TABLE IV
CLASSIFICATION RESULTS FOR LAKE CONTOURS

		MICHIGAN	SUPERIOR	ERIE	HURON
10% missing	MICHIGAN	20	0	0	0
	SUPERIOR	0	20	0	0
	ERIE	0	0	20	0
	HURON	0	0	0	20
20% missing	MICHIGAN	20	0	0	0
	SUPERIOR	1	19	0	0
	ERIE	0	0	20	0
	HURON	0	3	0	17
30% missing	MICHIGAN	18	1	1	0
	SUPERIOR	3	17	0	0
	ERIE	0	2	18	0
	HURON	0	8	0	12

TABLE V
CLASSIFICATION RESULTS FOR AIRCRAFT CONTOURS

		CLASS I	CLASS II
10% missing	CLASS I	44	1
	CLASS II	1	74
20% missing	CLASS I	36	9
	CLASS II	5	70
30% missing	CLASS I	23	22
	CLASS II	29	46

periment except that the translation and rotation parameters used in the preparation of the testing set used some other arbitrary values. For the great lakes data, 81.25 percent recognition accuracy was obtained even when 30 percent of the data was missing. For the aircraft data, the recognition accuracy reduced to 57.5 percent even when only 30 percent of the data was missing. The use of only low order coefficients may indeed be the reason for low success rate when detail features are present as in the case of aircrafts. We also have to note the aircraft shapes used are quite similar to each other which is certainly a hard situation for any algorithm.

As to the applicability of the algorithm, we may have the partial boundary as the contour of an occluded object where the occluding boundary can be detected and then the algorithm can be applied to the occluded contour.

ACKNOWLEDGMENT

The authors wish to acknowledge the helpful discussions of Dr. P. F. Singer during various stages of this work.

REFERENCES

[1] W. F. Schreiber *et al.*, "Contour coding of images," in *Picture Bandwidth Compression*, T. S. Huang and O. J. Tretiak, Eds. New York: Gordan and Breach, 1972, pp. 443–448.
[2] R. Chellappa and R. Bagdazian, "Fourier coding of image boundaries," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-6, pp. 102–105, 1984.
[3] A. Oosterlink *et al.*, "Evaluation of different profile description and decomposition methods for banded chromosomes," in *Proc. 3rd Int. Joint Conf. Pattern Recognition*, Nov. 1976, pp. 334–338.
[4] H. Wechsler and J. Sklansky, "Finding the rib cage in chest radio graphs," *Pattern Recognition*, vol. 9, pp. 21–30, Jan. 1977.
[5] A. Ambler *et al.*, "A versatile system for computer controlled assembly," *Artificial Intell.*, vol. 6, pp. 129–156, 1975.
[6] C. Rosen *et al.*, "Exploratory research in advanced automation," Stanford Res. Inst., Tech. Rep., Aug. 1974.
[7] S. A. Dudani *et al.* "Aircraft identification by moment invariants," *IEEE Trans. Comput.*, vol. C-26, pp. 39–46, Jan. 1977.
[8] C. W. Richard and H. Hemami, "Identification of three-dimensional objects using Fourier descriptors of the boundary curve," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp. 371–378, July 1974.
[9] M. K. Hu, "Visual pattern recognition by moment invariants," *IEEE Trans. Inform. Theory*, vol. IT-8, pp. 179–187, Feb. 1962.
[10] G. H. Granlund, "Fourier preprocessing for hand printed character recognition," *IEEE Trans. Comput.*, vol. C-21, pp. 195–201, Feb. 1972.
[11] C. T. Zhan and R. S. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Comput.*, vol. C-21, pp. 269–281, Mar. 1972.
[12] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 170–179, Mar. 1977.
[13] R. L. Kashyap and R. Chellappa, "Stochastic models for closed boundary analysis: Representation and reconstruction," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 627–637, Sept. 1981.
[14] P. F. Singer and R. Chellappa, "Classification of boundaries on the plane using stochastic models," in *Proc. Conf. Computer Vision and Patt. Recog.*, Washington DC, June 1983, pp. 146–147.
[15] S. R. Dubois and F. H. Glanz, "An autoregressive model approach to 2-D shape classification," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-8, pp. 55–66, Jan. 1986.
[16] K. S. Fu, *Syntactic Pattern Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[17] T. Pavlidis and F. Ali, "A hierarchical syntactic shape analyzer," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-1, pp. 2–9, Jan. 1979.
[18] L. Davis, "Shape matching using relaxation techniques," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-1, pp. 60–72, Jan. 1979.
[19] B. Bhanu and O. D. Faugeras, "Shape matching of two-dimensional objects," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-6, pp. 137–156, Mar. 1984.
[20] K. C. You and K. S. Fu, "A syntactic approach to shape recognition using attributed grammars," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, pp. 334–344, June 1979.
[21] W. H. Tsai and K. S. Fu, "Attributed grammars: A tool for combining syntactic and statistical approaches to pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, pp. 873–885, Dec. 1980.
[22] O. D. Faugeras and M. Berthod, "Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-3, pp. 412–424, July 1981.
[23] O. R. Mitchell and T. A. Grogan, "Global and partial shape discrimination for computer vision," *Opt. Eng.*, vol. 23, pp. 484–491, Oct. 1974.
[24] P. F. Singer and R. Chellappa, "Machine perception of partially specified planar shapes," in *Proc. Conf. Computer Vision ana Patt. Recog.*, San Francisco, CA, June 1985, pp. 497–502.
[25] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[26] T. Poggio and V. Torre, "Ill-posed problems and regularization analysis in early vision," Massachusetts Institute of Technol. AI Lab. Memo 773, 1984.
[27] T. A. Grogan, "Shape recognition and description: A comparative study," Ph.D. dissertation, School of Elec. Eng., Purdue Univ., 1983.

The Adaptive Hough Transform

J. ILLINGWORTH AND J. KITTLER

***Abstract*—We introduce the Adaptive Hough Transform, AHT, as an efficient way of implementing the Hough Transform, HT, method for the detection of 2-D shapes. The AHT uses a small accumulator**

Manuscript received May 2, 1986; revised February 4, 1987. This work was supported by Alvey/SERC under Grant MMI/078.

The authors are with the Department of Electronic and Electrical Engineering, University of Surrey, Guildford GU2 5XH, England.

IEEE Log Number 8715805.

array and the idea of a flexible iterative "coarse to fine" accumulation and search strategy to identify significant peaks in the Hough parameter spaces. The method is substantially superior to the standard HT implementation in both storage and computational requirements. In this correspondence we illustrate the ideas of the AHT by tackling the problem of identifying linear and circular segments in images by searching for clusters of evidence in 2-D parameter spaces. We show that the method is robust to the addition of extraneous noise and can be used to analyze complex images containing more than one shape.

