Unsupervised Clustering in Hough Space for Identification of Partially Occluded Objects

Oscar Yáñez-Suárez, *Member*, *IEEE*, and
Mahmood R. Azimi-Sadjadi, *Senior Member*, *IEEE*

Abstract—An automated approach for template-free identification of partially occluded objects is presented. The contour of each relevant object in the analyzed scene is modeled with an approximating polygon whose edges are then projected into the Hough space. A structurally adaptive self-organizing map neural network generates clusters of collinear and/or parallel edges, which are used as the basis for identifying the partially occluded objects within each polygonal approximation. Results on a number of cases under different conditions are provided.

Index Terms—Image analysis, occluded objects, unsupervised clustering SOM network, Hough space.

1 Introduction

The demand of automated procedures for the analysis of digital scenes has gone beyond the traditional applications of robotic assembly and production-line inspection. In these applications, object occlusion probably generates the most significant problem when trying to accurately describe the scene. Over the years, several approaches have been proposed [1], [2], [3] which, for the most part, rely on the existence of a template library. This library generally consists of a set of exemplar patterns or extracted features describing the objects that are expected to appear in the scene. Some sort of matching algorithm is then defined that allows the identification of the partially observed objects by comparing the real scene information with the template library contents.

For certain applications, however, the compilation of a template library makes little sense as the objects to be analyzed in the scene will never repeat. In such cases, the library must be replaced by some high-level description of the object properties. We refer to a scenario like this as "template-free" object identification. For example, in evaluating airborne fiberglass particle distributions using scanning electron micrographs (SEM), the objects of interest have shapes resembling parallelograms, but no two objects are identical [4]. In this paper, we introduce an approach that automatically searches the fibers in digitized micrographs. This method is based on Hough-space representations [5], [6] of polygonal approximations for the objects in the scene. Clusters in the transformed space identify sets of edges that can be combined to describe a fiber object. To find the unknown number and location of clusters, we utilize a structurally adaptive variant of the self-organizing map (SOM) neural network that we have developed [7]. This provides a fast, object-based clustering and facilitates the implementation of the template-free fiber identification.

2 Object Segmentation and Modeling

The faultless detection of closed, eight-connected contours is a required step for modeling the objects in the scene. To obtain these contours, a threshold-based segmentation scheme is favored over edge detection approaches, as the available edge extraction methods do not guarantee that the required closed contours would be obtained (which imposes the need for an edge linking procedure). On the other hand, extracting the connected contours from binary images is straightforward.

SEM images consistently exhibit a gray-level distribution with a bimodal profile. However, the prior probabilities of the histogram modes vary significantly among the studied cases (see, for example, Fig. 1). Simple valley detection strategies for the selection of an adequate threshold fail when one of the priors is very small. The scheme adopted here selects the threshold using a Bayesian approach based on a two-component Gaussian model of the image histogram. Such a model is obtained through the use of the expectation maximization (EM) estimation algorithm [8]. In order to avoid significant preprocessing overhead due to the application of the EM iterations over the entire set of pixels in an SEM image, a 1:4 uniformly subsampled version of the image is actually used in the computations.

The eight-connected contours are obtained from the binary images by means of a morphological erosion followed by an exclusive-OR masking [9]. For a binary image with M objects, there are contour pixel sets with N_m, $m = 1 \dots M$ elements each. Every contour pixel set can be associated to a unique (up to a rotation) chain code descriptor [10]. Thus, the scene can be represented by the M ordered sets $O_m = \{(x_1, y_1)_m, \dots, (x_{N_m}, y_{N_m})_m\}$, $m = 1 \dots M$, where $(x_n, y_n)_m$ are the coordinates of the nth pixel in the ordered contour corresponding to the mth object present.

