
2. For straight line detection, a local operator may be applied which exploits the constraint that at least two points are required to define a line[7]. This algorithm does succeed in reducing the memory allocation requirements. However, because of the complexity of its implementation it does not offer a significant reduction in computational load and the net result of using a local operator is a reduction in resolution.
3. A further method for straight line detection preprocesses the edge data to form a list whose elements are ordered on contiguity and form the equivalent of a chain coded representation of the edge image data[8]. A variable sized sliding window is then used in conjunction with this list to generate point pairs which are mapped into the parameter space. Again the method simply shifts the computational burden to the preprocessing stage and offers no reduction in memory allocation. A significant disadvantage of the method is that it inherits all of the weaknesses inherent in the chain coding process.

The second class of methods involves absolute or iterative reductions in resolution of either the transform or the image space.

4. Segmenting the image. In [11] segmentation of the image is considered. Segments of the same line contribute separately to the same entry in the transform space. But still all possible values of the dependent parameter, are calculated. Segmenting the image in this case is equivalent to a reduction in resolution, which may not be acceptable.
5. Using an adaptive method[4], [5], [6], in which a coarse resolution in transform space is initially made, advancing to a finer resolution around candidate peaks. It is an iterative process, with the results of each stage depending on results obtained in the previous one. This method may work well for simple images, but for images with multiple instances of curves, a coarse search in the transform space may not detect all maxima, thus failing to detect them in the iterations that follow.

Evaluation of the information generated in the transform space may present difficulties. Problems associated with the detection of maxima in the transform space may be partially solved by the use of matched filtering techniques to detect those maxima[12], [13]. However, a major shortcoming

** King's College London, Department of Physics
* King's College London, Department of Electrical Engineering

AVC 1989 doi:10.5244/C.3.28

of the technique remains in that all information about feature points contributing to a maxima in the transform space is lost in the transformation process. It is therefore not possible to determine either contiguity of feature points nor end points of curves. Gerig[14] attempts to solve these problems, in the case of circle detection, using a technique which maps information from the parameter space back to the image space. In this way each image point has associated with it a most probable parametrisation. A second transformation is performed where, for each image point, only the cell in parameter space associated with the most probable parametrisation of that image point is incremented. The technique works well in that it is a reliable strategy for interpreting the accumulator space. It is however still computationally complex and offers no reduction in memory allocation.

The proposed method seeks both to cut significantly the computational burden involved in the implementation of the transform, to provide an efficient feedback mechanism linking the accumulated boundary point evidence and the contributing boundary point data and to facilitate the detection of maxima.

1 . Combinatorial Hough transform

An expression for the Generalized Hough Transform, GHT, may be written in the form suggested by Deans[15]

$$f(\xi, p) = \iint_D F(x, y)\delta(p - C(x, y; \xi))\, dx\, dy \quad (1)$$

where $F(x, y)$ is an arbitrary generalized function[16] defined on the xy plane D. The argument of the delta function defines some family of curves in the xy plane parametrized by the scalar p and the components $\xi_1, \xi_2, \ldots \xi_n$ of the vector ξ. If, $F(x, y)$, represents a binary image the integral of equation 1 will have a value of 1 when the argument of the delta function evaluates to zero. The evaluation of the argument, in its discrete form,

$$p_j = C(x_i, y_i; \xi_j)$$

is used to calculate the standard GHT. The i, j subscripts refer to ordered pairs in the image and the transform space respectively. For every point, (x_i, y_i), of the image, i is fixed and the values p_j are calculated using combinations of stepwise increments of the components of ξ_j. Each point, (p_j, ξ_j), in the transform space will refer to a possible curve in image space which passes through the point (x_i, y_i). The SHT therefore provides a great redundancy of information concerning the image. This is because each image point is treated independently.

The present technique proposes that each image point be tested for the most probable, as opposed to all possible, membership of curves. If, when the image is scanned for candidate feature points, a list of those feature points is maintained, then probable membership of curves may be tested by calculating the parameters associated with combinations of the minimum number of points necessary to define an instance of the shape under detection[17].

It is clear that where n parameters are associated with the shape under detection then a minimum of n points are required to test the membership of a curve of any given $n-1$ points with the image point under consideration. For an image containing m points, to test each point in this way would require $C_m^n = \frac{m!}{(m-n)!n!}$ computation cycles. If the number of feature points, m, is large enough, the number of computations required far exceeds those required when using a standard GHT algorithm. This problem may be resolved by applying the technique *dynamically* in a manner appropriate to the curve under detection.