***Index Terms*—Adaptive search, circle finding, Hough Transform, line finding, shape detection.**

Introduction

Many image analysis problems involve recognizing instances of models which are characterized by a few parameters. The Hough Transform, HT, is a powerful method of parameter extraction. It utilizes local image features to efficiently accumulate evidence for all possible model instances. This makes it both robust to the addition of extraneous data and sensitive to the presence of only part of an instance. Its independent use of image features makes it suitable for implementation in a parallel computing system. It has been used in computer vision for the determination of shape [1], motion [2], [3], and geometric transformation parameters [4].

The HT method involves a mapping from features in an image space to sets of points in a parameter space. Each point of parameter space represents an instance of the model in image space. Image features are mapped into parameter space using a function which generates all parameter combinations compatible with both the observed image feature and the hypothesized model. Each image feature will generate a different surface in multidimensional parameter space but all surfaces generated by features belonging to the same instance of the model will intersect at the common point which describes the instance. The basis of the HT is to generate these surfaces and identify the parameter point where they intersect.

The standard implementation of the HT involves representing the continuous space of parameters by an appropriately quantized space. The fineness of quantization is chosen to reflect the accuracy required in the determination of individual parameters. The surface generated by an image feature can be marked by testing for its intersection with cells of this quantized space and then incrementing a counter associated with this cell. The collection of counters forms a multidimensional array which is usually referred to as the accumulator array. Image features can be said to provide evidence for, or vote for a model instance which has parameter values encompassed by the boundaries of the incremented cell. The point of intersection of many parameter surfaces is denoted by a high number of counts in the relevant accumulator element. Therefore, the HT method requires the identification of significant local maxima in the number of counts within the accumulator array.

In the standard HT implementation the peak finding problem is solved by exhaustive enumeration and search of a finely quantized parameter space. However the use of a large accumulator array is undesirable in many respects. It requires a large amount of computer storage. If k parameters have to be determined and each parameter range is divided into α intervals the storage required is α^k elements. If either k or α is large this may become an intolerably large burden. A large accumulator array also produces a large computational problem because it requires that many parameter cells have to be tested and updated to map out the parameter surface of each image feature. However, a lot of this computation is concerned with computing a general background above which we would like to find significant local peaks. We are really only interested in the small number of cells in the vicinity of these peaks and therefore we need a method of locating the peak areas without computing the background level in fine detail. We suggest that in many cases this can be done using an intelligent iterative "coarse to fine" accumulation and search strategy. This method uses a small fixed size accumulator and partitions the range of parameters of current interest into these few intervals. Initially each parameter is resolved only very coarsely. The basic HT can be accumulated and the pattern of counts can be used to guide a redefinition of the parameter limits.

In the simplest case of a single noise free parameter peak the parameter range can be reduced and the HT reaccumulated at a higher resolution. Silberberg *et al.* [4] have described how iteration of this strategy can swiftly lead to the accurate determination of transformation viewing parameters. However in many situations involving several objects and/or spurious data the parameter space is more complex and quantization effects make its interpretation more difficult. In these cases it is necessary to adopt more complex strategies involving combinations of shifting, expanding and reducing the range of parameters of current interest. If peaks become broad and ridge-like it is important that these changes can be made independently in each of the parameter dimensions. We have called our iterative windowing implementation of the Hough Transform method the Adaptive Hough Transform, AHT, because the method basically involves adapting to the information contained in the accumulator array.

Li *et al.* [5], [6] have recently published two elegant approaches to the implementation of the HT which are similar in spirit to the AHT in so far as they investigate the HT at several spatial scales and attempt to evaluate it in detail only in those areas which have a high density of points. However while we attempt to incorporate maximum flexibility in parameter redefinition in order to successfully analyze complex parameter spaces they adopt very regular hierarchical data structures in order to develop a simple algorithm which is maximally efficient in both storage and computation and is very easily extended to multidimensional parameter spaces. Section VI discusses in greater detail some of the merits of each approach.

In this correspondence we develop the AHT approach to detect shapes which can be characterized by two parameters. This class of shapes includes both linear and circular image features and we illustrate the method using these shapes. In Section II we discuss the parameterizations of lines and circles that are used. Section III describes the AHT algorithm for two parameters in greater detail while Section IV presents a simple analysis which demonstrates that both the computational and storage demands of the AHT are significantly less than those of the standard HT. These assertions are tested experimentally in Section V for images containing both several features and a lot of extraneous noise. Section VI constrasts the AHT with the work of Li *et al.* and considers how the AHT can be extended to higher dimensional parameter spaces. Section VII offers brief conclusions.

II. Shape Parameterizations

A large class of man-made objects and natural features can be modelled using linear and circular primitives. The recognition of these shapes is therefore basic to many image analysis tasks. In this section we consider the parameterizations which we might use with the HT to detect these shapes.

Line Finding

An infinite length straight line can be described by the equation

$$y = mx + c \tag{1}$$

where (x, y) are coordinates of points in image space and (m, c) are two parameters, the slope and the intercept of the line with the y axis. In the HT image points (x, y) are mapped into all (m, c) values which are compatible with (1). The inversion of (1) indicates that these values lie on the surface

$$c = -xm + y \tag{2}$$

which is a straight line in parameter space with slope $-x$ and intercept y. The parameters of lines in image space are given by the intersection of many of these parameter space lines.

A problem with the (m, c) parameterization of lines is its inability to describe vertical lines, i.e., $m \to \infty$. Therefore (1) is only used for lines with $|m| < 1$. A second equation which swaps the roles of the x and y axes,

$$x = m'y + c' \tag{3}$$

is used for lines with $|m| > 1$. This introduces a second independent parameter space (m', c').

A popular parameterization of a straight line is via the equation of the normal vector from the origin to the straight line, viz.

$$\rho = x \cos(\theta) + y \sin(\theta) \tag{4}$$

where ρ is the length of the vector and θ is the angle it makes with the x axis. Using this parameterization each image point (x, y) generates a sinusoidal locus in (ρ, θ) space and thus lines are identified by the intersection of many of these sinusoids. In the current work we did not investigate this parameterization.