The information conveyed by every ordered contour pixel set O_m can be compactly represented after performing a piecewise linear approximation of its associated plane curve [11], [12], [13], [14]. The goal of the approximation is to find a polygon whose vertices form a subset $V_m \subset O_m$ of k points $V_m = \{(x_a, y_a)_m, (x_b, y_b)_m, \dots, (x_k, y_k)_m \mid 1 \le a < b < \dots < k \le N_m\}$ such that

$$\begin{gathered}\max\left|\left(y_n - m_{(p,q)}x_n - b_{(p,q)}\right)\left(m^2_{(p,q)} + 1\right)^{-\frac{1}{2}}\right| < \epsilon, \\ p \le n \le q,\ a \le p, q \le k,\end{gathered} \tag{1}$$

with $m_{(p,q)}$ the slope and $b_{(p,q)}$ the intercept of the line connecting $(x_p, y_p)_m$ with $(x_q, y_q)_m$. Note that a similar expression, with interchanged coordinate roles, is used when considering nearly vertical edges. Condition (1) guarantees that the maximum distance from any point $(x_n, y_n)_m$ occurring in the interval $p \le n \le q$ of the ordered contour set to its approximating polygon edge is less than some arbitrary threshold ϵ. The higher the value of ϵ, the coarser the polygonal approximation results. The search for the solution polygon V_m can be implemented with a recursive split algorithm, an efficient implementation of the Pavlidis-Horowitz approximation procedure [11], as follows: The ordered pixel set O_m is initially split into two subsets, $O_{m,1}$ and $O_{m,2}$, at the positions of the upper-leftmost and lower-rightmost pixels in O_m, which are usually high curvature points in the contour. For each subset $O_{m,k}$, the error condition (1) is tested. If the condition fails for one of the subsets, the point of maximum distance $(x_{split}, y_{split})_m$ is added to the solution set V_m. The corresponding subset is then split into two new subsets, which are recursively tested for compliance with the approximation criterion.

3 Analysis of Complex Objects

Whenever an image is segmented into the observable objects in the scene, the piecewise linear approximation described above will produce a number of polygons V_m that can be either convex or concave. Convex polygons can be assumed to represent single

- *The authors are with the Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523.*
E-mail: yaso@lamar.colostate.edu, azimi@engr.colostate.edu.

Manuscript received 17 Oct. 1998; revised 4 June 1999.
Recommended for acceptance by K.V. Mardia.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108076.

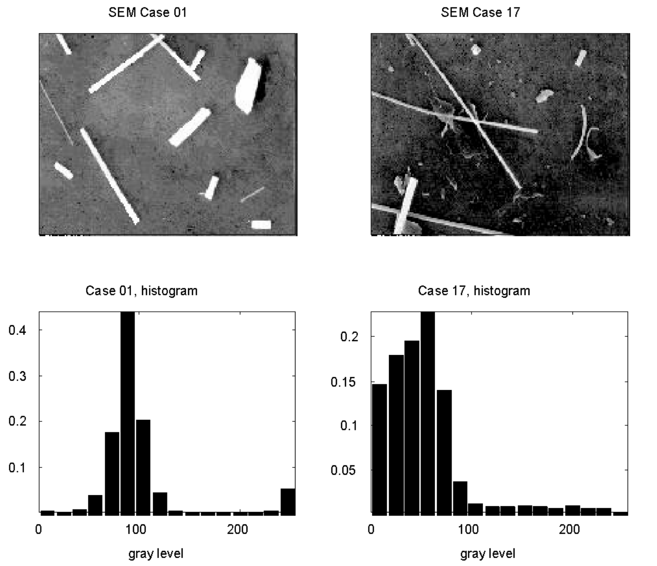

Fig. 1. Gray level histograms for two micrograph instances. While valley detection works for SEM case 01, it fails for case 17. Model-based segmentation provides adequate thresholding regardless of the intensity distribution for the particular experiment.

objects, possibly a fiber (mostly parallelograms) or some regularly shaped debris. Concave polygons are likely to represent complex objects, i.e. mixtures of simpler convex polygons that are either touching, overlapping or crossing. Thus, further processing of the concave polygons is required to analyze their constituents.