1.1 Dynamic Combinatorial Hough Transform for Straight line detection

The simplest possible combination of image points is that of two point colinearities. For two such co-linear points, (x_1, y_1), (x_2, y_2), the equation of the line joining them is given by:

$$p = x\cos\theta - y\sin\theta \quad (2)$$

where:

$$\theta = \tan^{-1}\left(-\frac{(x_1 - x_2)}{(y_1 - y_2)}\right) \quad (3)$$

The Dynamic Combinatorial Hough Transform, DCHT proposes that the value of θ be calculated for each two-point colinearity involving the first point, (x_1, y_1), and the remaining points, (x_i, y_i), on the list of image points. These values are then accumulated in a θ histogram. If m of the points in the list are co-linear with the first point, it results in a peak of value m at the θ value of this line in the θ histogram. After such a peak has been detected, the value of r for this line may be calculated using the (x, y) coordinates of the first point. It can be reasonably assumed that the m co-linear points do not reside on another line. Thus, they may be removed from the list, and the next pass of the algorithm handles a shortened list. In each of the following passes the first point in the shorted list is combined with all other points, and a new θ histogram is generated. Points contributing to peaks are successively deleted from the list. Each set of points may be tested for contiguity and the end points of line segments determined. The procedure continues till all points in the list are deleted.

1.2 The Dynamic Combinatorial Hough Transform for Circle detection

In general, the Hough technique requires that a new parametric transform space be used with respect to each new shape, the dimensionality of the transform space being a function of the number of parameters associated with a particular shape. Using the Dynamic Combinatorial Hough Transform this massive memory requirement may be avoided and the computational load significantly decreased.

Circles may be successively detected by calculating the parameters associated with the circle which passes through combinations of three non-colinear points such that the first image point, (x_1, y_1), in each combination is fixed. The calculated parameters are accumulated in one dimensional histograms. Peaks in the histograms will indicate the parameters associated with the most probable instance of the circle in image space of which the point, (x_1, y_1), is a member.

To calculate the parameters of the circle, the present approach uses the following property, that the perpendicular bisector of a chord passes through the center of the circle. For a triplet of edge points, (x_1, y_1), (x_2, y_2), (x_3, y_3), on a circle, the perpendicular bisectors of the two chords generated by these three points will intersect at the center of the circle. For example, the equation of the perpendicular bisector of the chord formed by the colinearity of the points (x_1, y_1) and (x_2, y_2) may be expressed in the form:

$$y - y_{b_1} = m_1(x - x_{b_1}) \tag{4}$$

where $x_{b_1} = \left(\frac{x_1+x_2}{2}\right)$ and $y_{b_1} = \left(\frac{y_1+y_2}{2}\right)$ are the co-ordinates of the foot of the perpendicular and its orientation is given by:

$$m_1 = \tan^{-1}\left(\frac{x_1 - x_2}{y_1 - y_2}\right) \tag{5}$$

The center co-ordinates, (x_c, y_c), and the radius, r, of the circle on which all three points reside are given by:

$$\begin{aligned} x_c &= \frac{m_2 x_{b_2} - m_1 x_{b_1} + (y_{b_1} - y_{b_2})}{m_1 - m_2} \\ y_c &= y_{b_1} + m_1(x_c - x_{b_1}) \\ r &= \left((x - x_c)^2 - (y - y_c)^2\right)^{\frac{1}{2}} \end{aligned} \tag{6}$$

The calculated center co-ordinates and radius are accumulated in three one dimensional histograms. Maximum values in the three histograms indicate the parameters of the circle of which the point (x_1, y_1) is most probably a member. Any points residing on this circle may then be removed and the next pass of the algorithm handles a shortened list of feature points.

2 . Implementation and Illustrations

Fig 1(a) shows a 256 × 256 test image of an hexagonal nut. A Laplacian edge detector is applied to the test image. Fig 1(b) shows the resulting binarized edge image. To reduce the computational load, the edge image is segmented into sixteen sub-images, each processed independently.

Following the method outlined in section 1.1, the co-ordinates of the edge points in a segment are listed in the order of their appearance. A point on the list is then fixed. The fixed point is paired with all other points in that segment provided that the distances between the points are large enough to minimize inaccuracies in grid representation.

