COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 44, 87–116 (1988)

SURVEY

A Survey of the Hough Transform

J. ILLINGWORTH AND J. KITTLER

Department of Electronics and Electrical Engineering, University of Surrey, Guildford GU2 5XH, United Kingdom

Received August 10, 1987, revised April 19, 1988

We present a comprehensive review of the Hough transform, HT, in image processing and computer vision. It has long been recognized as a technique of almost unique promise for shape and motion analysis in images containing noisy, missing, and extraneous data but its adoption has been slow due to its computational and storage complexity and the lack of a detailed understanding of its properties. However, in recent years much progress has been made in these areas. In this review we discuss ideas for the efficient implementation of the HT and present results on the analytic and empirical performance of various methods. We also report the relationship of Hough methods and other transforms and consider applications in which the HT has been used. It seems likely that the HT will be an increasingly used technique and we hope that this survey will provide a useful guide to quickly acquaint researchers with the main literature in this research area.

1. INTRODUCTION

The Hough transform, HT, was first introduced as a method of detecting complex patterns of points in binary image data [1]. It achieves this by determining specific values of parameters which characterize these patterns. Spatially extended patterns are transformed so that they produce spatially compact features in a space of possible parameter values. The HT converts a difficult global detection problem in image space into a more easily solved local peak detection problem in a parameter space.

The key ideas of the method can be illustrated by considering identifying sets of colinear points in an image. A set of image points (x, y) which lie on a straight line can be defined by a relation, f, such that

$$f((\hat{m}, \hat{c}), (x, y)) = y - \hat{m}x - \hat{c} = 0, \tag{1}$$

where m and c are two parameters, the slope and intercept, which characterize the line. Equation (1) maps each value of the parameter combination $(\hat{m}, \hat{c})$ to a set of image points. We have used the hat symbol to denote quantities in the domain of the mapping. The mapping is one to many from the space of possible parameter values to the space of image points. The HT uses the idea that Eq. (1) can be viewed as a mutual constraint between image points and parameter points and therefore it can be interpreted as defining a one to many mapping from an image point to a set of possible parameter values. This corresponds to calculating the parameters of all straight lines which belong to the set that pass through a given image point $(\hat{x}, \hat{y})$. We call this operation backprojection of the image point and the defining relation which achieves this is given by

$$g((\hat{x}, \hat{y}), (m, c)) = \hat{y} - \hat{x}m - c = 0. \tag{2}$$

0734-189X/88 $3.00

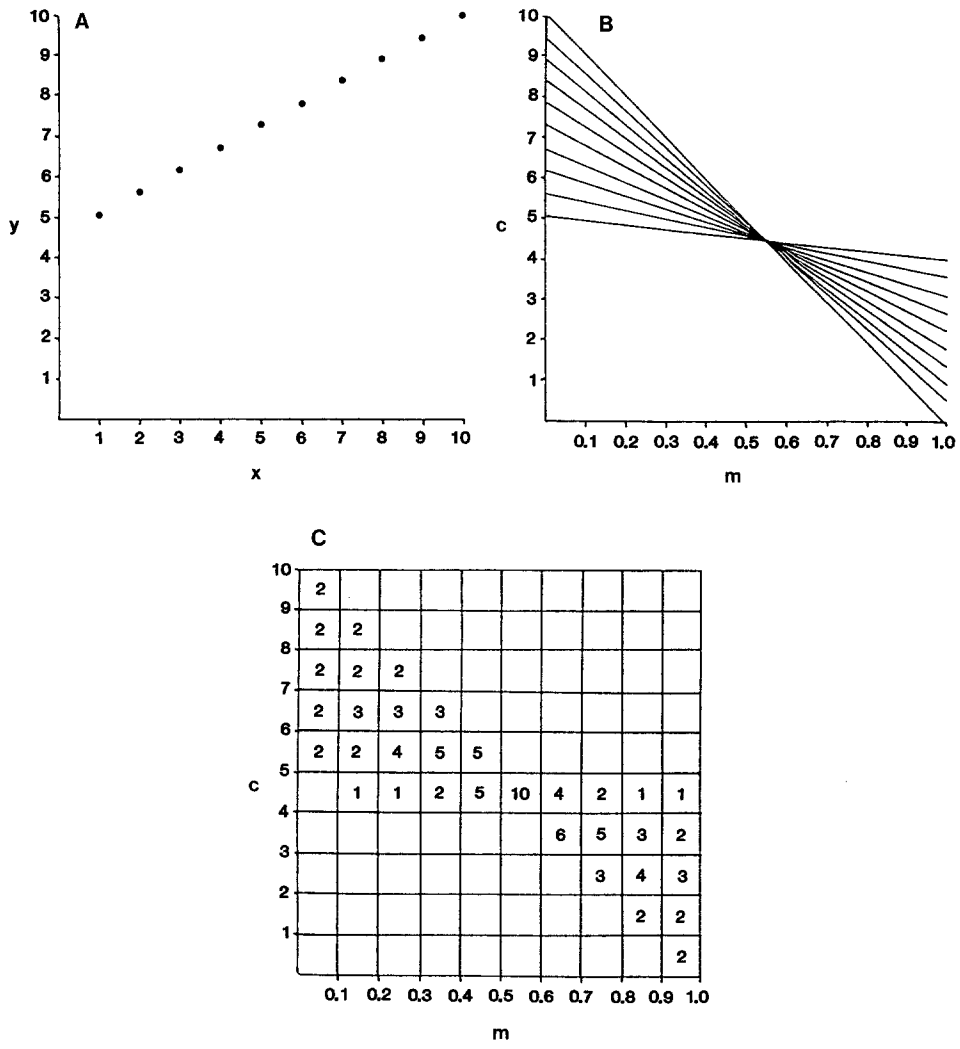

FIG. 1. The basis of the Hough transform for line detection: (A) (x, y) point image space; (B) (m, c) parameter space; (C) accumulator space corresponding to (B).

In the case of a straight line each image point $(\hat{x}, \hat{y})$ backprojects or defines a straight line in (m, c) parameter space. Figure 1a is a typical point image and Fig. 1b shows the parameter lines produced by backprojecting image points into parameter space using Eq. (2). Points which are colinear in image space all intersect at a common point in parameter space and the coordinates of this parameter point characterizes the straight line connecting the image points. The HT identifies these points of intersection in parameter space. Determination of the point of intersection in parameter space is a local operation and should be considerably easier than detecting extended point patterns in image space.

The extension of the method to detect parametrically defined image curves other than straight lines is straightforward. Image points on a curve characterized by n

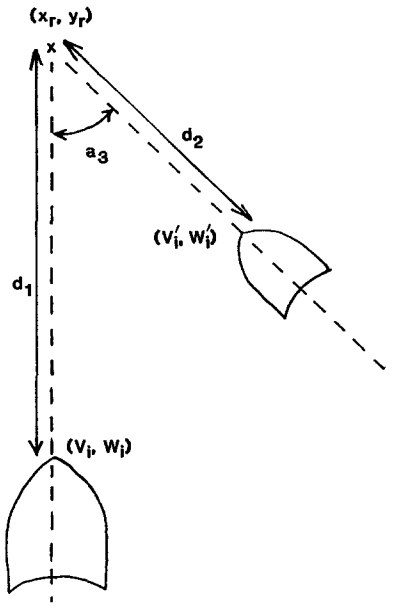

FIG. 2. Scaling and rotation of template about (x_r, y_r); a_4 is the ratio of length d_1 and d_2 or any other corresponding lengths.

parameters, $a_1, \ldots, a_n$, can be defined by an equation of the form

$$f((\hat{a}_1, \ldots, \hat{a}_n), (x, y)) = 0. \tag{3}$$

By swapping the roles of parameters and variables, Eq. (3) can be used to give the defining relation for the backprojection mapping of image points to parameter space, i.e.,

$$g((\hat{x}, \hat{y}), (a_1, \ldots, a_2)) = 0. \tag{4}$$

This backprojection equation maps out a hypersurface in the n-dimensional parameter space. The most probable parameters for image curves are indicated by the intersection of several of these hypersurfaces.

The method can be extended to detect arbitrary shapes. In the generalized HT the shape is represented by a basic template which is a list of boundary points $\{(v_i, w_i)\}$, $i = 1, \ldots, N$. This list is a particular instance of a whole class of boundaries which belong to the same shape class. Other examples of the class can be generated from this example by translating, rotating, or scaling the coordinates in the basic list. Figure 2 illustrates how an arbitrary scaling and rotation of the template about some arbitrary point (x_r, y_r) produces a new template whose boundary points are given by

$$v_i' = x_r + (y_r - w_i)a_4\sin(a_3) - (x_r - v_i)a_4\cos(a_3) \tag{5}$$

$$w_i' = y_r - (x_r - v_i)a_4\sin(a_3) - (y_r - w_i)a_4\cos(a_3); \tag{6}$$

a_3 and a_4 are parameters which describe the transformation between the two templates. Translation relative to coordinates (x_r, y_r) can be expressed by considering two further parameters (a_1, a_2), i.e.,

$$v_i'' = v_i' - a_1 = x_r + (y_r - w_i)a_4\sin(a_3) - (x_r - v_i)a_4\cos(a_3) - a_1 \quad (7)$$

$$w_i'' = w_i' - a_2 = y_r - (x_r - v_i)a_4\sin(a_3) - (y_r - w_i)a_4\cos(a_3) - a_2. \quad (8)$$

If an image point (x_j, y_j) is to be part of the sought-for shape then $x_j - v_i'' = 0$ and $y_i - w_i'' = 0$ for some point i on the template shape. The HT involves considering every combination of image point and template point and then using derivations of Eqs. (7) and (8) to calculate parameter combinations $a_1, \ldots, a_4$ which transform between them. For a specified scale and orientation, i.e., fixed values of a_3 and a_4, the backprojection equations allow all values of (a_1, a_2) to be predicted, i.e.,

$$a_1 = x_j + x_r - v_r' = x_j - (y_r - w_i)a_4\sin(a_3) + (x_r - v_i)a_4\cos(a_3) \quad (9)$$

$$a_2 = y_j + y_r - w_i' = y_j + (x_r - v_i)a_4\sin(a_3) + (y_r - w_i)a_4\cos(a_3). \quad (10)$$

Each image point when backprojected into parameter space produces a parameter hypersurface. The number of hypersurfaces which intersect at a common point in parameter space indicates the number of image points which lie on the template shape described by these parameters.