Our approach to the analysis of concave polygons is based on the Hough parametrization of the edges [5]. Such representation has the useful property that collinear edges share the same set of parameters ρ, the distance of the edges to the origin, and θ, the angle of the normal vector to the edge with respect to the abscissa axis, regardless of the choice of coordinate origin. Parallel lines will share the same θ while presenting different values of ρ. Thus, when mapping the edges of a complex object defined by a set V_m into the Hough space, several clusters will be formed, each one associated to one of the (partially occluded) convex polygon constituents of the complex object. Further, if the missing line segments that would otherwise connect collinear edges from the same side of a fiber were present in the scene (not occluded), these segments would also map to the same regions as the edges they would link together. It is important to note that, given that all the lines (edges) in the scene are already known after the polygonal modeling, Hough transformation is not required or used here for line detection (as in the standard grid accumulation procedure [10]). Rather, this transform is used to provide a more convenient representation of the concave polygon structures.

3.1 Unsupervised Clustering of Edge Parameters

There is a major issue in applying the above-mentioned clustering approach when breaking a complex object into its constituent parts. The number of simple objects is usually unknown and, thus, there is no way to know a priori how many clusters could be formed out of all the edge parameters of the concave polygon. This means that clustering procedures such as K-means or the self-organizing map neural network (SOM, [15]), which require the prior definition of the number of clusters to be found among the data, are inappropriate for the application. In addition, in any given micrograph, the number of independent complex objects is also unknown. This means that the clustering problem has to be solved several times per image using an efficient algorithm. We address this problem by using a variant of the SOM neural network that is built using a minimum spanning tree (MST) [16] organization. The MST-SOM [7] is a structurally adaptive unsupervised learning network in which the number of processing units and neighborhood relations are modifiable during the training phase. This architecture is well-suited for the complex