A θ histogram is generated using equation (3) and the peak extracted. Using equation (2) p is calculated for the extracted value of θ. Points which contribute to this value of (p, θ) are removed from the list and the next pass of the algorithm handles a shortened list. The process continues until all points contributing to straight line segments have been removed from the list and from the edge image, See Fig 1(c).

Circle detection proceeds using this reduced edge image, Fig 1(c). In this second stage the same segments are used. One image point is fixed and used in combinatorial variations of that point and two other points from the list following it, provided that such sets of points are not co-linear. The circle parameters are calculated as outlined in section 1.2. Three histograms are generated for x_c, y_c and r, the circle parameters. Each histogram exhibits a peak, these are detected and the points which have contributed to that particular circle are removed from the list.

If the choice of first image point is good then the values accumulated in the histograms converge to single maximum values in the first iterations of this particular stage of the algorithm. Should it happen that the first point is a poor choice, i.e. a spurious point not located on a circle, then multiple peaks will occur in the histograms.

Such multiple peaks begin to develop in the first iterations of the algorithm. Thus a check can be made relatively early in the computation cycle and where multiple local maxima are detected then this pass of the algorithm may be abandoned at little computational cost and a new first point chosen.

Extraneous associations of points which reside on different line segments, or on a line and a circle, occur, but their incidence is much reduced by segmentation, and by the removal of straight line segments from the image before the algorithm for circle detection is applied.

Once all segments have been processed, a list of parameters of straight lines and circles in the image is obtained, and can be used to reconstruct the image (fig. 1d).

3 . Conclusion

A new algorithm, the Dynamic Combinatorial Hough Transform, DCHT, has been presented. It uses information present in the location of the feature points to reduce the generation of evidence in the transformation process. The DCHT offers significant improvements to previously suggested implementations of the Hough transform. The algorithm is computationally less intensive. It is also much more efficient in memory utilization. Rather than using an accumulator array whose dimension, n, corresponds to the number of parameters under detection, it requires only n one dimensional vectors in which to accumulate the results of the transformation. The length of the vectors is chosen with respect to the required resolution of the detection process.

Further computational savings may be made by segmenting the image. In addition, if the calculated parameters are not within the range of possibilities suggested by the shape under detection, this pass of the algorithm may be abandoned. Such a strategy will deal with membership of extraneous curves accidentally generated.

A further advantage of using the algorithm is that peak detection is one dimensional and the method provides a feedback mechanism whereby the end points of curves may be detected.

References

[1] **Hough P.V.C.** *Method and means for Recognising complex patterns.* U.S. Patent No. 3069654, 1962

[2] **Ballard D.H.** *Generalizing the Hough Transform to detect arbitrary shapes*, Pattern Recognition, Vol 13. No. 2, 111-122, 1981.

[3] **Gerig G. and Klein F.** *Fast contour identification through efficient Hough transform and simplified interpretation strategies.* Proc. 8th Int. Conf. Pattern Recognition, Vol 1, Paris 1986.

[4] **Li H., Lavin M.A. and LeMaster R.J.** *Fast Hough Transform* Proc. of 3rd workshop on computer vision: Bellair, Michgan 1985

[5] **Li H.** *Fast Hough Transform for multidimensional signal processing,* IBM Research report, RC 11562, York Town Heights 1985

[6] **Illingworth J. and Kittler J.** *The adaptive Hough transform* in press IEEE T-PAMI 1987

[7] **Forman A.V** *A modified Hough transform for detecting lines in digital imagery,* 151-160, SPIE, Vol 635, Applications of Artificial Intelligence III, 1986.

[8] **Jain A.N. and Krig D.B.** *A robust Hough transform technique for machine vision,* Proc. Vision 86, Detroit, Michigan, 86.

[9] **Illingworth J. and Kittler J.** *A survey of the Hough transform,* accepted for publication in IEEE Trans. on Pattern Analysis and Mach. Intell., 1987.

[10] **Leavers V.F. and Sandler M. B.** *An efficient Radon transform* BPRA 4th International Conf., Cambridge, 1988.

[11] **R.S. Wallace** *A modified Hough transform for lines,* IEEE conference on computer vision and pattern recognition, San Francisco 1985, 665-667.

[12] **Leavers V.F.** *Method and Means of Shape Parametrisation,* British Patent Application N0. 8622497. September 1986.