Circle Finding

A circle can be described by

$$(x - a)^2 + (y - b)^2 = r^2 \tag{5}$$

where (a, b) are the coordinates of the circle center and r is the radius of the circle. The HT using this equation maps each (x, y) image point into all parameter points which lie on the surface of an inverted right angled cone whose apex is at $(x, y, 0)$. Circle parameters are identified by the intersection of many conic surfaces. However, in order to unify our treatment of line and circle finding we decided not to use (5) directly. The circle finding problem can be reformulated to produce a two stage algorithm. The first stage involves a two parameter HT to find the circle center while the second stage is a simple histogramming step to identify the radius of the circle. The second step can be regarded as a form of one dimensional HT. To locate the circle center we incorporate the constraint that the vectors which are normal to the circle boundary must all intersect at the circle center (a_0, b_0). Estimates of these normal directions can be obtained by local gray-level edge detection operators, i.e., the Sobel operator. Fig. 1 illustrates how knowledge of (x, y, θ) leaves only (a, b) as parameters. A mapping of (x, y, θ) triples into 2-D parameter space produces a straight line. The intersection of many of these lines identifies the circle center coordinates. Much of the code used to identify straight lines in image space can be used in this center finding step. The circle radius can be found by histogramming, $\delta = (x - a_0)^2 + (y - b_0)^2$ where (a_0, b_0) are the center coordinates determined by the first stage two parameter HT. The radius can be identified as the largest peak in the δ histogram.

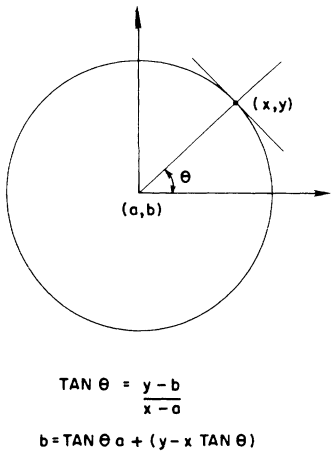


Fig. 1. Relationship between (x, y, θ) and center parameters (a, b) for a circle.

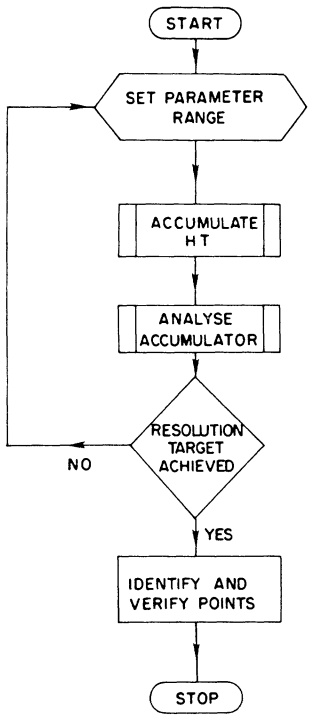


Fig. 2. Flowchart for AHT algorithm.

III. The AHT Algorithm

This section deals in detail with the AHT as applied to 2-D parameter spaces. Fig. 2 is a flowchart which illustrates the control strategy used to identify an instance of an object. The scheme consists of accumulating the HT in a small size accumulator and using this information intelligently to redefine the parameter range so that interesting areas can be investigated in greater detail. This cycle of events continues until parameters are determined to a prespecified accuracy. The located parameters can be used to identify, by an inverse mapping, the data points which may belong to the object. Some post analysis of the subset of identified points may be necessary to validate their identification.

If the points identified as belonging to an instance of an object class are removed from the list of candidate points, then the search can be reinitialized at the coarsest resolution and a search for another object instance can be made. Images containg multiple objects can therefore be processed until the parameter space contains no significant structure.

One of the first decisions which has to be made in the AHT method is the size of the small accumulator array. Efficiency arguments would lead to selection of the smallest size array which partitions the space, i.e., 2×1 or 2×2. However it is also necessary to provide enough samples of the parameter space so that the general topology of the space can be reliably interpreted. In the present work we chose a 9×9 accumulator array as a reasonable compromise between these considerations.

To accumulate the HT it is necessary to identify those parameter cells which are intersected by the surfaces generated by each image feature point. The 2-D parameterizations we have used map points into straight lines in the parameter spaces. The intersection of these lines with each of the grid lines which define the cells partitioning the quantized space can be easily calculated. For a parameter axis divided into β intervals there will be at most $(\beta + 1)$ such intersections. The cells either side of the intersection points lie on the line and a bit in a logical array can be set to denote this. Repetition of this along the second parameter axis will lead to all relevant cells being identified and flagged. The accumulator array is incremented if the corresponding logical array bit is set. The accumulation of the logical array requires the calculation of $2(\beta + 1)$ line-grid intersections for each image feature point.

Once the HT has been accumulated the parameter space must be analyzed. The detection of significant maxima can be achieved using a scheme which binarizes the accumulator array and labels and analyzes connected components in this binary array. The threshold chosen to binarize the accumulator was calculated as a fixed fraction of the highest count found in a single accumulator bin. A typical figure was 0.9 of the highest count. The connected components algorithm [7] transforms a binary array into a symbolic array so that all cells which are 0 remain 0 but every connected subset of 1 values is labeled by a distinct integer. The idea of the algorithm is to raster scan the image and assign labels to the 1 values. We can illustrate this by considering two lines of a binary image

line $r-1$ $\quad x\ x\ B\ x\ x$

line r $\quad x\ C\ A\ x\ x$

where A, B and C are the binary values of interest in the relabelling decision and the x values are of no interest. If the pixel labelled A is a 1 we assign A a label as follows:

if $B = C = 0$ then A is given the next available label

if $B = 0\ C \neq 0$ then A is given C's label

if $B \neq 0\ C = 0$ then A is given B's label

if $B \neq 0\ C \neq 0$ then A is given the smaller of the two labels and the equivalence of B's label and C's label is stored.

Equivalent labels can be resolved in a second pass through the image. The smallness of the accumulator array and the fact that this analysis was performed only once per iteration meant that the computational burden of the strategy was low. Features of the individual connected components, such as the number of counts in the accumulator bins which constituted the connected component and the extent of the component along each of the parameter axes were calculated. These were used to choose the most significant peak and were used to guide the adaptive strategy for the adjustment of the parameter limits.

In the current correspondence the criteria used to identify the best peak was the density of the connected component, i.e., number of counts/number of accumulator cells. The action taken in redefining parameter limits for the next iteration depended on the shape, extent and location of the best peak. An important aspect of the algorithm is its ability to redefine parameters independently in each parameter dimension at each iteration. The center of the new parameter range was always placed at the centroid of the best peak. The width of the new range was set to the extent of the connected component along each of the parameter axes. However if the extent was less than three bins the width was set to three bins. This was a conservative strategy used to overcome problems which might be encountered if the placement of cell boundaries led to peaks straddling two bins.

In two situations the redefinition of the parameter limits did not follow the simple strategy of the paragraph above. First, whenever the best connected component was compact in extent but adjacent to the edge of the accumulator it was assumed that the peak extended outside the current parameter range and therefore the parameter limits were set to center about this component but the width of the range was left unchanged. A second exceptional configuration was often encountered during the first iteration of line finding. The shape of the connected component was a bar of 1 bin thickness stretching across the accumulator. In this case the parameter limits were reduced in the thin direction but left unchanged in the orthogonal parameter direction. This serves to fan out the distribution of parameter lines and leads to a more evenly populated and therefore more easily interpreted accumulator array.