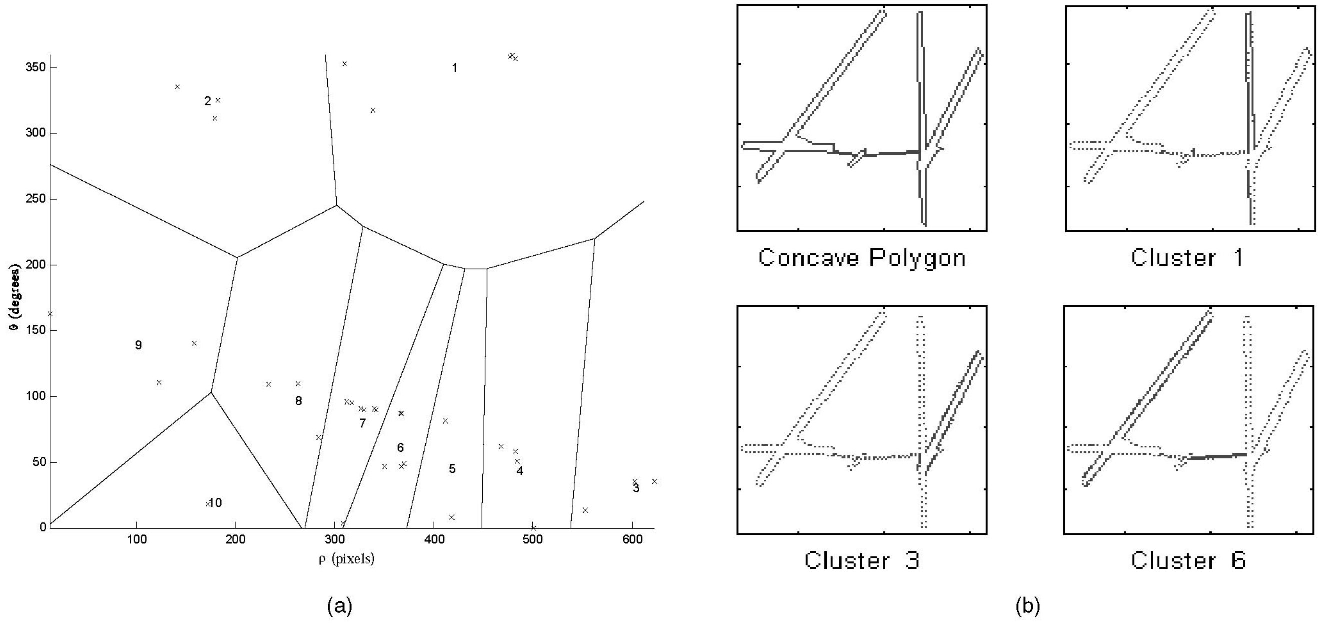

Fig. 2. (a) Voronoi partition of the Hough space derived from the weights of the 10 processing units of the MST-SOM built for clustering the edges of the concave polygon shown in Fig. 2b. Each cross indicates an element $\theta_{(p,q)} \in E_m$. The numbers identifying the clusters also indicate the location of the corresponding trained network weight. (b) Edge clustering with the MST-SOM. For the concave polygon illustrated in the upper left box, the network generates 10 different clusters (Fig. 2a) that corresond to subsets of polygon edges, as shown here for three different cases. For reference, the dotted lines indicate the original object contour. Solid lines indicate clustered polygon edges.

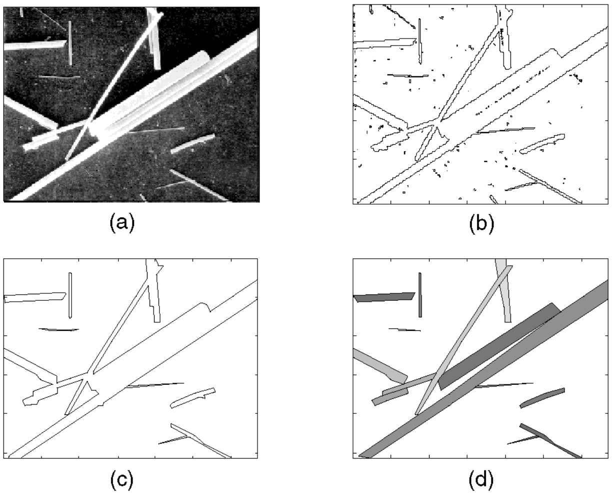

Fig. 3. Different stages of the analysis procedure for SEM #13 in Table 1. Note that the highly overlapped object extending from the upper right corner of the micrograph was missed by the detection procedure.

object analysis task as each processing unit in the network defines a cluster.

The SOM is a neural network architecture in which an ordered set of reference vectors (processing units or nodes) is adaptively distributed among the sample space in order to fit the distribution of the observed data. The number of nodes in the network defines how many clusters will be formed, while the location of the reference vectors in the sample space defines the corresponding partition. Positioning of the reference vectors is carried out through competitive learning, where local interactions among neighboring network nodes determine the final distribution of clusters. The MST-SOM network differs from the standard SOM in that the neighborhoods of the processing units are defined adaptively within the sample space based on the construction of an MST. This neighborhood definition facilitates the addition and removal of processing units during the training phase, as no specific geometry has to be established among the units at any given time. The structural adaptation amounts to the creation and elimination of clusters among the data. The criterion being optimized through the structural adaptation is the relative entropy [17] of the processing units' activation distribution with respect to an ideal, equiprobable activation distribution. A small value of the relative entropy corresponds to a spatial organization of the processing units that better describes the distribution of the training data. No specific constraint on cluster size is imposed in the formulation, although the relative entropy goal will indirectly influence this size, generating large clusters in scarcely-populated regions of the sample space. For a detail discussion on the MST-SOM weight updating rule, the reader is referred to [7]. It is important to note, however, that this training has fast convergence properties, is relatively insensitive to initial placement of the units, and does not require a learning rate parameter, all of which are desirable features for implementation. After each updating step, the relative entropy is monitored and the structure of the network is modified accordingly. If the relative entropy is not diminishing, then a new processing unit is added to the network in an attempt to redistribute the activations. All the inactive units are also removed at this point. When the network updating process is terminated, the clusters defined by the Voronoi partition associated with the final network weights will group the edges of the related objects together. As an example of this procedure, Fig. 2a shows the partition resulting from the unsupervised clustering of the edges of the complex object shown in Fig. 2b, where some of the associated edge subsets are also highlighted.