For a given analytic shape or shape template $\{(v_i, w_i)\}$, the HT of a set of image points $\{x_j, y_j\}$, $j = 1, \ldots, L$ can be stated formally as follows: let $g(\hat{x}_j, \hat{y}_j, a_1, \ldots, a_n)$ be the defining relation for backprojecting an image point into parameter space, i.e.,

$$g = \begin{cases} f(\hat{x}_j, \hat{y}_j, a_1, \ldots, a_n) = 0 & \text{analytic shape} \\ \prod_{i=1}^{N} \left\{ \left[a_1 - (\hat{x}_j + \hat{x}_r - v_r') \right]^2 \right. & \\ \qquad \left. + \left[a_2 - (\hat{y}_j + \hat{y}_r - w_i') \right]^2 \right\} = 0 & \text{template shape.} \end{cases} \quad (11)$$

Then the HT, $H(a_1, \ldots, a_n)$ is defined as

$$H(a_1, \ldots, a_n) = \sum_{j=1}^{L} h(\hat{x}_j, \hat{y}_j, a_1, \ldots, a_n), \quad (12)$$

where

$$h(\hat{x}_j, \hat{y}_j, a_1, \ldots, a_n) = \begin{cases} 1 & \text{if } g(\hat{x}_j, \hat{y}_j, a_1, \ldots, a_n) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

For the purpose of calculating the HT on a digital computer the continuous parameter space is usually considered to be composed of the union of a number of finite-sized regions. In standard implementations the space is partitioned into

suitably sized multidimensional rectangles and each rectangle is associated with an element of a multidimensional array. The elements of the array act as counters and are incremented when a hypersurface from the backprojection of an image point passes through the region of parameter space associated with the element. The array is called the accumulator array. When several image points backproject to the same parameter combinations, i.e., their hypersurfaces intersect or pass close to one another, then the corresponding array element accumulates a large value. Figure 1c shows the accumulator array after backprojecting all the image points in Fig. 1a using the equations for straight line detection. A peak of counts is obvious where several hypersurfaces intersect. Hypersurface intersection detection is therefore approximated by detection of local peaks in the number of accumulator counts. Detection of peaks in parameter space is much easier and computationally less demanding than matching spatially extended patterns of points in image space. The size of the peak is a function of several factors such as the number of points composing the line, the number of other points in the image and the choice of parameter cell size. The size of parameter cells is usually chosen to correspond to the required precision of parameter estimation.

The HT can be viewed as an evidence gathering procedure. Each image point votes for all parameter combinations that could have produced it, if it were part of the sought-after shape. The votes are counted in the accumulator array and the final totals indicate the relative likelihood of shapes described by parameters within the corresponding parameter cell.

In our presentation of the generalized HT technique we frequently referred to templates and the HT is obviously closely related to template-matching techniques. One obvious difference is that template matching is carried out entirely in the image domain. It involves generating from a basic template all other templates and then determining how many of these template points match image points. In most cases a corresponding image point does not exist and the effort in calculating this template point is irrelevant to calculating the degree of match between model and image. In contrast, the HT always assumes a match between a given basic template point and a selected image point and then calculates the transformation parameters which connect them. Thus although the HT and template matching calculate the same quantity, the HT is more efficient as it does not involve generation of inessential data.

The HT method has many desirable features. First, each image point is treated independently and therefore the method can be implemented using more than one processing unit; i.e., parallel processing of all points is possible. This makes it an algorithm suitable for real-time applications and a possible module for shape detection in biological systems. Second, its independent combination of evidence means that it can recognize partial or slightly deformed shapes. Occlusion is a severe problem for most other shape detection techniques but the HT degrades gracefully because to first order the size of a parameter peak is directly proportional to the number of matching boundary and template points. The size and spatial localization of the peak provides a measure of similarity of shape and model. Third, the HT method is very robust to the addition of random data produced by poor image segmentation. Random image points are very unlikely to contribute coherently to a single bin of the accumulator and therefore produce only a very low level background of counts in the array. A more serious problem than random data is data

from the boundaries of shapes other than those searched for. These can produce structured backgrounds and some care must be taken to eliminate or identify such situations. Finally, the HT can simultaneously accumulate evidence for several examples of a particular shape class occurring in the same image. Generally each instance of the shape simply produces a distinct peak or cluster in the accumulator array.

The principal disadvantage of the standard implementation of the HT is its large storage and computational requirements. The determination of q parameters each resolved into α intervals requires an accumulator of α^q elements. This can be prohibitively large if either α or q is large. The major computational cost of the algorithm is the calculation of parameter cell-parameter surface intersections. In the simplest case the parameter surface spans $(q - 1)$ of the q parameter dimensions. The number of calculations is therefore very large and scales exponentially with the dimensionality of the problem. The efficiency of the HT can be increased by devising methods which use small size accumulators or which use extra data to restrict the range of parameters which need to be addressed.

In this paper we offer a comprehensive review of the HT up to early 1988. Previous reviews are either out of date [17] or rather limited in scope [73]. In Section 2 we give a brief survey of how the HT developed historically up to the time of the development of the generalized HT method by Ballard. We then discuss specific aspects of the HT in greater detail. Section 3 deals with parameterizations which have been proposed for shapes and describes how complex transforms can often be decomposed into more tractable problems. Section 4 deals with efficient ways for accumulating the HT while Section 5 discusses ways of analyzing the accumulator to detect significant local peaks or other structures. Section 6 reviews the analytic results relating to the HT and Section 7 shows its relationship to the Radon transform and projection techniques. Applications are discussed in Section 8 and the suitability of parallel computer architectures is considered in Section 9. Section 10 offers a few simple conclusions.

2. EARLY DEVELOPMENT OF THE HT

The HT was introduced by Paul Hough [1] in a patent filed in 1962. It was used to detect curves in bubble chamber photographs [2, 4] and was brought to the attention of the mainstream image processing community by Rosenfeld [3]. In an influential paper Duda and Hart [5] suggested that straight lines might be most usefully parameterized by the length, ρ, and orientation, θ, of the normal vector to the line from the image origin, viz.

$$\rho = x \cos(\theta) + y \sin(\theta). \tag{14}$$

This has distinct advantages over the (m, c) parameterization which has a singularity for lines with large slopes, i.e., $m \to \infty$. Using the (ρ, θ) parameters means that image points map into sinusoidal curves in a two-parameter space. O'Gorman and Clowes [6] used Hough techniques to recover straight lines in digitized images of scenes consisting of polyhedral blocks. During the 1970's Shapiro published numerous papers [7, 10–12, 14, 17–22, 26, 28] on the analysis of properties of the HT method, particularly in relation to issues such as the choice of quantization of the parameter space and the effect of noisy input data on parameter accuracy. Cohen

and Toussaint [13] studied the distribution of background counts for random noise points in finite sized images and Van Veen and Groen [38] considered how detection was affected by discretization effects in both the image and parameter spaces.

Several authors considered using the HT for shapes other than straight lines. Kimme, Ballard, and Sklansky [8] showed how circular arcs could be detected and applied the technique to a medical image processing task. Essential to their method was the use of edge direction information to constrain the range of parameters which had to be addressed. Parabolas and ellipses have been investigated by several authors [16, 24, 59]. Usually the dimensionality of the parameter spaces are kept low by determining groups of parameters in several sequential stages. Merlin and Farber [9] showed how the HT could be generalized to detect an arbitrary shape at a given orientation and a given scale. Stockman and Agrawala [15] demonstrated that the HT method was simply an efficient implementation of template matching. The same point was made by Sklansky [23], but, in addition, he noted the HT was potentially more useful as it is a method which can easily incorporate local constraints to improve the efficiency of matching. Ballard [29, 31] eventually extended the work of Merlin and Farber and showed how a generalized HT could be formulated which could efficiently find arbitrary shapes for any orientation or any scale. The key to the success of his formulation was the use of directional edge information.

3. SHAPE PARAMETERIZATIONS

The straight line is the simplest parametrically described shape and several schemes, other than the previously mentioned (m, c) and (ρ, θ) methods, have been suggested. Significant image feature points are often determined by thresholding the output of mask-based, edge detection operators, i.e., Sobel, Prewitt, etc. In these cases estimates of the line gradient, m, are provided directly by the measurement of the local gray-level gradient components, g_x and g_y. These measurements permit the one-to-many mapping to be restricted to a one-to-one mapping of image point to a parameter point; i.e., they uniquely define the image line that the image point lies on. Davies [86] has suggested that this unique line can be parameterized by its point of intersection with a normal vector from the image origin. He calls this the "foot of the normal" parameterization. It produces a parameter space which is congruent with the image space. Davies analyzed the error in parameter estimation of this method as a function of the accuracy of edge gradient estimates and showed that the error increases as the distance of the foot of the normal from the origin increases. He therefore suggests that the method is best applied to small subimages. At very small distances the method also becomes unreliable and therefore it may be necessary to accumulate the foot of the normal with respect to more than one origin in order to get an accurate estimate. Despite these problems Davies reports line location to pixel accuracy and line orientation accuracy of 1–2°.

Wallace [84] introduces a new bounded parameterization for straight lines. He calls his modified Hough method the Muff transform. He parameterizes a line by the two intersection or projected intersection points with the perimeter of the image. These points can be stored as distances s_1 and s_2 along the outer edge of the image. Distance is measured in a counterclockwise direction using the lower left corner of the image as an origin. Uniqueness of the representation can be assured by demanding that $s_1 < s_2$. All possible (s_1, s_2) values which pass through a given image point (x, y) can be calculated without recourse to transcendental functions.

Wallace suggests that one of the features of the method is that the resolution of the lines represented is uniform and matches exactly the set of lines which can be drawn across the image using digital graphics techniques.

Forman [90] experimented with a hybrid of the (ρ, θ) and Muff representations. He parameterized lines by the coordinates where they first leave the image, i.e., s_1 and a measured orientation θ. The algorithm can be made efficient by employing a local image operator which searches for two close line elements. Forman uses his method to extract linear roads from FLIR images.

Casasent and Krishnapuram [106, 120] have suggested a method that uses the straight line (ρ, θ) HT to detect curves. It is based on the idea that a curve may be represented as a succession of short straight line segments. Each segment will map to a distinct cell of the (ρ, θ) space with a value proportional to its length. The accumulator corresponding to a curved shape will consist of a characteristic pattern of small peaks. Random background peaks are eliminated by semi-thresholding and then shape specific transformations which involve the shifting of bins of the accumulator can be applied. Accumulator transformations which result in all the characteristic peaks lying on a common sinusoid identify the parameters and orientation of the sought-after curve. An inverse straight line transform of the modified accumulator will then produce a peak in the inverse space at a location corresponding to the position of the curve in the original image space. Casasent and Krishnapuram give simple methods for determining the shape specific transforms which are applied to the accumulator and illustrate the detection of circles, ellipses, and parabolas.

High-dimensional parameterizations can often be decomposed into smaller sets of parameters which can be determined sequentially. This idea is extremely beneficial in terms of storage required for the problem and the amount of calculation required. For example, circles are usually described by 3 parameters, their center coordinates (a, b), and the circle radius r. However, if we use extra information concerning the edge direction of image features then the circle detection problem can be decomposed into a two-stage process which involves a 2D HT to find estimates of the center parameters, (a, b) [118], followed by a trivial 1D HT, or histogramming step, to find the best value of the circle radius. Similar ideas have been used in the detection of ellipses [24, 59]. Ellipses are described by 5 parameters but by using relationships between image features it is possible to decompose this to a series of problems whose maximum dimensionality is 2.