IV. Complexity of the AHT Method

We can make an approximate calculation of the potential efficiency of the AHT relative to the standard HT if we consider the case of detecting a single object characterized by 2 parameters and assume that at each iteration of the AHT we increase the resolution of search by the largest factor, i.e., we focus in to the correct peak in as few iterations as possible. If we wish to resolve each parameter to $1/\alpha$ ($\alpha > 1$) of its full range then in the standard implementation we would use an accumulator of size α^2. In the AHT we iteratively use an accumulator of size β^2 where $\beta < \alpha$. The storage advantage R_s is therefore

$$R_s = \frac{\text{standard HT accumulator size}}{\text{AHT accumulator size}} = \left(\frac{\alpha}{\beta}\right)^2. \tag{6}$$

This can be a very large ratio. The computational advantage R_c depends on the amount $1/\gamma$ by which the parameter range is reduced at each iteration. If we assume that this is the same ratio in both dimensions for all iterations then the number of iterations n before the AHT reaches the same resolution as the HT is given by

$$\left(\frac{1}{\gamma}\right)^n = \frac{1}{\alpha} \tag{7}$$

thus

$$n = \frac{\log(\alpha)}{\log(\gamma)}. \tag{8}$$

If the number of input feature points N is sufficiently large then the major computational effort at each iteration is in accumulating the HT. We have seen in Section III that for each point this step involves $2(\beta + 1)$ line-grid intersection calculations. For the standard HT we would have to calculate α values of the second dependent parameter as we scanned along the α possible independent values of the first parameter. Therefore a single iteration of the AHT is of the order of $\alpha/2(\beta + 1)$ times more efficient than the standard HT. Thus the relative efficiency for finding the same object will be

$$R_c = O\left(\frac{\alpha}{2(\beta + 1)} \frac{\log(\gamma)}{\log(\alpha)}\right). \tag{9}$$

As an illustrative example consider the case of locating a circle whose center is in a 512×512 image. We would like to locate its center coordinates to an accuracy of 1 pixel, i.e., $\alpha = 512$. In our AHT implementation $\beta = 9$ and $\gamma = \beta/3 = 3$. Thus

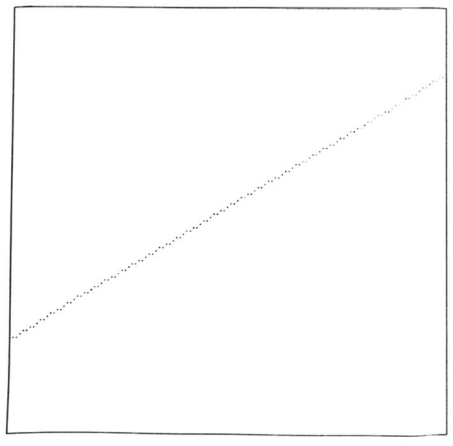
$$R_c = \frac{512 \log(3)}{20 \log(512)} = 5. \tag{10}$$

This example shows that potentially the AHT is approximately 5 times faster than the standard implementation and uses an accumulator which is some 3000 times smaller!

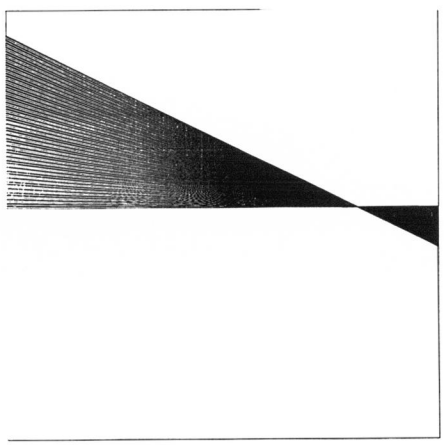
V. Experimental Results

The performance of the AHT was tested using simple images containing digital line and circle segments generated by standard graphics algorithms [8]. The suitability of the AHT to detect these shapes depends on correctly identifying the high density regions which they produce in the parameter space even when the space is coarsely quantized. This is mainly influenced by the form of the HT mapping and by the distribution and amount of spurious data.

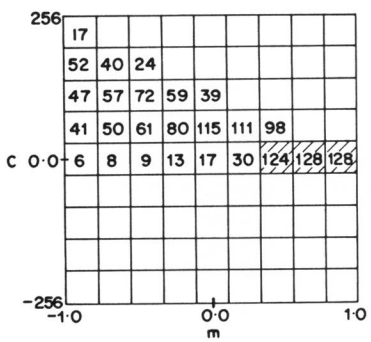
Figs. 3 and 4 illustrate the mapping of lines and circles into their respective (m, c) and (a, b) parameter spaces. Fig. 3(a) is the image of the line, Fig. 3(b) is a plot of the parameter lines that image points generate in (m, c) space, and Fig. 3(c) is the corresponding accumulator totals when the space is coarsely quantized in a 9×9 array. Fig. 4 is divided into three similar sections. It is clearly seen that the two mappings produce dramatically different parameter spaces. The points generated by an image space line populate a cone-shaped region of (m, c) space and produce several accumulator bins, near the apex of the cone, with approximately the same high number of counts. When the accumulator array is thresholded the three bins which are shown hatched in Fig. 3(c) are identified as the global peak. The new parameter limits of the



(a)

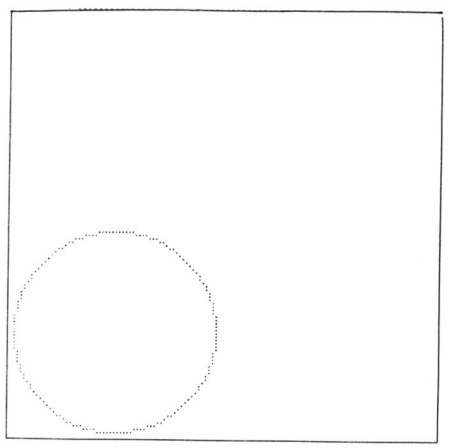


(b)

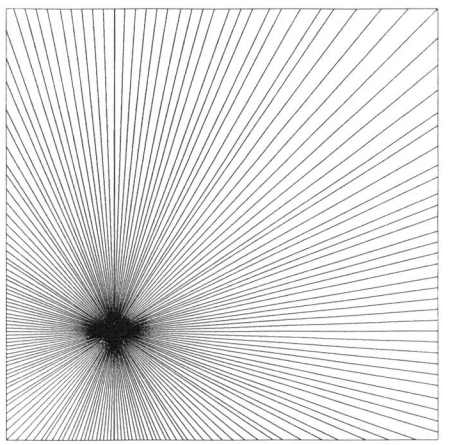


(c)

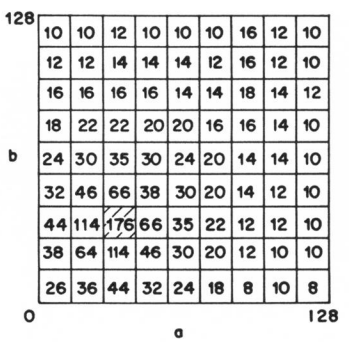
Fig. 3. Detection of linear segments using the AHT. (a) Image. (b) Parameter space (limits $m = -1 \rightarrow 1$, $c = -256 \rightarrow 256$). (c) Coarsely quantized accumulator. The shaded cells indicate the area selected by thresholding the array.