The final step in complex object analysis is to connect the edge subsets in order to define meaningful convex polygons. Given that the elements of V_m are ordered according to their positions on the contour's chain code, it follows that the polygon edges are also ordered consecutively, forming the set:

$$E_m = \{\Theta_{(a,b)}, \Theta_{(b,c)}, \ldots, \Theta_{(k-1,k)}, \Theta_{(k,a)}\}, \tag{2}$$

where $\Theta_{(p,q)} = (\rho_{(p,q)}, \theta_{(p,q)})$, $a \leq p, q \leq k$, are the normal parameters of the polygon edge connecting $(x_p, y_p)_m$ with $(x_q, y_q)_m$. Unsupervised clustering of the elements of E_m generates P subsets $C_{m,n} \subset E_m$, $n = 1, \ldots, P$ of edges. The elements of these subsets are still ordered, but, in general, they will not represent a consecutive set of edges. Consider, for example, some edge cluster $C_{m,n}$ described by the following set:

$$C_{m,n} = \{\Theta_{(a,b)}, \Theta_{(c,d)}, \Theta_{(d,e)}, \ldots, \Theta_{(k,a)}\}. \tag{3}$$

Two conditions can be observed in (3). The cluster has grouped together two edges, namely $\Theta_{(c,d)}$ and $\Theta_{(d,e)}$, that occur consecutively in the perimeter of the concave polygon, i.e., the edges share an endpoint. These edges could possibly pertain to the same simple object. On the other hand, the cluster does not include the $\Theta_{(b,c)}$ edge, although it does include the previous and the next edges from E_m. These disjoint edges, $\Theta_{(a,b)}$ and $\Theta_{(c,d)}$, may or may not correspond to the same object. The clustering process will generate many of these disjoint edge conditions because there will always be partially occluded objects within a complex object.

The missing connections between nonconsecutive edges are the necessary pieces that will break a complex object into its components. Some assumptions about the shape of the simple objects that are being sought for have to be made in order to decide if the missing, unobserved edges should be added. Since the goal is to search for fiber objects that are closely described by parallelograms, then a sensible and broad consideration to take is that any added connection should aim at the construction of a parallelogram. It is also natural to enforce that any edge to be added must at least lie within the boundaries of the original concave polygon.

When the unobserved and possibly occluded edge $\Theta_{(b,c)}$ in (3) lies within the concave polygon, the MST-SOM can be used to investigate if it is also clustered in $C_{m,n}$. If this is the case, then this edge can be added to the set, as it will not deviate the simple polygon from resembling a parallelogram. When the unobserved edge is not clustered in $C_{m,n}$, a second test has to be applied to decide whether the edge is lying across the width of the parallelogram or not. Two edges of this kind are required so that the extracted object corresponds to a closed polygon. Ignoring an unobserved connecting edge implies that a valid closed polygon cannot be formed unless the edge $\Theta_{(c,d)}$ is removed from $C_{m,n}$. This action will create $\Theta_{(b,d)}$, a new unobserved edge that has to be analyzed as before for possible inclusion. The procedure just described for completing parallelograms out of the clustered edges will likely leave some of the edges unused. A typical situation in which this will happen is when analyzing two fibers that are lying side by side and touching each other. Extra passes of the described analysis procedure have to be performed on the remaining edges until no more simple objects can be formed.