Many theories of vision assume that descriptions of objects are stored in object-centered representations. Object recognition involves determining a transformation which maps a 2D view of an object into a 3D object-centered description. This viewing transformation can be described by 7 parameters which cater for arbitrary changes in scale, orientation, and translation. However, Ballard and Sabbah [46] show that there is a natural ordering or dominance of these parameters such that they can be sequentially determined in subgroups corresponding to scale, orientation, and translation. They give specific algorithms, based on the extensions of the generalized HT to 3D, which determine these parameters for objects described by oriented edges and planar surface patches.

In discussing the generalized version of the HT we have described shapes by a template, specified as a list of (v, w) points, and an arbitrary reference point (x_r, y_r). An alternative representation, used by Ballard [31] in his exposition of the

method, is to store each boundary point as a vector relative to the reference point, i.e., distance, r, and direction, α, of the line connecting boundary and reference point. Ballard organized this list of points so that entries are indexed by the local edge direction at the boundary point. The resulting list is called an R-table. Indexing by edge direction decreases the computational expense of the method as it means that image points are only compared with a subset of R-table entries, i.e., those boundary points which have similar edge direction.

Complex shapes can often be decomposed into simpler shapes. It is therefore attractive to have a shape representation which can accommodate this type of structural description. Ballard [31] discusses how the R-table of a composite shape can be constructed from the R-tables associated with simpler primitives. Davis [41] suggests that a structured hierarchical approach is of greater utility. His approach allows better control over the shape recognition process in so far as it permits the system to demand that specific primitive elements of the shape are detected before confirming the detection of the composite.

The major computational effort of the HT is in determining which cells of the parameter space are intersected by the surfaces generated by image features. Although the general hypersurface–hypercell problem is very difficult there are several well-defined and efficient methods for calculating whether n-dimensional planes intersect n-dimensional rectangles. Li, Lavin, and LeMaster [75] have therefore suggested that shapes should be parameterized so that they produce hyperplanar surfaces in the corresponding parameter spaces. They call this the "hyperplane formulation" of the HT. Straight line detection in 2D and plane detection in 3D fall naturally within this formalism and experiments with 3-parameter circle finding have been made [119]. Circle center finding also involves intersection of straight lines in a 2D parameter space [118].

Kasif, Kitchen, and Rosenfeld [54] have shown that shape detection using the generalized HT, GHT, is similar to the problem of subgraph isomorphism, i.e., the matching of two relational graph structures. A node x in a prototype labelled graph can be described with respect to an arbitrary node r by the sequence of arc and node labels that appear on an arbitrary path between x and r. A list of such relational sequences describes the structure of the graph relative to the reference node x and therefore can be used to predict or vote for possible reference nodes as a second graph is investigated.

Wahl and Biland [100] have suggested that shapes can be represented as distributed patterns of peaks in parameter space. They considered the problem of recognition of straight-edged polyhedra in a single 2D image. They transformed the image to a (m, c) parameter space representation and then interpreted the pattern of peaks in this domain. They use simple rules concerning the relationship of parameter clusters in the (m, c) space. For example, vertically aligned clusters in parameter space are indicative of parallel lines in image space. A colinear arrangement of n clusters in parameter space is evidence for a n-line vertex in image space. Wahl and Biland have investigated general principles for deciding on the best order in which to apply some of these rules.

4. EFFICIENT HT ACCUMULATION

In this section we consider several methods which utilize novel data structures or use techniques that employ nonuniform or multiple resolutions in order to achieve

storage and computation savings. Many schemes are based on the observation that it is necessary to have high accumulator resolution only in places where a high density of votes accumulate. O'Rourke and Sloan [35, 37, 69] were among the first to use this principle. They developed two data structures. The first structure, *dynamically quantized spaces* (DQS) [35, 69], consists of a binary tree in which each node encloses a rectangular region of space. As data is accumulated rules are used to split and merge cells so that each cell contains, irrespective of its size, approximately the same number of counts evenly distributed throughout its volume. The total number of cells in the tree is an input parameter of the method, but in the final tree there will be a concentration of small cells around peaks and this allows peaks to be located more accurately than using the same number of cells in a standard uniform quantization scheme. The second data structure, *dynamically quantized pyramid* (DQP) [37, 69], is based on a multiresolution, multidimensional quadtree in which the number and connectivity of cells is small and fixed. A disadvantage of the quadtree structure is that its boundaries, and hence its spatial resolutions, are fixed. This limitation is overcome in the DQP data structure using a dynamic technique, called hierarchical warping, to modify the boundaries of cells by tracking the mean position of data points in their regions of space. The net effect of the boundary adjustment is to produce cells containing approximately equal numbers of votes. However, the warping process introduces some errors in the final accumulator totals and means that the final division of space depends on the order in which the data is accumulated; i.e., recently accumulated points are most influential on the final result. O'Rourke and Sloan use both data structures in a series of experiments where a 3-parameter space is divided into about 64 cells. They demonstrate the relative abilities of each algorithm to focus on peaks, to dynamically change attention, and to detect multiple peaks. Their general conclusion is that DQSs are difficult to implement but perform somewhat better than DQPs. However, Blanford [102] has recently reconsidered DQPs and modified the algorithm so that votes are more correctly distributed throughout the quadtree. This rebalancing strategy requires about three times the number of computations of the original DQP algorithm but it ensures that the result is less dependent on the order of data presentation.

A useful idea suggested by several authors is the iterative focusing of the HT to identify peaks with high counts. Initially the HT can be accumulated in a coarse but uniform resolution parameter space. Regions or cells with high counts at this resolution can then be investigated at higher resolution. Only a few iterations of this procedure can lead to very high parameter resolution. Consider searching for a peak in a q-dimensional space resolved into α accumulator bins. The iterative use of a much smaller accumulator of size β requires $O(\log(\alpha)/\log(\beta))$ iterations to focus down to the same resolution. The computational saving of the method is proportional to the ratio of the number of cells in the two accumulators, i.e., $(\alpha/\beta)^q$ times the ratio of iterations, i.e., $O((\alpha/\beta)^q * \log(\beta)/\log(\alpha))$. This can be very large. This strategy provides enormous efficiency benefits in terms of both the amount of computation and the amount of storage space required. It has been used by Adiv [43] as an efficient way to search for motion parameters in high-dimensional spaces and by Silberberg [72] to identify the viewing parameters relating stored models to observed shapes.

Recently, Li *et al.* [75, 76, 94] have detailed a focusing algorithm, the fast Hough transform, FHT, which uses a multidimensional quadtree in conjunction with HTs

which map image points into hyperplanes. This use of hyperplanes is advantageous as the approximate intersection between planes and parameter cells can be efficiently computed using an incremental test which depend on the known spatial relationship between a quadrant and its four sons. Only additions and shifts are required to implement the test and the computational cost scales only linearly with the dimensionality of the parameter space. Computational gains of $O(10^3)$ for 2D lines and $O(10^6)$ for 3D plane detection can be achieved. The algorithm is regular in structure and is therefore easily extended to higher dimensional spaces and easily implemeted in VLSI hardware. However, the FHT currently identifies areas for focusing by requiring that quadrants have a vote count which exceeds a fixed threshold. In complex images this threshold may be difficult to determine and if it is set too low the efficiency of the method will deteriorate as the algorithm will explore too many quadrants. Another problem is related to the incremental intersection testing method. This requires that each quadrant must store the distance of every image feature from its center. If there are many image features this can represent a large overhead. It is also not clear that all shapes can be naturally mapped into hyperplanes. Finally, the quadtree data structure is a static structure and cannot optimally adapt to situations which may require different parameter resolutions along different parameter axes. Li *et al.* [93] have recognized the last problem and have suggested using a data structure, called the bintree, in which space is partitioned into rectangular regions.

Illingworth and Kittler [118, 119] have developed an iterative coarse-to-fine search strategy for detecting lines and circles in 2D and 3D parameter spaces. Their implementation is called the adaptive Hough transform, AHT. It uses a small accumulator array (typical $\beta = 9$) which is thresholded and then analyzed by a connected components algorithm. The shape and extent of connected components determine the parameter limits which are used in the next iteration. Limits can be decreased, expanded, rotated, or merely translated depending on the distribution of counts in the accumulator. Parameter limits are defined independently for each parameter dimension and therefore the method selects appropriate precision for each parameter. The accumulator storage gain of the method is of the order expected for focusing methods, i.e., $(\alpha/\beta)^q$, and the method works well for single object detection. A simple example [119] involving searching for circles in a 3-parameter space demonstrated that it was several hundred times faster than the standard method. However, serious interpretation problems have been found when the method is applied to complex images. In these cases the method is led astray at coarse resolutions where extended patterns of votes from two or more objects can overlap and produce a spurious peak whose position is unrelated to the parameters of any true feature in the image. Solutions for these difficulties are currently being investigated.

Brown [40, 47, 49] attempted to overcome the storage problems associated with the standard HT by replacing the accumulator array by a small fixed size content addressable store, i.e., a hash table in software or a cache if implemented in hardware. Counts are accumulated in the store until it is full and then a flushing or garbage collection method is invoked to release some storage for further use. The simplest flushing strategy is to remove entries with the fewest votes. However, it is possible to devise schemes which will favor the keeping of recent information or which will incorporate geometric locality information. Brown studied caches of size

32, 64, and 128 and compared their performance against standard accumulators as he varied flushing strategy and the order of accumulation of the data. He concluded that finite length caches were as practical as accumulators and he was able to predict qualitatively the degradation of their performance as noise increased and cache length decreased.

David and Yam [27] overcome some of the problems associated with the storage requirements of the generalized HT by investigating schemes which compute only a projection of the full HT. They found the GHT was still a robust shape matcher and they discused ways of recovering the parameter values lost due to computing projections.

5. PEAK DETECTION IN HT

Once the HT has been accumulated the pattern of counts in the accumulator has to be analyzed to estimate the presence and location of local peaks. The most common method is to determine a global threshold. Any accumulator cell with more counts than the threshold indicates a possible instance of the searched for shape. The threshold is chosen either using prior knowledge or it can be automatically selected by analyzing the distribution of counts in the array, e.g., a fixed fraction of the maximum count in any single accumulator bin.

O'Gorman and Sanderson [68] have suggested an algorithm called the converging squares algorithm as a computationally efficient way of detecting peaks in multidimensional data. It is based on a resolution pyramid data structure and involves sequential investigation of the data from a low to a high resolution. Several overlapping hypercube shaped regions of the accumulator are compared and the highest density region is investigated in greater detail. The method is robust with respect to noisy data and was found to be 8–10 times faster than some commonly used peak detection methods. However, as it is based on comparing cubic regions it may not give the most desirable results when applied to accumulators which contain elongated or nonconvex peaks. Brown [49] has also looked at a pyramid-based approach to the detection of peaks in n-dimensional histograms.