(a)



(b)



(c)

Fig. 4. Detection of circle centers using the AHT. (a) Image. (b) Parameter space (limits $a = 0 \rightarrow 128$, $b = 0 \rightarrow 128$). (c) Coarsely quantized accumulator. The shaded cell is the peak selected by thresholding the array.

next iteration are chosen around this peak. However because the peak touches the edge of the array the parameter limits are translated but their range is not reduced. This results in the apex being centred in the array. Accumulation of the HT at this resolution confirms that the peak is compact and totally within the parameter limits and therefore in subsequent iterations the method reduces the parameter range and focuses in on the peak. The generated parameters of the line were $m = 0.62$ and $c = 20$. After 7 iterations of the AHT the parameters were determined as $m = 0.624 \pm 0.007$ and $c = 20.02 \pm 0.58$.

In the case of the circle of Fig. 4(a) the parameter space is much more symmetrically populated about the true point of intersection. The thresholded peak is therefore more localized and much easier to interpret. The single hatched bin in Fig. 4(c) is identified and the parameter limits are redefined to include only this bin and its immediate neighbors. Subsequent iterations provide similar focusing behavior and the parameters of the circle center are determined as $a = 31.87 \pm 0.79$ and $b = 31.87 \pm 0.79$ after only 4 iterations. The circle was generated with its centre at $a = 32$, $b = 32$. The number of iterations required agrees with the prediction of Section IV.

The effect of spatially random point noise on the AHT method can be gauged by studying the parameter space plots in Fig. 5. Fig. 5 shows how 1000 spatially random noise points map into the (m, c) and (a, b) spaces. It is notable that the (a, b) space is much more uniformly populated than the (m, c) space. The uniformity of the background in (a, b) space means that it requires a consid-

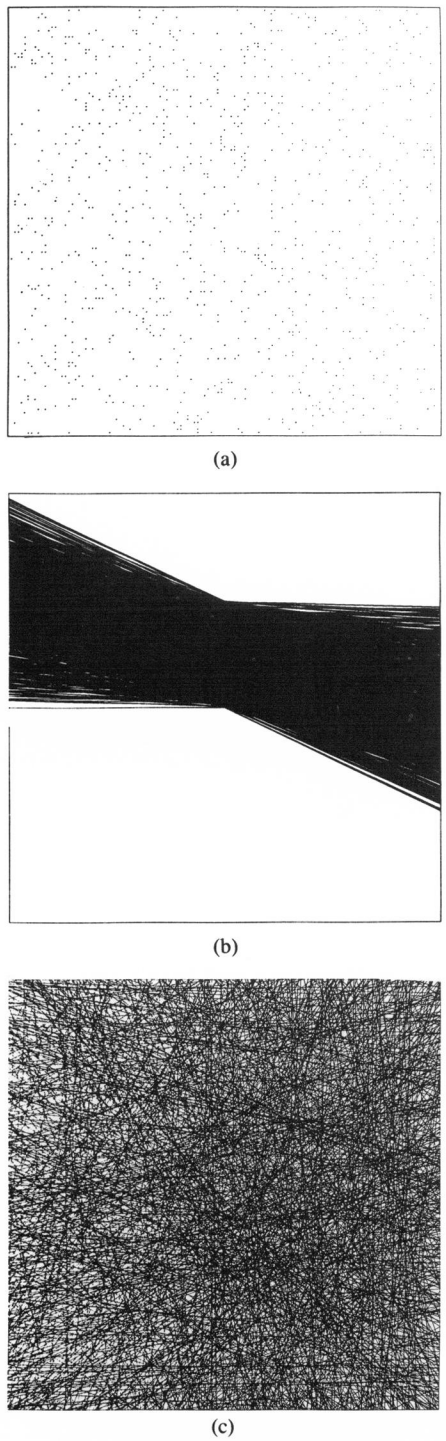


Fig. 5. Mapping of spatially random noise pixels into (a) (m, c) space, (b) (a, b) space.

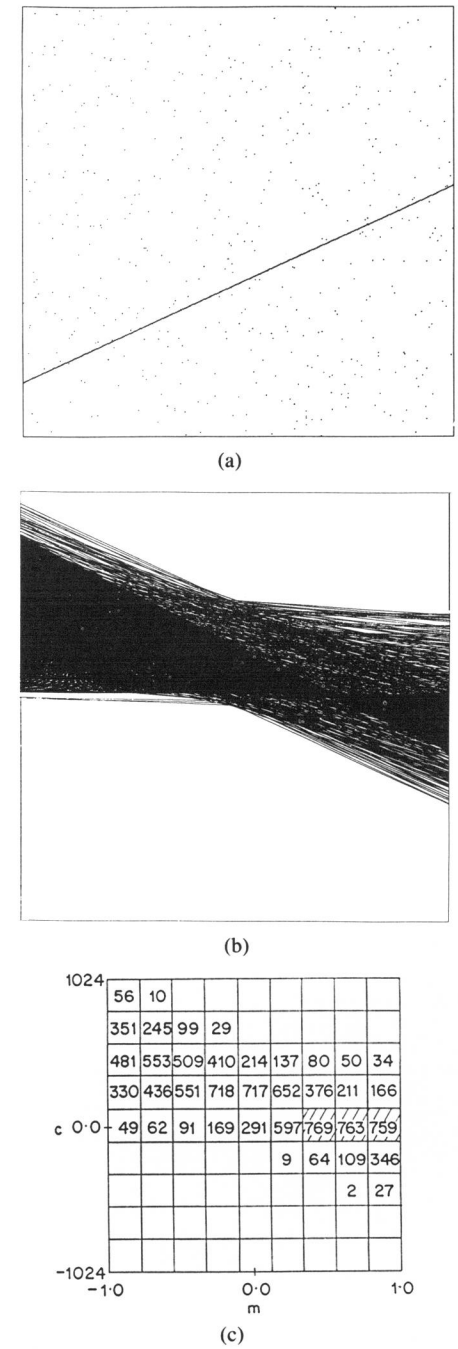


Fig. 6. Detection of line in noisy image. (a) Image, 512 line points, 512 random points. (b) Parameter space. (c) Accumulator at coarsest quantization.

erable amount of point noise before peaks due to true circles are confused by the background. The visibility of peaks above the background is obviously affected by the structure of the background, the location of the peak, and the ratio of the number of signal and noise points. These issues are discussed in relation to the standard HT method in [9] and [10]. It is suggested that when the characteristics of the noise are known, its effects can be compensated by either choosing a nonlinear maximum entropy quantization of parameters or by subtracting an empirically determined background factor. In the current work neither of these corrections were necessary for realistic levels of random point noise.