4 Experimental Results

Twenty-six SEM preparations of fiberglass were acquired from Schuller, Inc. in Denver, Colorado, and analyzed. Images from each preparation were obtained by digitizing the video output of the microscope at x1000 magnification power. The spatial resolution of the setting was 0.2 μm per pixel, generating images of 512

TABLE 1
Visually Identified (Observed) and Detected Object Counts for the 26 Cases Studied

SEM case	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
observed objects	12	12	13	10	5	7	4	6	12	4	10	20	17	12	15	7	7	4	9	9	7	12	7	9	11	6
detected objects	14	13	16	12	6	10	4	8	13	6	11	16	14	20	21	16	15	9	11	8	11	13	10	12	9	5

Case 13 corresponds to the example shown in Fig. 3.

by 512 pixels, with a grayscale resolution of four bits per pixel. Coordinate origin was assumed to be at the lower left corner of the image. These images varied in fiber density and amount of debris present. Fig. 3 (case 13) shows different stages in the analysis of a sample SEM. Table 1 summarizes the counts of visually identified objects per image and the number of objects detected by our method. Processing time is dependent on the complexity of the scene, as every complex object found requires an MST-SOM run and its subsequent cluster analysis. Processing a typical object with 15-20 line segments takes from 5 to 10 MST-SOM iterations.

5 Discussion

To the authors' knowledge, the method presented here for the identification of partially occluded objects is unique in the sense that it does not rely on a template library for shape matching. The use of such libraries is the common denominator found in related literature on the subject. This is understandable, as many of the applications on occluded object identification are aimed at robotic vision for sorting, placing, or selecting known parts in a production line. Our application required a different approach given that every SEM image presents a unique set of objects for which no previous information is available. Although we have restricted ourselves to simple objects resembling parallelograms in this application, it seems feasible to extend the present approach to other situations by adequately redefining the corresponding geometrical constraints. For example, in more complicated scenarios, the projection of the polygon parameters onto higher-dimensional Hough spaces may result in a simplified representation of the objects in the scene. Nonetheless, it must be expected that the lack of reference templates and the generality of the geometrical assumptions would also be a source of errors that have to be dealt with in an appropriate way.

It is important to note that the object counts reported in Table 1 indicate that most of the times our method is detecting more objects than those actually existing in the original SEM. The occurrence of these "extra" detected objects is the result of the assumptions made with respect to the shape of the partially occluded objects. As a result of these assumptions, the separation of the complex object components can be sometimes carried out in more than one way. The most frequent case occurs when the shape of the object being analyzed is close to a convex polygon, but yet it is detected as concave due to some segmentation noise. In this case, the (almost convex) approximating polygon can be broken down into various overlapping parallelograms, each one covering most of the area of the original object. These solutions are clearly redundant and can be easily detected and eliminated, reducing the extra object count. It is important to note that for the images analyzed in this work, this condition alone accounts for 80 percent of the extra object counts.

A second condition generating extra objects can be observed with curved fibers. Under the parallelogram assumptions, such fibers will be broken into a series of multiple objects. In this case, the relevant property of these extra objects is that they share a common edge. Unfortunately, this can also happen in other situations not involving curved fibers and, therefore, cannot be used as an indicator of this condition. This means that, in its current version, our method has a compromise between discovering curved fibers and failing to identify other, noncurved objects.

There are also instances of missing detections, as happens for the case shown in Fig. 3. Very narrow fibers occasionally have a low intensity content and, in some cases, the model-derived segmentation threshold can confuse them completely or partially with the background. Fibers with extensive overlap, like the large ones seen toward the center of the scene, cannot be resolved with our procedure. In this case, the information derived from the clusters is insufficient to completely decide that an edge having the length of the entire overlapping fiber has to be supplied. A different technique in generating the micrograph preparation may reduce the occurrences of such highly ambiguous cases.

6 Conclusions

An approach to the template-free identification of partially occluded objects in a digital scene has been presented. Objects in the scene are modeled by polygons that approximate their contours up to a given piecewise maximum error measure. Unavoidable ambiguities introduced by the segmentation procedure are accounted for by analyzing the distribution of the polygon edge parameters in the normal space. This analysis is efficiently accomplished by using a structurally adaptive neural network that builds clusters of edge parameters in an unsupervised way. Only broad assumptions are made about the geometric characteristics of the partially occluded objects, avoiding the need of an object template library and the corresponding shape matching algorithm. Application of the proposed method to the detection of fiberglass particles in scanning electron microscopy imagery has shown that the scheme is computationally fast and performs adequately in varying object density conditions, even in the presence of severe background clutter.