Several authors have suggested ways of sharpening peaks in parameter space and thereby easing parameter space interpretation. Many authors suggest using weighting factors so that the most prominent or most certain image features contribute more to accumulator cells than less certain data. Most of these schemes have little formal justification and they usually involve a weighting factor derived from edge gradient measurements [31, 38]. One interesting variant has been suggested by Thrift and Dunn [58] who propose a HT method in which each image feature increments every parameter combination. The value which is contributed to each accumulator bin is determined by a heuristic function, called an influence function, which depends inversely on the distance between the parameter point and the parameter hypersurface generated by the image feature. The method has intimate connections with the more common ideas of fitting of curves to a set of data points. Thrift and Dunn report that the method is superior to standard HT on noise perturbed examples of simple shapes.

Leavers and Boyce [92, 121, 122] suggest enhancing peaks in the accumulator array by filtering the array. They consider the (ρ, θ) parameterization of lines and analyze the expected shape of the distribution of counts in the accumulator. For an image line the shape is a characteristic "butterfly" and they suggest suitable

coefficients for a convolution mask which will enhance or detect these peaks. They have also predicted the distribution of counts produced in the (ρ, θ) accumulator by a circle. This yields a broad band of counts which has a characteristic falloff shape at the edge of the band. They are able to design a filter to enhance these edges and hence produce two approximately linear edges in the filtered (ρ, θ) space. These lines, and thus the circle parameters, can be located by treating the filtered parameter array as an image and performing a second line detection step.

Gerig and Klein [91, 114] use a different approach to peak sharpening. They used the HT to detect circular boundaries of cells in complex images where the boundaries are partially missing, obscured, or distorted. To analyze the accumulator array they suggested using a backtransform strategy in which each image point is assigned to a unique parameter cell. This cell is selected from those intersected by the hypersurface associated with the image point and is the cell which contains the largest number of counts. Reaccumulation of the parameter space using this unique image point to parameter cell assignment results in a strong sharpening of peaks.

Eckhardt and Maderlechner [131] have considered the effect of projecting the full HT accumulator array into smaller subspaces in the hope that interpretation will be easier in these cases. They have shown that this process will only be useful if the composite operation of HT accumulation and projection has certain properties. In particular, if the composite operator is linear and translation invariant then the result is trivial and useless. They are therefore led to explore projection methods which involve nonlinear operators.

6. ANALYSIS OF HT PERFORMANCE AND PARAMETERS

Early analytic work on the properties and performance of the HT concerned the effect of statistical measurement error on the position and localization of parameter peaks. Shapiro [10–12, 14, 17–22, 26] published much on the variance of parameter estimates as a function of measurement error for transforms in which each image or feature point produced a single vote in parameter space. Sklansky [23] suggested a geometric construction for straight line detection which could be used to investigate the precision of curves derived from estimated parameters. This graphical technique was extended by Shapiro and Iannino [26] to the case of noisy image measurements and was used to derive results relating quantization errors and the accuracy of parameter estimation. These provided useful guides for determining accumulator quantization.

Cohen and Toussaint [13] were the first authors to consider the density of counts produced in parameter space by a uniform distribution of image points scattered over a finite size image or retina. They found that the finite extent of the retina produces a distribution of counts which is peaked independent of the presence of a shape. The effect can be understood by considering the relative numbers of image points in lines which pass close to the center of a circular image and those which cut the edge of the circular image. The former lines are long and populate the low ρ regions of parameter space while the shorter lines at the edge of the image provide far fewer counts to the high ρ regions of the parameter space. This means that a ρ independent globally derived threshold is unsuitable for peak detection. Cohen and Toussaint suggested that these effects be overcome by either an empirical background subtraction method or by using a nonlinear quantization of the ρ parameter axis so that each bin of the accumulator contains equal counts from random noise.

The general case of detecting m-dimensional lines in an n-dimensional space and the problems of parameter quantization were investigated by Alagar and Thiel [30]. They emphasized that Cohen and Toussaint's formula for nonuniform quantization of the ρ parameter axis only produces a uniform distribution of counts for each value of θ. They showed that in two-dimensional (ρ, θ) parameter space the quantity $(d\rho, d\theta)$ was the appropriate invariant to consider if we want a uniform distribution of counts in image space to produce a uniform density of counts in parameter space. Analysis using their invariant leads to a parameter space quantization which is related to a beta distribution.

Alagar and Thiel [30] suggested that the performance of line detection methods might be judged by the complexity of the algorithm and by its effectiveness or robustness. They suggested three measures of robustness called precision P, recall R, and goodness G. If an image space is known to contain M_k lines and a given algorithm detects m_k lines of which m_T are identified with true lines present in the image and m_F are false detected lines then

$$\begin{aligned} P &= m_T/m_k \\ R &= m_T/M_k \\ G &= (m_T - m_F)/M_k. \end{aligned} \tag{15}$$

A suitable single measure of goodness can be derived by linearly combining these three measures. Using these measures, Alagar and Thiel present empirical results for the performance of line finding procedures which utilize different parameter space quantization schemes.

Maitre [95] is the most recent author to consider the effects of random image noise on the density of counts in parameter space. For the case of a circular image he derives the same formula for quantization of (ρ, θ) space as Alagar and Thiel, but using a different method. He also considers the density of counts for detecting lines in a (m, c) space and presents a formula for cases involving rectangular images. Using a signal processing approach he derives some results concerning the signal-to-noise ratio for line detection as a function of the reliability and precision of the linear feature extractor.

Several authors have investigated theoretically the distribution of counts in a finite accumulator due to the backprojection of real, nonrandom sets of image points. Once again, most attention has been on straight line detection. Van Veen and Groen [38] studied straight line detection using the (ρ, θ) parameterization and derived analytic results for the shape and extent of parameter peaks produced by true lines as a function of the quantization of the image space, the quantization of the parameter space, and the width of line segments. They suggested that peaks could be sharpened by weighting the number of accumulator counts contributed by a point by a factor related to the edge gradient at the point.

Brown [48] recast the HT as a linear imaging problem. In this view the pattern of parameter space counts generated by each image point is called the feature point spread function, feature psf. The final accumulator for a single object is a superposition of the feature psfs for each of its constituent image points and is called the parameter point spread function, parameter psf. The parameter psf consists of a central peak surrounded by a background or sidelobe distribution. For continuous,

noiseless conditions the parameter psf of a shape is just the image of the shape in a pinhole like camera whose pinhole aperture is the shape itself. The operation carried out by the pinhole camera is autoconvolution. Brown uses this model to compute the parameter psf of different shapes and finds that for many shapes, such as simple lines, they are extended ridges rather than sharp symmetric peaks. This has important consequences for peak-finding strategies. He also modelled images containing several objects and spurious points and considered how the detection of peaks was affected by the mutual interference of their parameter and feature psfs. He was led to propose the complementary HT, CHough. In this method image points not only contribute positively to some parameter values but they also contribute negatively to other parameter values. The proposed net effect of this procedure is to cancel out some of the effects of sidelobes and produce more prominent peaks. It is of greatest use in transforms which produce symmetric parameter psfs. Brown found that in comparison to the standard HT method the CHough method reduced the mean and the variance of the background and was more robust to spurious image features. It did, however, suffer more from quantization effects.

Davies [111] has used ideas similar to the CHough in order to improve the accuracy with which the GHT can be used to longitudinally localize lines. The basic idea is that if the length of a line is known then detection of an edge point provides evidence both for possible locations of the line but also it provides evidence concerning positions where the line cannot occur. Davies accumulates positive and negative evidence in separate parameter spaces. The positive evidence space provides the best measure for line detection but line localization is improved by looking at the signal represented by the difference of the positive and negative evidence spaces.

Leavers and Boyce [92, 103, 121] are the most recent authors to analyze the expected shape of peaks in the Hough parameter space. They discuss the expected distributions for the projection of linear and circular image features into the (ρ, θ) space. They find that lines produce a butterfly shaped peak while circles produce a broad band with a characteristic falloff at the band edges. As previously mentioned this information can be used to produce digital filters which will enhance these peaks.

Davies [110] has considered the relationship between the HT and matched filtering and within this framework has suggested that votes should be weighted in parameter space by an amount which is proportional to their edge gradient magnitude and to their a priori edge gradient. In the same paper he clearly emphasizes the distinction between the accuracy of object location and the sensitivity of object detection. He points out that sensitivity of object detection is often sacrificed in order to decrease computational complexity.

Hunt *et al.* [116, 117] have recently presented a very interesting comparison of line detection using the HT and line detection using a statistical signal detection theory approach. They consider gray level images corrupted by various types of statistically independent, additive noise: Gaussian, Laplacian, and uniform distributions. Unlike the HT, the signal detection methods explicitly take into account line length, the a priori distribution of lines in the image, and the noise distribution. Hunt *et al.* present results for both line detection and line location performance. In both respects it is possible to devise a signal detection method which is superior to

the HT but if the noise characteristics are not well known then the HT is useful as its performance is insensitive to these details. The results presented by Hunt *et al.* concern the detection of a single line and it is noted that the superior performance of the signal detection methods has to be considered in the light of the increased computation which they involve.

7. THE HT AND OTHER TRANSFORMS

In 1981 Deans [32] pointed out that the Hough transform for linear features was a special case of the Radon transform. The Radon transform has been known since 1917 and therefore has a richer mathematical literature associated with it. It has been much studied in relation to computer-aided tomography. The Radon transform of a function $f(x, y)$ on a two-dimensional Euclidean plane is defined as

$$R(\rho, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\, \delta(\rho - x\cos(\theta) - y\sin(\theta))\, dx\, dy, \qquad (16)$$

where δ is the Dirac delta function. The delta function term forces integration of $f(x, y)$ along the line

$$\rho = x\cos(\theta) + y\sin(\theta). \qquad (17)$$

The Radon transform yields the projections of the function $f(x, y)$ across the image for various values of offset ρ and orientation θ. The Radon transform is equivalent to the HT when considering binary images. The Radon transform can be computed via the Fourier transform [96, 129]. This property has been used for line detection and enhancement in noisy SAR images. Suter [82] has also used this property to develop a parallel incremental algorithm for computing the digital Radon transform.

The Radon transform for shapes other than straight lines can be obtained by replacing the δ function argument by a function which forces integration of the image along contours appropriate to the shape. The equivalence of the two methods has stimulated investigation into the implementation of shape detection in real time using analog optical systems. Eichmann and Dong [52] have suggested a coherent optical processor while Stier and Shori [99] proposed an optical method which utilizes incoherent optics. Gindi and Gmitro [62] have implemented a hybrid scheme which uses both optical and digital hardware.

The link between projection data and the HT has been utilized by several other authors. Sloan [42] suggested the use of a representation called the "dot product space" to describe bloblike objects. It utilizes local shape evidence projected into a transform space where it is easier to make a global judgement. Like the HT it is robust to occlusion and incomplete boundary information.