Fig. 6 illustrates the line finding AHT for a line with parameters $(m = 0.46, c = 64)$. The line has 512 points and an equal number of spatially random pixels have been added to the image. Fig. 6(b) shows the parameter space produced by the HT mapping and Fig. 6(c) shows the accumulator at its coarsest resolution. The method determines the parameters as $m = 0.461 \pm 0.002$, $c = 63.99 \pm 0.62$ after 7 iterations.

Fig. 7 shows a circle defined by (a, b, r) parameters (64, 64, 32) embedded in an image containg 10 times as many noise points. Fig. 7(b) is the parameter space after the mapping. After 6 iterations the AHT method finds parameters of $a = 64.02 \pm 0.3$ and $b = 64.03 \pm 0.3$. The number of iterations agrees well with the formulas of Section IV. The radius of the circle can be found by histogramming the distance of points from the determined circle centre coordinates. Fig. 7(c) shows this histogram. A distinct peak

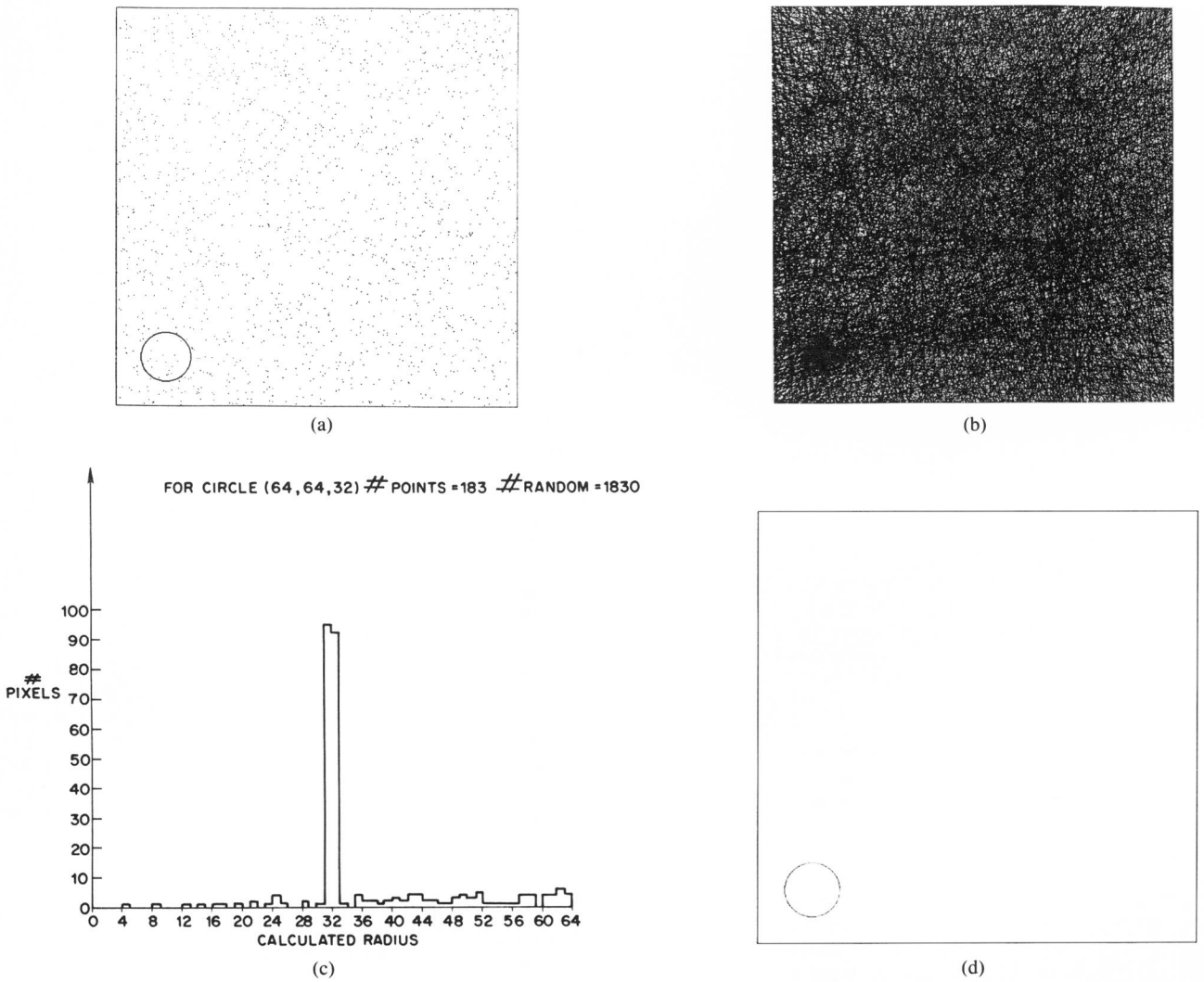


Fig. 7. Detection of circle in a noisy image. (a) Image: 183 circle points, 1830 random points. (b) Parameter space: peak accumulator bin has 420 counts above an average background of 306. (c) Radial distance of image points from estimated center. (d) Identified circle points.

is seen at $r = 32$. Fig. 7(d) shows the points which match the determined set of parameters.

As a final example of the use of the AHT we consider detecting the elements of the semicircle shown in Fig. 8. If the program is asked to search for a circle it correctly identifies all the points which form the arc of the semicircle. These points can be removed from consideration and then the line can be found by searching the (m, c) space. A series of such searches can lead to the identification of many of the simple features in an image and these can be matched to structural models of a scene.