Acknowledgments

The data for this study was provided by Schuller, Inc., Denver, Colorado. The authors would like to especially thank Dr. Robert Hamilton for his suggestions and help throughout the research. Also, the primary author wishes to thank the Universidad Autónoma Metropolitana-Iztapalapa, México and the CONACyT, México, for their support of this research.

REFERENCES

[1] J.R. Ullman, "Edge Replacement in the Recognition of Occluded Objects," *Pattern Recognition,* vol. 26, no. 12, pp. 1,771-1,784, 1993.

[2] B. Bhanu and J.C. Ming, "Recognition of Occluded Objects: A Cluster Structure Algorithm," *Pattern Recognition,* vol. 20, no. 2, pp. 199-211, 1987.

[3] M.W. Koch and R.L. Kashyap, "Using Polygons to Recognize and Locate Partially Occluded Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 4, pp. 483-494, Apr. 1987.

[4] O. Yáñez-Suárez and M.R. Azimi-Sadjadi, "Identification and Measurement of Fiber Glass Particles in Electron Microscopy Imagery Using a High-Order Correlation Process," *Review of Progress in Quantitative Non-Destructive Evaluation,* D.O. Thompson and D.E. Chimenti, eds., vol. 16B, pp. 1,463-1,470. New York: Plenum Press, 1997.

[5] R.O. Duda and P.E. Hart, "Use of the Hough Transform to Detect Lines and Curves in Pictures," *Comm. ACM,* vol. 15, 1972.

[6] I.D. Svalbe, "Natural Representations for Straight Lines and the Hough Transform on Discrete Arrays," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 9, Sept. 1989.

[7] O. Yáñez-Suárez and M.R. Azimi-Sadjadi, "Entropy-Driven Structural Adaptation in Sample-Space Self-Organizing Feature Maps for Pattern Classification," *Proc. Int'l Conf. Neural Networks,* no. 1, pp. 287-291, Houston, Tex., 1997.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B.,* vol. 39, pp. 1-38, 1977.

[9] J.S.J. Lee, R.M. Haralick, and L.G. Shapiro, "Morphological Edge Detection," *Int'l J. Robotics and Automation,* vol. RA-3, no. 2, pp. 142-156, 1987.

[10] A.K. Jain, *Fundamentals of Digital Image Processing,* first ed., pp. 362-374. Englewood Cliffs, N.J.: Prentice Hall, 1989.

[11] T. Pavlidis and S.L. Horowitz, "Segmentation of Plane Curves," *IEEE Trans. Computers,* vol. 23, no. 8, pp. 860-870, Aug. 1974.

[12] C.S. Fahn, "An Adaptive Reduction Procedure for the Piecewise Linear Approximation of Digitized Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 9, Sept. 1989.

[13] J. Yuan and C.Y. Suen, "An Optimal O(n) Algorithm for Identifying Line Segments from a Sequence of Chain Codes," *Pattern Recognition,* vol. 28, no. 5, pp. 635-646, 1995.

[14] X. Xie, R. Sudhakar, and H. Zuang, "Corner Detection by a Cost Minimization Approach," *Pattern Recognition,* vol. 26, no. 8, pp. 1,235-1,243, 1993.

[15] T. Kohonen, *Self-Organizing Maps,* first ed. Berlin: Springer Verlag, 1995.

[16] N.K. Bose and P. Liang, *Neural Network Fundamentals with Graphs, Algorithms, and Applications,* first ed. New York, NY: McGraw-Hill, 1996.

[17] G. Deco and D. Obradovic, *An Information-Theoretic Approach to Neural Computing,* first ed., pp. 7-22. New York: Springer-Verlag, 1997.