Nagoa and Nakajima [124] have written about the variable size slit method of image analysis. This involves investigating image data by considering its projection onto the edges of a rectangular window of arbitrary size and orientation as it is moved over the image. They discuss the relationship between this and the HT method.

The idea of the HT as a projection has been used by Mostafavi and Shah [57] in conjunction with hardware features of commercially available image processing

systems to produce a rapid algorithm for line detection. The same idea has been used and extended in the recent work of Sanz and Dinstein [80, 126–128]. Lines at a single orientation, θ, can be detected by multiplying the binary image of the line with a template image. In the template image each pixel lying on a line a normal distance ρ from the origin has an intensity of ρ. The resultant histogram of gray levels should have large peaks only at values of intensity equal to the ρ value of the line. By generating and evaluating a series of histograms for each value of θ the full HT can be built up. Features of specialized image processing and general pipelined architectures can be used in the template image generation, the image convolution and the gray-level histogram generation steps. The method can be used for shapes other than lines by using different template images.

8. APPLICATIONS OF THE HT

The HT has proved a valuable method in a large range of machine vision problems. One of the major reasons is that straight lines and other simple shapes occur in most natural and man made scenes. For example, man-made objects are produced by processes which employ milling, grinding, or drilling and these produce simple linear or circular features. Even complex objects can often be identified by their distinctive combination of these simple features. Wallace [60, 130] has proposed an industrial inspection system which exploits these principles by including linear and circular feature detection modules in a hypothesis and test vision system. Arbuschi, Cantoni, and Musso [44] have suggested using the generalized HT as the basis of a system for detecting and locating mechanical parts.

Examples of the use of the straight line HT are numerous. It has been used by Inigo *et al.* [66] to find straight edges of roads and tracks. This information can then be used to guide a robot vehicle. Dyer [51] has applied the straight line HT to the inspection of the scaling accuracy of needle-type measuring instruments. The angular position of the straight pointer can be determined in a series of images taken for a specified sequence of input analog signals. The method requires no detailed knowledge of the gauge position, size, or orientation and can be implemented in high-speed hardware.

Huang *et al.* [74] detected both linear and hyperbolic features in seismograms using the HT. Time travel curves of direct and refracted seismic waves produce straight line patterns in the seismogram while reflected waves lead to hyperbolic tracks. The HT is a particularly attractive technique because of the cluttered and fragmented nature of the seismic data.

Kushnir, Abe, and Matsumoto [55, 77] studied the problem of automatic recognition of Hebrew characters. These consist almost entirely of linear strokes. Peaks detected in (ρ, θ) parameter space of a HT line detector were used as features in a pattern recognition system. For machine printed characters a recognition rate 99.6% was reported over a sample set of 18 print alphabets [55]. An extended system for hand printed character recognition achieved an average recognition rate of 86.9% [77].

Lin and Dubes [56] experimented with a straight line HT method of counting ridges in an automatic fingerprint analysis system. After image preprocessing and thresholding, an image window in which ridges appear as straight lines was selected. Each such ridge produces a distinct peak in the (ρ, θ) accumulator space and therefore ridge counting is transformed to a peak counting problem. In practice Lin

and Dubes found the HT method less successful than a conventional approach. The HT was computationally expensive and it was difficult to select a reliable threshold for peak detection.

The linear HT has been used in target detection and tracking. Cowart, Snyder, and Rueder [50] have studied frame to frame difference images in which moving but nonmanouvering targets appear as straight line segments. The HT is especially good for the common situation of having to simultaneously detect multiple targets.

Shibata and Frei [36] used the HT to detect a single rectangular target in a single image. Their method is to look for a characteristic pattern of 4 peaks in (ρ, θ) parameter space corresponding to the 4 sides of the rectangle. The final design envisaged was to be realizable in real time.

Recently Skingley and Rye [129] have reported using the HT to detect faint lines in synthetic aperture radar, SAR, images. They discuss detection of peaks in the transform space, the detection of line end points and the removal of false alarm detections. They show how the probability of detecting a line in terms of its relative length and intensity can be calculated. Using their post processing methods they achieve good line detection performance.

Shu, Li, Mancuso, and Sun [135] have used the HT to extract edge contours of resist lines in scanning electron microscope, SEM, images. They have made interesting modifications to the basic line detecting HT so as to preferentially detect connected edge segments. This is achieved by augmenting the accumulator array with another bit array. Incrementation of an accumulator cell is conditional on the status of this bit array. Shu *et al.* also use a threshold for line detection which is adaptive to the orientation and location of the line segment considered. They have proposed a systolic architecture for the implementation of their algorithm.

Thomson and Sokolowska [136] have applied the line finding HT to the analysis of cleavage cracks in minerals. Micrographs of thin sections of rock were edge detected and the HT was applied to this edge data. The HT was filtered using the methods of Leavers and Boyce [92] and peaks were identified in projections of the resulting accumulator.

Nixon [79] has considered the use of the HT to detect linear variations for illumination across images. He presents an efficient algorithm for the removal of these artifacts.

The generalized HT has been applied and extended to many practical applications. Hakalahti *et al.* [63, 78] have investigated its use for general shape recognition and have incorporated both contour curvature and local image contrast information into the method. They use a two stage algorithm which initially explores the parameter space at a coarse resolution. Davies [88] has considered the problem of detecting known size polygons using a generalized HT and has suggested ways in which symmetry assumptions permit reductions to be made in the size of accumulator used. He has also considered using the generalized HT to detect both sharp and blunted corners [87]. He discusses the choice of reference or localization point which will yield the best compromise between accuracy of corner location and sensitivity for corner detection. Costabile and Pieroni [85] used the HT to detect corresponding edges in shapes approximated by polygons. Turney, Mudge, and Volz [83] have developed a Hough-like algorithm for shape detection. It accumulates evidence for shapes by matching many overlapping boundary subtemplates. Their work includes the development of methods of determining a measure or weighting for each

subtemplate which indicates how effective it is in discriminating a given object from the set of all other objects which are likely to occur in the image. Yam and Davis [39] have experimented with the use of the HT for image registration.

The HT has been used by several authors in the determination of motion parameters from a sequence of images. One of the first attempts was by Fennema and Thompson [25] who developed a relationship between rigid object translational velocity and both the rate of change of image intensity and the local spatial gradient of the image. They used this as a constraint which could be solved by a Hough accumulation method. They found it particularly useful for images containing several independently translating objects and for images in which objects were occluded. Jayaramamurthy and Jain [53] have used a slightly different constraint to define a HT which is used as part of a method to segment textured objects moving against a textured background. Samy and Bozzo [71] have suggested a method for matching objects in image sequences which uses a HT scheme in which the accumulator is incremented by an amount dependent on the observed local velocities.

Ballard and Kimball [45] have investigated how the 9 parameters of rigid body motion, i.e., position, translational velocity, and rotational velocity, can be estimated from depth and 2D optic flow values. Both depth and optic flow can be computed by parallel, local relaxation algorithms if boundary conditions are known. These boundary conditions can be calculated from the rigid-body motion parameters. Hence the two processes are coupled; i.e., initial estimates of motion parameters lead to increased accuracy in optic flow and depth estimates which in turn allow the motion parameters to be inferred more accurately. Ballard and Kimball view the HT as a general constraint transform described by $\langle x, p, f \rangle$, where x is a space of features, p is a space of parameters, and f is a relation between elements in the two spaces. The key to the successful use of these transforms is the decomposition of high-dimensional problems into several groups of parameters which can be estimated sequentially using spaces of lower dimensionality.

Adiv [43] used the generalized HT to detect patterns of motion in a displacement field image produced by the time difference of two images. The major problem of the method apart from the computational expense was that small objects produce only small peaks in parameter space. To overcome these problems Adiv suggested detecting large objects using coarse parameter resolution and detecting small objects by partitioning the image space into small subimages. He also proposed an iterative focusing approach to parameter estimation. Adiv was able to detect rotation, expansion, and translation parameters.

Radford [98] has considered how motion parameters for translation and rotation can be mapped into a 3D parameter space. Two of his parameters are the familiar (ρ, θ) used in straight line detection. The third parameter is a length, l, which measures the motion of an image feature between successive frames. Radford uses this formulation to detect the focus of expansion of translational motion and the center of rotation for angular motion. By grouping points with similar motion parameters a scene can be segmented.

Several authors have considered the determination of viewing parameters which match features in images to stored models. We have previously mentioned the work of Ballard and Sabbah [46] who sequentially determine groups of parameters which describe scale, orientation, and translation of the model. Silberberg, Davis, and

Harwood [72] have also used the generalized HT to estimate the viewing parameters which match straight line image segments with corresponding features of a 3D model. Line segments, which were detected using a Laplacian edge finder followed by standard linking and thinning algorithms, were backprojected to all transformation parameters which could match a line segment with a suitable stored model edge. The backprojection can be made efficient by invoking heuristics such as matching image lines only to model edges if the projected edge length is greater or equal to the measured image length. Silberberg also used an iterative focusing implementation of the HT.

Peek, Mayhew, and Frisby [70] have used a Hough type of accumulator array to obtain estimates of the angle of gaze, g, and the distance to the point of focus, d, of two stereo eyes. These viewing parameters are linearly related to one another by the equation

$$g = \left(\frac{Vd}{Iy} \right) - x, \tag{18}$$

where I is the interocular distance, V is the vertical disparity of two matched points, and (x, y) is the retinal eccentricity of the points. Possible values of the viewing parameters g and d are plotted for each matched point and are recorded in an accumulator array. The bin of the array which receives the most votes represents the best global values of the viewing parameters.

Several authors have considered using range data in 3D shape detection. Henderson and Fai [64] have used the generalized HT for object detection in 3D data from a laser range-finding system. The main problem is to choose suitable distinctive feature points so that the transform is computed for only a few of the otherwise large number of matches. Henderson first detects planar segments in the data and then matches these with his models.

Muller and Mohr [67] have reported results on 3D shape detection using range data. They considered fitting quadric surfaces as well as planes to local neighborhoods of data points. The points are then accumulated in a parameter space corresponding to the surface type which they best fit. Efficient transform calculation can be achieved by a divide-and-conquer-based focusing method.

Dhome and Kasvand [89] have used the principle of the generalized HT to recognize 3D polyhedra. The polyhedra are stored as records which describe the attributes and relationship between adjacent pairs of planar faces. Segmented images provide similar records for candidate planar surfaces and they can be compared to the list of stored models to hypothesize all possible locations of objects. Dhome and Kasvand use a hierarchical procedure based on a clustering technique to identify hypotheses which occur most frequently.

Shapiro [28] has suggested the use of the HT for contour or waveform data compression. He considers a model where a curve is a concatenation of segments from a set of known shapes. Each segment will produce a cluster in an appropriate parameter space. All parameter spaces can be accumulated simultaneously and consecutive points contributing to the same parameter peak can be identified. Curves can be reconstructed from estimated segment parameters and estimated points of transition between two segment types.