VI. Discussion

Li *et al.* [5], [6] have recently suggested two elegant and efficient approaches to the implementation of the HT. They call their methods the Fast Hough Transform, FHT, and the Fast Hough Transform based on the Bintree data structure, FHTB. The approach of their methods is similar to our own in so far as the parameter space is investigated at several resolutions but is only evaluated in fine detail in those regions where a high density of points occur. However their methods differ from ours in the degree of freedom allowed in the redefinition of parameter limits, i.e., the degree of flexibility in the placement and choice of shape of the window which defines the range of parameters under study. The methods of Li *et al.* employ hierarchical data structures together with a simple thresholding strategy to select single cells which are then investigated at higher resolution. Once the initial parameter limits have been chosen, these methods permit only a restricted set of possibilities for the positioning and size of subsequent parameter windows. Li *et al.* acknowledge that in some cases peaks may straddle the boundaries between these fixed windows and then further analysis will be needed to successfully interpret such situations. In the AHT method these situations are handled more naturally because at each resolution we employ a complex strategy to analyze the parameter space and this means we can define more appropriate parameter limits for subsequent processing. We permit translation of the parameter limits by nonintegral multiples of the cell distance and can easily allow for the range of parameters to increase, decrease, or remain the same. It is especially interesting to note that changes in resolution can be made independently in each of the parameter dimensions every time the parameter limits are redefined. This means that the relative precision with which each parameter is finally determined can be different in each of the parameter dimensions and is decided, by the data itself, dynamically as the method proceeds. This is a very useful property in cases where parameter peaks are likely to be highly asymmetric or ridge like. The FHTB method of Li is formulated with these con-

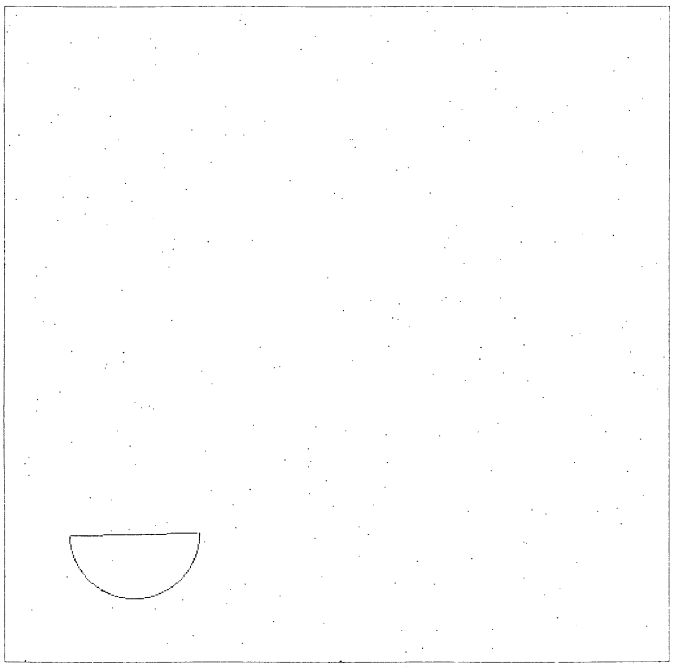


Fig. 8. Semicircle in noisy image. AHT can sequentially identify points belonging to the circular arc and the circle diameter.

siderations in mind but again the adoption of a basically static data structure means that it is less flexible than the AHT. In summary the strength of the AHT is that the size, shape, and positioning of the parameter windows is very flexible and is determined by the data itself rather than being imposed as a result of the choice of data structure and initial parameter limits.

The strong points of FHT and FHTB algorithms are that their data structures are very regular and the analysis is simple and therefore easily formulated for the multidimensional case. FHT and FHTB use a formulation of the HT in which image feature points map to hyperplanes in multidimensional space and a lot of the computational burden then involves calculating the distance between these planes and points which define the accumulator cell limits. The static nature of the data structures reduces this calculation to a simple form of bookkeeping in which distance can be recalculated by incremental formulas, i.e., after initial calculation of distances subsequent values are calculated from known fixed spatial relationships as relative shifts. The overall regularity and simplicity of the analysis means that it can exploit a high degree of parallelism and is very appropriate for implementation in special purpose hardware.

Thus far we have successfully experimented with the AHT for the detection of shapes described by two parameters. This is sufficient to recognize two of the commonest image features, the line and the circle. However in many applications we may wish to determine more than two parameters. The general principles of the AHT method are just as relevant to this multidimensional case as to the two-dimensional case but details of the implementation need some adaptation. In particular, the methods of determining parameter surface–accumulator cell intersection and the method of labeling connected cells in the thresholded multidimensional array need some revision.

The parameter surface-accumulator cell intersection problem is much simplified if we adopt the hyperplane formulation of the HT, i.e., we express the transformation equations so that image features always map into surfaces which are hyperplanes in the parameter space. Such a hyperplane partitions the space into two regions or half spaces and if a hypercell is intersected by the hyperplane then some of its vertices lie in one half space while others lie in the other half space. All vertices lie in the same half space if the hyperplane does not intersect the hypercell. The implicit equation of a hyperplane evaluated at any coordinate point yields a signed distance measure whose magnitude is proportional to the perpendicular distance between the point and the plane and whose sign indicates on which side of the plane the point lies. Thus a simple method of identifying all intersected cells consists of determining and storing the sign of the implicit equation of the hyperplane evaluated at the cell vertex coordinates and then searching all vertices of each cell to discover whether at least one of them differs in sign from all the others.

An important element in our analysis of the parameter space is the use of an efficient raster algorithm to label connected binary components. In order to extend the AHT beyond the two parameter case we need to extend these methods to find components in multidimensional binary arrays. Some work has been done in this direction by Lumia [11]. He demonstrates that the 2-D algorithm can be extended to label connected components in a 3-D binary array. Essentially a 3-D array can be viewed as a series of 2-D arrays arranged in bands. Cells are adjacent both to cells within these bands as well as to certain cells in the immediately neighboring bands. The 3-D problem can be viewed as similar to the 2-D problem except that bands replace lines. The problem can be solved by running the 2-D connected component algorithm on each of the bands separately and then determining cell adjacencies between bands. Only a small equivalence table is needed and the computational efficiency of the method only increases linearly with the number of cells in the array.

The potential benefit of using the AHT method in higher dimensions is even greater than the two dimensional cases we have demonstrated. The relative space efficiency is $R_s = (\alpha/\beta)^k$ for a k-dimensional case and is therefore a very huge saving factor. The number of iterations to reach the same resolution as the standard HT is independent of the dimensionality of the problem and therefore the calculational efficiency depends on the complexity of the procedure used in accumulating and analyzing the fixed size accumulator. In the revised method of surface–cell intersection we calculate the sign of the implicit equation of the plane evaluated at each hypercell vertex. In a k-dimensional array there are β^k cells which share $(\beta + 1)^k$ vertices. We therefore need to calculate for each image feature $(\beta + 1)^k$ implicit function values to determine the position of each vertex relative to the hyperplane. The 2^k vertices of each cell have then to be compared to determine whether they lie on different sides of the hyperplane. In the standard HT α^{k-1} values of the function are calculated to map the hyperplane into the accumulator array. If we assume the function evaluations are the major computational burden then the relative efficiencies of these multidimensional methods are $R_c \propto \alpha^{k-1}/(\beta + 1)^k$. Thus an increase of one in the dimensionality of the problem leads to an increase in efficiency of α/β. This will give a very significant saving for high dimensional spaces.