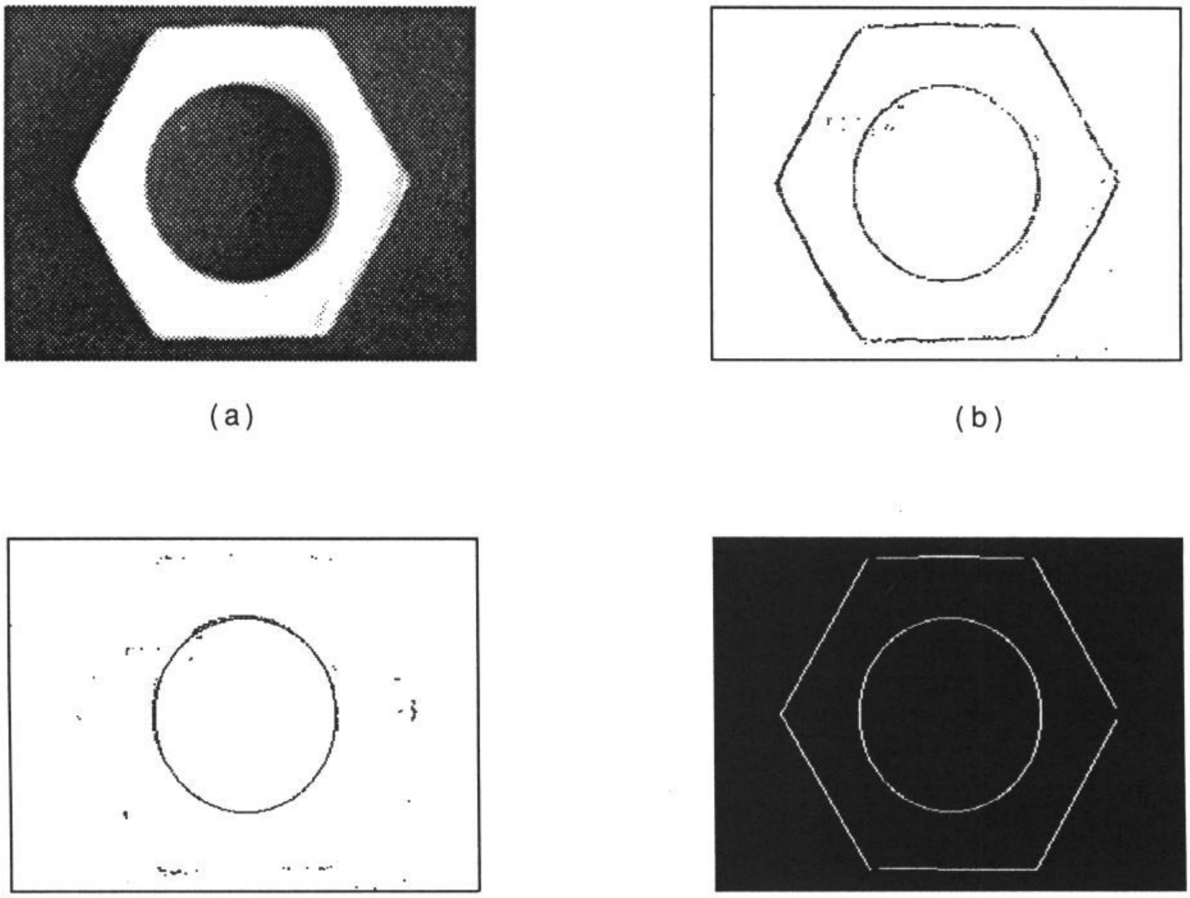
[13] **Leavers V.F. and Boyce J.F.** *The Radon transform and its application to shape detection in computer vision,* Image and Vision Computing Vol.5, May 1987.

[14] Gerig G. *Linking image-space and accumulator-space. A new approach for object recognition* Proc. of First Int. Conf. on Computer Vision, London, June 1987

[15] **Deans S.R.** *Hough transform from the Radon transform.* IEEE Trans. Pattern Analysis and Machine Intelligence. Vol. PAMI-3, No., March 1981.

[16] **Gel'fand I.M., Graev M.I. and Vilenkin N.Ya.** *Generalized functions* Vol 5, Academic Press, New York, 1966.

[17] **Leavers V.F.** *Dynamic Generalized Hough Transform*, British Patent Application, April 1989.



(c) (d)

Fig. 1 Applying the DCHT algorithm to an image :
(a) the original image (b) the edge image
(c) the edge image after the removal of straight lines
(d) the reconstruction of straight lines and circles in the image.