Poelzleitner [97] uses a method which he claims is based on HT ideas to improve the segmentation of wooden boards prior to classification into quality classes. Images of wooden boards are segmented and symbolic labels attached to individual pixels. The value of features and symbols in the neighborhood of a pixel are used in conjunction with predefined rules to increment an accumulator which is then thresholded to provide a new cleaned-up symbolic image.

Kasif *et al.* [54] have used the generalized HT to match subgraphs derived from geographical maps. They discuss the merits of the method relative to other methods used for the subgraph isomorphism problem, i.e., backtracking search or relaxation labeling. The method can be implemented on an appropriately connected set of simple processing units.

9. ARCHITECTURES FOR THE HT

One of the main characteristics of the HT is that it consists of a series of fairly simple calculations carried out independently on every feature in an image. In this section we will review ideas to implement the HT in real-time hardware and show how some have attempted to capture the HTs inherent parallelism on specialized parallel architectures. Most of these attempts have been restricted to consideration of implementing the $\rho - \theta$ line finding HT.

Hanahara, Maruyama, and Ichiyama [132] have implemented a $\rho - \theta$ line finding HT which pipelines the ρ intersection calculation and the accumulator incrementation steps. They implemented their system in standard TTL medium and small scale integration circuits using a MC68000 as the main processor. The whole process which includes edge detection, HT calculation, accumulation, and peak detection was found to take 0.79 s for 1024 feature points.

Rhodes [134] *et al.* have used the techniques of RVLSI, restructurable VLSI, to produce a HT processor from a standard wafer containing many sum and multiply cells. In RVLSI, large area wafers are filled with standard logic cells and the wafer is tested after fabrication to determine which cells are functional. Working cells can then be flexibly interconnected by fusing or breaking links of an uncommitted metal interconnect matrix. Rhodes *et al.* were able to quickly design and implement a HT processor which runs at frame rates.

Baringer *et al.* [101] have proposed an architecture called PPPE, parallel pipeline projection engine, which uses the ideas of the Radon transform as a projection operation to produce a realtime hardware implementation of the HT. A set of VLSI chips is currently under design and fabrication and they expect to achieve real-time implementation of the Radon transform using only one or two ICs.

Leavers and Sandler [133] have considered efficient implementation of the line-finding HT on multitasking and asynchronous parallel architectures. By utilizing edge information to restrict the number of accumulator cells which have to be incremented and by using table lookup wherever possible they produce an efficient algorithm.

Several authors have investigated the implementation of the HT on currently available SIMD architectures. These usually consist of square arrays of simple processing elements, PEs, connected so that each can communicate with its 4 or 8 neighbors. All processors concurrently execute the same instructions on different items of data. Most often each PE has only a single bit arithmetic logic unit, several registers, and a few kilobits of a RAM chip. The mapping of the HT onto these

architectures was first considered by both Silberberg [81] and Li [76]. Silberberg considered distributing both the image and the accumulator arrays among all PEs, i.e., each PE is assigned both an image feature and a cell of parameter space. However, this necessitated a great deal of data transfers between PEs in order to pass the votes to the correct PE for accumulation. Over 98% of the algorithm time was taken in shifting data around the processing array and the end result was that the use of 8000 simple PEs led to a speedup factor of only 7, relative to a standard implementation on a VAX 11/785. This poor result illustrates the care required in mapping algorithms onto parallel machines.

Li [76] considered two schemes for running his fast Hough transform, FHT, on a SIMD architecture. In the first scheme each PE is assigned an image feature and the coordinates of a parameter cell are broadcast simultaneously to every PE by a central controller. Each PE decides whether the hypersurface generated by its image feature intersects the cell and if it does it sends a vote back to the controller. The votes from each PE can be summed by the central controller and stored for later analysis. A second scheme is to assign each PE a volume of parameter space and broadcast the imge features. The choice of method depends on the number of available PEs, the number of image features and the number of parameter cells. For the standard HT the number of parameter cells increases exponentially with dimensionality of the problem and therefore the first alternative is likely to be the most feasible.

Simplistically one might argue that the use of n PEs, one for each of n image features, should speed up the HT by a factor of $O(n)$. However, the votes have to be routed to their correct places in the accumulator and on a simple square 2D mesh with only local shifting operations this requires $O(\sqrt{n})$ steps. As Silberberg found out this can be a significant overhead and even the dominant time factor. However, if the array is augmented by a tree-based voting network this can be reduced to $O(\log(n))$ and the expected gains in efficiency should again approach $O(n)$.

Little, Blelloch, and Cass [123] describe a possible implementation of the HT on an architecture called the *connection machine*. This is similar to the previously mentioned SIMD architectures but, as well as PEs communicating with near neighbors, there is a hardware router which implements rapid communication between any pair of processors. The architecture is based on a 12-dimensional hypercube, i.e., every processor can be reached from any other by traversing at most 12 edges of the cube. Unfortunately this paper concentrates on aspects of programming and addressing and does not give any figures on the efficiency gains in using this parallel implementation.

In the most recent literature Cypher, Sanz, and Snyder [107] describe 5 new algorithms for calculating the Radon transform (and by implication the HT) on a $n \times n$ mesh array. Three of their algorithms have a time complexity of $O(P + n)$, where P is the number of projections into which the full transform is decomposed. This compares with a complexity of $O(Pn)$ for Silberberg's implementation of the HT. The first four of the methods of Cypher *et al.* are based on rotating the image so that all pixels lying on lines at angle θ align with image rows. Once this is achieved, projection calculation simply involves a series of horizontal shifts and adds.

Guerra and Hambrusch [115] have also presented two efficient algorithms for HT line finding on an $n \times n$ mesh, in particular, the *massively parallel processor* (MPP).

Their first algorithm, the block algorithm, involves partitioning the mesh into submeshes, performing projections in these submeshes, and then combining partial results. Their second method is similar to one of the algorithms of Cypher *et al.* and performs projection by tracing lines through the image in a pipeline fashion. Although this tracing algorithm is asymptotically optimal in terms of complexity, Guerra and Hambrusch expect the block algorithm to outperform it in an actual implementation.

Fisher and Highman [113] describe a neat method of doing line finding HT using a machine called SLAP, scan line array processor. This is a linear array of SIMD vector processors. A single PE is assigned to each column of the image and the array moves over the image and processes all pixels of a row concurrently. Theirs is a projection-based idea and the principle is to create and move accumulator cells between PEs so that they are always in the PE which is processing a pixel on the line corresponding to the parameters of the accumulator cell. Cells are created at one edge of an image and move across the line of processors at a rate determined by their θ parameter. When they emerge at the opposite edge of the linear array they have processed data from all pixels of the line that they represent. Fisher and Highman are implementing their ideas in a 3μ CMOS chip and expect to be able to compute a full HT on a 512×512 image in 2–3 frame times.

Olsen, Bukys, and Brown [125] discuss the operation of the HT on a MIMD computer architecture. They use a commercially available system called the BBN Butterfly Parallel Processor™. This consists of 256 processing nodes connected by a switching network. Each processor is a microprocessor from the Motorola 68000 series and each node has a substantial amount of local memory, i.e., 1 Mbyte. Memory is shared in the sense that a switching network allows any processor to address the local memory of any other processor. Olsen *et al.* implemented two different line finding HTs using different parameterizations for lines. The versions differed in whether the image was partitioned among the processors with the parameter cells accessed from common memory or vice versa. They consider whether the algorithm is computation or communication resource bound and their results indicate that it is possible to achieve processor utilization factors in the region of 80%, i.e., if 100 processors are used they will give a speed up of 80 times, relative to using a single processor.

The HT has attracted attention from researchers interested in human vision as it is a prime example of the ideas of the connectionist school of artificial intelligence [61]. The unifying principle of this approach to intelligence is that low and medium level vision tasks are done by massively parallel, cooperative computations on large networks of simple neuron like units. Low-level pixel-based properties, like edge or gray level estimates, can be represented by nodes in a feature network which is spatially indexed to the image while parameter values associated with higher level image entities, such as straight lines, can be represented by nodes in a separate parameter network. Each node records a measure of confidence of the occurrence of its feature or parameter value and direct connections between the two networks define possible ways in which nodes can influence these confidence values. Different connection patterns can be used to impose different image space to parameter space mappings, i.e., connections can be established so that when many low-level units which lie on a straight line have high confidence then the higher level unit describing the parameters of this line will acquire a large confidence value. The major

characteristic of this type of implementation is the huge number of feature and parameter units needed and the very large number of connections which are required between them.

Cantoni *et al.* [104, 105] have considered how the HT might be used as part of a hierarchically organized shape recognition system. They use the HT to group low-level features and provide structural descriptions of the shapes which they wish to find. They discuss implementation of the process on both a SIMD array and a pyramid machine. In the pyramid approach, features can be detected at low resolution and final integration can be achieved by looking at the detected features at higher resolutions. They report near real-time implementation of their voting process even on a sequential simulation.

Blanford [102] has described how the *dynamically quantized pyramid* method and his modification of it can be naturally mapped to a parallel pyramid machine. However, one possible problem is that these algorithms require some multiplication and division operations and these are not efficiently done using the simplest bit serial processors which are envisaged in the current generation of these machines.

Fischler and Firschein [112] have recently shown that the HT is an example of an algorithm which can be implemented on a blackboard or database architecture. They invoke a principle which they call parallel guessing which says that it is often beneficial computationally to try to guess a solution rather than exhaustively compute a solution. They suggest computing the HT incrementally and terminating computation when a sufficiently significant parameter peak has been identified.

10. SUMMARY

The HT is an important method for many computer vision tasks. It is a very robust method in the presence of extra data and can cope well with situations where some items of data are missing. Nevertheless, there have traditionally been difficulties which have meant it has not been enthusiastically adopted. Its major problems have been that in its simplest implementation it requires a lot of computation and a lot of storage for high-dimensional arrays. However, we have seen that much current research on focusing approaches have led to very fast and/or space-efficient digital implementations. The relationship to other transforms has also led to the development of real-time analog implementations. Significant initial work has been started on topics such as peak detection and enhancement methods, the detailed analysis of accumulator distributions for both noise and real image features, low-dimensional parameterizations of curves, and new specialized computer architectures for the HT. Taking all this together suggests that the HT is provoking more interest and offering more promise than ever before.