VII. Conclusions

We have demonstrated the feasibility of line and circle detection using the AHT and shown that the method retains the desirable HT characteristics of robustness to extraneous data and the ability to detect model parameters from partial data. The amount of storage used by the method is much less than that used in the standard HT implementation and this greatly increases the range of applications which can be attempted on small memory computers. The small accumulator size also produces significant computational advantages. Both of these efficiency gains should increase as the detection of more complicated shapes, characterized by higher dimensional shapes, is attempted.

References

[1] D. H. Ballard, "Generalizing the Hough Transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[2] R. A. Samy and C. A. Bozzo, "Moving object recognition using motion enhanced Hough Transform," in *Digital Signal Processing-84*,

V. Cappellini and A. G. Constantinides, Eds. Amsterdam, The Netherlands: Elsevier, pp. 770–775.
[3] J. O'Rourke, "Motion detection using Hough techniques," in *Proc. Conf. Pattern Recognition and Image Processing*, Dallas, TX, 1981, p. 737.
[4] T. M. Silberberg, L. Davis, and D. Harwood, "An iterative Hough procedure for three-dimensional object recognition," *Pattern Recognition*, vol. 17, no. 6, pp. 621–629, 1984.
[5] H. Li, M. A. Lavin, and R. J. LeMaster, "Fast Hough Transform," in *Proc. 3rd Workshop Computer Vision: Representation and Control*, Bellair, MI, 1985, pp. 75–83.
[6] H. Li and M. A. Lavin, "Fast Hough Transform based on Bintree data structure," in *Proc. Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, 1986, pp. 640–642.
[7] R. Lumia, L. Shapiro, and O. Zuniga, "A new connected components algorithm for virtual memory computers," *Comput. Graphics Image Processing*, vol. 22, pp. 287–300, 1983.
[8] A. Bowyer and J. Woodwark, *A programmers geometry*. London: Butterworth, 1983.
[9] M. Cohen and G. T. Toussaint, "On the detection of structures in noisy pictures," *Pattern Recognition*, vol. 9, pp. 95–98, 1977.
[10] T. M. Van Veen and F. C. A. Groen, "Discretization errors in the Hough Transform," *Pattern Recognition*, vol. 14, pp. 137–145, 1981.
[11] R. Lumia, "A new three-dimensional connected components algorithm," *Comput. Vision, Graphics, Image Processing*, vol. 23, pp. 207–217, 1983.

Least-Squares Fitting of Two 3-D Point Sets

K. S. ARUN, T. S. HUANG, AND S. D. BLOSTEIN

***Abstract*—Two point sets $\{p_i\}$ and $\{p_i'\}$; $i = 1, 2, \cdots, N$ are related by $p_i' = Rp_i + T + N_i$, where R is a rotation matrix, T a translation vector, and N_i a noise vector. Given $\{p_i\}$ and $\{p_i'\}$, we present an algorithm for finding the least-squares solution of R and T, which is based on the singular value decomposition (SVD) of a 3 × 3 matrix. This new algorithm is compared to two earlier algorithms with respect to computer time requirements.**

***Index Terms*—Computer vision, least-squares, motion estimation, quaternion, singular value decomposition.**

I. Introduction

In many computer vision applications, notably the estimation of motion parameters of a rigid object using 3-D point correspondences [1] and the determination of the relative attitude of a rigid object with respect to a reference [2], we encounter the following mathematical problem. We are given two 3-D point sets $\{p_i\}$; $i = 1, 2, \cdots, N$ (here, p_i and p_i' are considered as 3 × 1 column matrices)

$$p_i' = Rp_i + T + N_i \tag{1}$$

where R is a 3 × 3 rotation matrix, T is a translation vector (3 × 1 column matrix), and N_i a noise vector. (We assume that the rotation is around an axis passing through the origin). We want to find R and T to minimize

$$\Sigma^2 = \sum_{i=1}^{N} \| p_i' - (Rp_i + T) \|^2. \tag{2}$$

Manuscript received July 2, 1986; revised April 9, 1987. Recommended for acceptance by S. W. Zucker. This work was supported by the National Science Foundation under Grant IRI-8605400.

The authors are with the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801.

IEEE Log Number 8715809.

An iterative algorithm for finding the solution was described in Huang, Blostein, and Margerum [3]; a noniterative algorithm based on quaternions in Faugeras and Hebert [4]. In this correspondence, we describe a new noniterative algorithm which involves the singular value decomposition (SVD) of a 3 × 3 matrix. The computer time requirements of the three algorithms are compared.

After the submission of our correspondence, it was brought to our attention that an algorithm similar to ours had been developed independently by Professor B. K. P. Horn, M.I.T., but not published.

II. Decoupling Translation and Rotation

It was shown in [3] that: If the least-squares solution to (1) is $\hat{R}$ and $\hat{T}$, then $\{p_i'\}$ and $\{p_i'' \triangleq \hat{R}p_i = \hat{T}\}$ have the same centroid, i.e.,

$$p' = p'' \tag{3}$$

where

$$p' \triangleq \frac{1}{N} \sum_{i=1}^{N} p_i' \tag{4}$$

$$p'' \triangleq \frac{1}{N} \sum_{i=1}^{N} p_i'' = \hat{R}p + \hat{T} \tag{5}$$

$$p \triangleq \frac{1}{N} \sum_{i=1}^{N} p_i. \tag{6}$$

Let

$$q_i \triangleq p_i - p \tag{7}$$

$$q_i' \triangleq p_i' - p'. \tag{8}$$

We have

$$\Sigma^2 = \sum_{i=1}^{N} \| q_i' - Rq_i \|^2. \tag{9}$$

Therefore, the original least-squares problems is reduced to two parts:

(i) Find $\hat{R}$ to minimize Σ^2 in (9).
(ii) Then, the translation is found by

$$\hat{T} = p' - \hat{R}p. \tag{10}$$

In the next section, we describe an algorithm for (i) which involves the SVD of a 3 × 3 matrix.

III. An SVD Algorithm for Finding $\hat{R}$

A. Algorithm

Step 1: From $\{p_i\}$, $\{p_i'\}$ calculate p, p'; and then $\{q_i\}$, $\{q_i'\}$.

Step 2: Calculate the 3 × 3 matrix

$$H \triangleq \sum_{i=1}^{N} q_i q_i'^t \tag{11}$$

where the superscript t denotes matrix transposition.

Step 3: Find the SVD of H,

$$H = U\Lambda V^t. \tag{12}$$

Step 4: Calculate

$$X = VU^t. \tag{13}$$

Step 5: Calculate, det (x), the determinant of X.
If det $(x) = +1$, then $\hat{R} = X$.
If det $(x) = -1$, the algorithm fails. (This case usually does not occur. See Sections IV and V.)

0162-8828/87/0900-0698$01.00 © 1987 IEEE