REFERENCES

The following abbreviations have been used in this list:

IEEE	Institute of Electrical and Electronic Engineers
T-PAMI	Transactions on Pattern Analysis and Machine Intelligence
T-COMP	Transactions on Computers
T-SMC	Transactions on Systems, Man and Cybernetics
T-NS	Transactions on Nuclear Science
T-ASSP	Transactions on Acoustics, Speech and Signal Processing
CACM	Communications of the Association of Computing Machinery
CAPAMI	Computer Architectures for Pattern Analysis and Machine Intelligence
CGIP	Computer Graphics and Image Processing

CVGIP	Computer Vision, Graphics and Image Processing
IVC	Image and Vision Computing
PR	Pattern Recognition
PRL	Pattern Recognition Letters
IJCPR	International Joint Conference on Pattern Recognition
IJCAI	International Joint Conference on Artificial Intelligence
PRIP	Pattern Recognition and Image Processing
CVPR	Computer Vision and Pattern Recognition
SPIE	Society of Photogrametric and Instrumentation Engineers

1962

1. P. V. C. Hough, A method and means for recognizing complex patterns, U.S. Patent 3,069,654.

1965

2. M. J. Bazin and J. W. Benoit, Offline global approach to pattern recognition for bubble chamber physics, *IEEE T-NS* **12**, 291.

1969

3. A. Rosenfeld, *Picture Processing by Computer*, Academic Press, New York/London.

1971

4. P. Bastian and L. Dunn, Global approach to pattern recognition for bubble chamber physics, *IEEE T-COMP*. **20**, 995.

1972

5. R. D. Duda and P. E. Hart, Use of the Hough transform to detect lines and curves in pictures, *CACM* **15**, 11–15.

1973

6. F. O'Gorman and M. B. Clowes, Finding picture edges through collinearity of feature points, in *3rd IJCAI, Stanford* (1973); *IEEE T-COMP* **25**, 449–456 (1976).

1974

7. S. D. Shapiro, Detection of lines in noisy pictures using clustering, in *2nd IJCPR, Copenhagen*, pp. 317–318.

1975

8. C. Kimme, D. H. Ballard, and J. Sklansky, Finding circles by an array of accumulators, *CACM* **18**, 120–122.
9. P. M. Merlin and D. J. Farber, A parallel mechanism for detecting curves in pictures, *IEEE T-COMP*. **24**, 96–98.
10. S. D. Shapiro, Transformations for the computer detection of curves in noisy pictures, *CGIP* **4**, 328–338.

1976

11. S. D. Shapiro, Aspects of transform method for curve detection, in *1976 Joint Workshop on Pattern Recognition and Artificial Intelligence*, pp. 90–97.
12. S. D. Shapiro, An extension of the transform method of curve detection for textured image data, in *3rd IJCPR, Coronado*, pp. 205–207.

1977

13. M. Cohen and G. T. Toussaint, On the detection of structures in noisy pictures, *PR* **9**, 95–98.
14. S. D. Shapiro and A. Iannino, Performance of transforms for curve detection, in *IEEE Conf. on PRIP, Long Beach*, pp. 378–385.

15. G. C. Stockman and A. K. Agrawala, Equivalence of Hough curve detection to template matching, *CACM* **20**, 820–822.
16. H. Wechsler and J. Sklansky, Automatic detection of ribs in chest radiographs, *PR* **9**, 21–30.

1978

17. A. Iannino and S. D. Shapiro, A survey of the Hough transform and its extension to curve detection, *IEEE Conf. on PRIP, New York*, pp. 32–38.
18. S. D. Shapiro, Generalization of the Hough transform for curve detection in noisy digital images, in *4th IJCPR, Kyoto*, pp. 710–714.
19. S. D. Shapiro, Feature space transforms for curve detection, *PR* **10**, 129–143.
20. S. D. Shapiro, Transform method of curve detection for textured image data, *IEEE T-COMP* **27**, 254–255.
21. S. D. Shapiro, Curve formation using transforms for pictures governed by differential equations, *IEEE T-SMC* **8**, 763–765.
22. S. D. Shapiro, Properties of transforms for the detection of curves in noisy images, *CGIP* **8**, 219–236.
23. J. Sklansky, On the Hough technique for curve detection, *IEEE T-COMP* **27**, 923–926.
24. S. Tsuji and F. Matsumoto, Detection of ellipses by modified Hough transformation, *IEEE T-COMP* **27**, 777–781.

1979

25. C. L. Fennema and W. B. Thompson, Velocity determination in scenes containing several moving objects, *CGIP* **9**, 301–315.
26. S. D. Shapiro and A. Iannino, Geometric constructions for predicting Hough transform performance, *IEEE T-PAMI* **1**, 310–317.

1980

27. L. S. Davis and S. Yam, *A Generalized Hough-like Transformation for Shape Recognition*, TR-134, University of Texas Computer Sciences.
28. S. D. Shapiro, Use of the Hough transform for image data compression, *PR* **12**, 333–337.
29. K. R. Sloan and D. H. Ballard, Experience with the generalized Hough transform, in *5th IJCPR, Miami Beach*, pp. 174–179.

1981

30. V. S. Alagar and L. H. Thiel, Algorithms for detecting m-dimensional objects in n-dimensional spaces, *IEEE T-PAMI* **3**, 245–256.
31. D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *PR* **13**, 111–122.
32. S. R. Deans, Hough transform from the Radon transform, *IEEE T-PAMI* **3**, 185–188.
33. H. Mostafavi and R. Ollenburger, Image analysis using polarized Hough transform and edge enhancer, *SPIE Conf.* **302**, 51–59.
34. J. O'Rourke, Motion detection using Hough techniques, in *IEEE PRIP81, Dallas*, pp. 82–87.
35. J. O'Rourke, Dynamically quantized spaces for focusing the Hough transform, in *7th IJCAI, Vancouver*, pp. 737–739.
36. T. Shibata and W. Frei, Hough transform for target detection in infrared imagery, *SPIE* **281**, 105–109.
37. K. R. Sloan, Dynamically quantized pyramids, in *7th IJCAI, Vancouver*, pp. 734–736.
38. T. M. Van Veen and F. C. A. Groen, Discretization errors in the Hough transform, *PR* **14**, 137–145.
39. S. Yam and L. S. Davis, Image registration using generalized Hough transform, *IEEE PRIP81*, Dallas, pp. 526–533.

1982

40. C. M. Brown and D. B. Sher, Modelling the sequential behaviour of Hough transform schemes, in *Image Understanding Workshop, Palo Alto*, by (L. S. Baumann, Ed.), pp. 115–123.

41. L. S. Davis, Hierarchical generalized Hough transforms and line segment based generalized Hough transforms, *PR* **15**, 277–285.
42. K. R. Sloan, Analysis of "Dot Product Space," shape descriptions, *IEEE T-PAMI* **4**, 87–90.

1983

43. G. Adiv, Recovering motion parameters in scenes containing multiple moving objects, in *IEEE CVPR Conf., Washington*, pp. 399–400.
44. Arbuschi, V. Cantoni, and G. Musso, Recognition and location of mechanical parts using Hough transform, in *Image Analysis*, (S. Levialdi, Ed.), Pitman, London.
45. D. H. Ballard and O. A. Kimball, Rigid body motion from depth and optical flow, *CVGIP* **22**, 95–115.
46. D. H. Ballard and D. Sabbah, Viewer independent shape recognition, *IEEE T-PAMI* **5**, 653–660.
47. C. M. Brown, M. B. Curtiss, and D. B. Sher, Advanced Hough transform implementations, in *8th IJCAI, Karlsruhe*, pp. 1081–1085; TR113, Computer Science Department, University of Rochester.
48. C. M. Brown, Inherent bias and noise in the Hough transform, *IEEE T-PAMI* **5**, 493–505.
49. C. M. Brown, *Hierarchical Cache Accumulators for Sequential Mode Estimation*, TR125, Computer Science Department, University of Rochester.
50. A. E. Cowart, W. E. Snyder, and W. H. Ruedger, The detection of unresolved targets using the Hough transform, *CVGIP* **21**, 222–238.
51. C. R. Dyer, Gauge inspection using Hough transform, *IEEE T-PAMI* **5**, 621–623.
52. G. Eichmann and B. Z. Dong, Coherent optical production of the Hough transform, *Appl. Opt.* **22**, No. 6, 830–834.
53. S. N. Jayaramamurthy and R. Jain, An approach to the segmentation of textured dynamic scenes, *CVGIP* **21**, 239–261.
54. S. Kasif, L. Kitchen, and A. Rosenfeld, A Hough transform technique for subgraph isomorphism, *PRL* **2**, 83–88.
55. M. Kushnir, K. Abe, and K. Matsumoto, An application of the Hough transform to the recognition of printed Hebrew characters, *PR* **16**, 183–191.
56. W. C. Lin and R. C. Dubes, A review of ridge counting in dermatoglyphics, *PR* **16**, 1–8.
57. H. Mostafavi and M. Shah, High speed digital implementation of linear feature extraction algorithms, *SPIE* **432**, *Applications of Digital Image Processing 6, San Diego*, pp. 182–188.
58. P. R. Thrift and S. M. Dunn, Approximating point set images by line segments using a variation of the Hough transform, *CVGIP* **21**, 383–394.
59. H. Tsukune and K. Goto, Extracting elliptical figures from an edge vector field, *IEEE CVPR Conf., Washington*, pp. 138–141.
60. A. Wallace, Grayscale image processing for industrial applications, *IVC* **1**, 178–188.

1984

61. D. H. Ballard, Parameter nets, *Artif. Intell.* **22**, 235–267.
62. G. R. Gindi and A. F. Gmitro, Optical feature extraction via the Radon transform, *Opt. Eng.* **23**, No. 5, 499–506.
63. H. Hakalahti, D. Harwood, and L. S. Davis, Two-dimensional object recognition by matching local properties of contour points, *PRL* **2**, 227–234.
64. T. C. Henderson and W. S. Fai, The 3D Hough shape transform, *PRL* **2**, 235–238.
65. H. A. H. Ibrahim, R. J. Kender, and D. E. Shaw, The Hough transform method on fine grained, tree structured SIMD machines, in *Image Understanding Workshop* (L. S. Baumann, Ed.), pp. 216–224.
66. R. M. Inigo, E. S. McVey, B. J. Berger, and M. J. Wirtz, Machine vision applied to vehicle guidance, *IEEE T-PAMI* **6**, 820–826.
67. Y. Muller and R. Mohr, Planes and quadrics detection using Hough transform, in *7th IJCPR, Montreal*, pp. 1101–1103.
68. L. O'Gorman and A. C. Sanderson, The converging squares algorithm: An efficient method for locating peaks in multidimensions, *IEEE T-PAMI* **6**, 280–288.
69. J. O'Rourke and K. R. Sloan, Dynamic quantization: Two adaptive data structures for multidimensional spaces, *IEEE T-PAMI* **6**, 266–279.
70. S. A. Peek, J. E. W. Mayhew, and J. P. Frisby, Obtaining viewing distance and angle of gaze from vertical disparity using a Hough-type accumulator, *IVC* **2**, 180–190.

71. 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, Ed.), pp.* 770–775.
72. T. M. Silberberg, L. Davis, and D. Harwood, An iterative Hough procedure for 3D object recognition, *PR* **17**, 621–629.

1985

73. F. Evans, A survey and comparison of the Hough transform, in *IEEE Conf. on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 378–380.
74. K. Y. Huang, K. S. Fu, T. H. Sheen, and S. W. Cheng, Image processing of seismograms: (A) Hough transformations for the detection of seismic patterns, *PR* **18**, 429–440.
75. H. Li, M. A. Lavin, and R. J. LeMaster, Fast Hough transform, in *Proceedings, 3rd Workshop on Computer Vision: Representation and Control, Bellair, Michigan*, pp. 75–83.
76. H. Li, *Fast Hough Transform for Multidimensional Signal Processing*, IBM Research report RC 11562, Yorktown Heights.
77. M. Kushnir, K. Abe, and K. Matsumoto, Recognition of handprinted Hebrew characters using features selected in the Hough transform space, *PR* **18**, pp. 103–114.
78. I. Moring and H. Hakalahti, Scale independent method for object recognition, in *Proceedings, Scandinavian Conf. on Image Processing, Trondheim*, pp. 881–889.
79. M. Nixon, Application of Hough transform to correct for linear variation of background illumination in images, *PRL* **3**, 191–194.
80. J. L. C. Sanz, E. B. Hinkle, and I. Dinstein, Computing geometric features of digital objects in general purpose image processing pipeline architectures, in *IEEE CVPR Conf., San Francisco*, pp. 265–270.
81. T. M. Silberberg, The Hough transform on the geometric arithmetic parallel processor, in *IEEE Conf. on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 387–393.
82. B. W. Suter, A parallel algorithm for the discrete Randon transform, in *IEEE Conf. on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 394–398.
83. J. L. Turney, T. N. Mudge, and R. A. Volz, Recognizing partially occluded parts, *IEEE T-PAMI* **7**, 410–421.
84. R. S. Wallace, A modified Hough transform for lines, in *IEEE CVPR Conf., San Francisco*, pp. 665–667.

1986

85. M. F. Costabile and G. G. Pieroni, Detecting shape correspondences by using the generalized Hough transform, in *8th IJCPR, Paris*, pp. 589–591.
86. E. R. Davies, Image space transforms for detecting straight edges in industrial images, *PRL* **4**, 185–192.
87. E. R. Davies, Corner detection using the generalized Hough transform, in *IEE 2nd International Conference on Image Processing and its Applications, London*, pp. 175–179.
88. E. R. Davies, Reduced parameter spaces for polygon detection using the generalized Hough transform, in *8th IJCPR, Paris*, pp. 495–497.
89. M. Dhome and T. Kasvand, Hierarchical approach for polyhedra recognition by hypothesis accumulation, in *8th IJCPR, Paris*, pp. 88–91.
90. A. V. Forman, A modified Hough transform for detecting lines in digital imagery, *SPIE* **635**, *Applications of AI(iii)*, 151–160.
91. G. Gerig and F. Klein, Fast contour identification through efficient Hough transform and simplified interpretation strategy, in *8th IJCPR, Paris*, pp. 498–500.
92. V. F. Leavers and J. F. Boyce, *Automatic Shape Parameterizations in Machine Vision*, National Physical Lab. report DITC 79/86.
93. H. Li and M. A. Lavin, Fast Hough transfoirm based on the bintree data structure, in *IEEE CVPR Conf., Miami Beach*, pp. 640–642.
94. H. Li, M. A. Lavin, and R. J. LeMaster, Fast Hough transform: A hierarchical approach, *CVGIP* **36**, 139–161.
95. H. Maitre, Contribution to the prediction of performances of the Hough transform, *IEEE T-PAMI* **8**, 669–674.

96. L. M. Murphy, Linear feature detection and enhancement in noisy images via Radon transform, *PRL* **4**, 279–284.
97. W. Poelzleitner, A Hough transform method to segment images of wooden boards, in *8th IJCPR, Paris*, pp. 262–264.
98. C. J. Radford, Optical flow fields in Hough transform space, *PRL* **4**, 293–305.
99. W. H. Steier and R. K. Shori, Optical Hough transform, *Appl. Opt.* , **25**, No. 16, 2734–2738.
100. F. M. Wahl and H. Biland, Decomposition of polyhedral scenes in Hough space, in *8th IJCPR, Paris*, pp. 78–84.

1987

101. W. B. Baringer, B. C. Richards, R. W. Broderson, J. L. C. Sanz, and D. Petkovic, A VLSI implementation of PPPE for real time image processing in Radon space—Work in progress, in *IEEE Workshop on CAPAMI, Seattle*, pp. 88–93.
102. R. P. Blanford, Dynamically quantized pyramids for Hough vote collection, in *IEEE Workshop on CAPAMI, Seattle*, pp. 145–152.
103. J. F. Boyce, G. A. Jones, and V. F. Leavers, An implementation of the Hough transform for line and circle detection, in *Proceedings, SPIE 4th Intl. Symposium on Optics and Optoelectronic Science and Engineering, The Hague.*
104. V. Cantoni and L. Carrioli, Structural shape recognition in a multiresolution environment, *Signal Process.* **12**, 267–276.
105. V. Cantoni, L. Carrioli, M. Ferretti, L. Lambardi, and K. Mathews, Object recognition in multi-resolution systems, in *Proceedings, NATO Advanced Research Workshop on Real-Time Object and Environment Measurement and Classification, Maratea, Italy, September 1987*, Springer-Verlag, New York/Berlin.
106. D. Casasent and R. Krishnapuram, Curved object location by Hough transformations and inversions, *PR* **20**, 181–188.
107. R. E. Cypher, J. L. C. Sanz, and L. Snyder, The Hough transform has $O(N)$ complexity on SIMD $N \times N$ mesh array architectures, in *IEEE Workshop on CAPAMI, Seattle*, pp. 115–121.
108. E. R. Davies, The performance of the generalized Hough transform: Concavities, ambiguities and positional accuracy, in *Proceedings, 3rd Alvey Vision Conference, Cambridge, England*, pp. 327–334.
109. E. R. Davies, A new parameterization of the straight line and its application for the optimal detection of objects with straight edges, *PRL* **6**, 9–14.
110. E. R. Davies, A new framework for analyzing the properties of the generalized Hough transform, *PRL* **6**, 1–8.
111. E. R. Davies, Improved localization in a generalized Hough scheme for the detection of straight edges, *IVC* **5**, No. 4, 279–286.
112. M. A. Fischler and O. Firschein, Parallel guessing: A strategy for high speed computation, *PR* **20**, 257–263.
113. A. L. Fisher and P. T. Highman, Computing the Hough transform on a scan line array processor, in *IEEE Workshop on CAPAMI, Seattle*, pp. 83–87.
114. G. Gerig, Linking image-space and accumulator-space: A new approach for object recognition, in *1st IEEE International Conf. on Computer Vision, London*, pp. 112–117.
115. C. Guerra and S. Hambrusch, Parallel algorithms for line detection on a mesh, in *IEEE Workshop on CAPAMI, Seattle*, pp. 99–106.
116. D. J. Hunt, L. W. Nolte, A. R. Reibman, and W. H. Ruedger, Hough transform and signal detection theory performance for images with additive noise, *CVGIP.*
117. D. J. Hunt, L. W. Nolte, and W. H. Ruedger, Performance of the Hough transform and its relationship to statistical signal detection theory, *CVGIP.*
118. J. Illingworth and J. Kittler, The adaptive Hough transform, *IEEE T-PAMI* **9**, No. 5, 690–697.
119. J. Illingworth, J. Kittler, and J. Princen, Shape detection using the adaptive Hough transform, in *Proceedings, NATO Advanced Research Workshop on Real-Time Object and Environment Measurement and Classification, Maratea, Italy, September 1987*, Springer-Verlag, New York/Berlin.
120. R. Krishnapuram and D. Casasent, Hough space transformations for discrimination and distortion estimation, *CVGIP* **38**, No. 3, 299–316.
121. V. F. Leavers and J. F. Boyce, The Radon transform and its application to shape parameterization in machine vision, *IVC* **5**, No. 2, 161–166.

122. V. F. Leavers and G. F. Miller, Radon transform of δ-function curves. A geometric approach, in *Proceedings, 3rd Alvey Vision Conference, Cambridge, England*, pp. 335–340.
123. J. J. Little, G. Blelloch, and T. Cass, Parallel algorithms for computer vision on the connection machine, in *1st IEEE International Conf. on Computer Vision, London*, pp. 587–591.
124. M. Nagoa and S. Nakajima, On the relation between the Hough transformation and the projection curves of a rectangular window, *PRL* **6**, 185–188.
125. T. J. Olson, L. Bukys, and C. M. Brown, Low level image analysis on an MIMD architecture, in *1st IEEE International Conf. on Computer Vision, London*, pp. 468–475.
126. J. L. C. Sanz and I. Dinstein, Projection based geometrical feature extraction for computer vision: Algorithms in pipeline architectures, *IEEE T-PAMI* **9**, 160–168.
127. J. L. C. Sanz, I. Dinstein, and D. Petkovic, Computing multi-colored polygonal masks in pipeline architectures and its application to automated visual inspection, *CACM* **30**, No. 4, 318–329.
128. J. L. C. Sanz and E. B. Hinkle, Computing projections of digital images in digital processing pipeline architectures, *IEEE T-ASSP* **35**, No. 2, 198–207.
129. J. Skingley and A. J. Rye, The Hough transform applied to SAR images for thin line detection, *PRL* **6**, 61–67.
130. A. M. Wallace, Matching segmented scenes to models using pairwise relationships between features, *IVC* **5**, No. 2, 114–120.

1988

131. U. Eckhardt and G. Mederlechner, Application of the projected Radon transform in picture processing, in *Proceedings, British Pattern Recognition Association 4th International Conf. on Pattern Recognition* (J. Kittler, Ed.), pp. 370–379, Springer-Verlag, New York/Berlin.
132. K. Hanahara, T. Maruyama, and T. Uchiyama, A real-time processor for the Hough transform, *IEEE T-PAMI* **10**, No. 1, 121–125.
133. V. F. Leavers and M. B. Sandler, On efficient Radon transform, in *Proceedings, British Pattern Recognition Association 4th International Conf. on Pattern Recognition* (J. Kittler, Ed.), pp. 380–389, Springer-Verlag, New York/Berlin.
134. F. M. Rhodes, J. J. Dituri, G. H. Chapman, B. E. Emerson, A. M. Soares, and J. I. Raffel, A monolithic Hough transform processor based on restructurable VLSI, *IEEE T-PAMI* **10**, No. 1, 106–110.
135. D. B. Shu, C. C. Li, J. F. Mancuso, and Y. N. Sun, A line extraction method for automated SEM inspection of VLSI resist, *IEEE T-PAMI* **10**, No. 1, 117–120.
136. R. C. Thomson and E. Sokolowska, Mineral cleavage analysis via the Hough transform, in *Proceedings, British Pattern Recognition Association 4th International Conf. on Pattern Recognition* (J. Kittler, Ed.), pp. 390–398, Springer-Verlag, New York/Berlin.