Pattern Recognition Vol. 16, No. 1, pp. 39–46, 1983.
Printed in Great Britain.

0031-3203/83/010039-08 $03.00/0
Pergamon Press Ltd.

TEXTURE ANALYSIS OF AERIAL PHOTOGRAPHS

RONALD LUMIA, ROBERT M. HARALICK, OSCAR ZUNIGA,
LINDA SHAPIRO, TING-CHUEN PONG and FAR-PEING WANG
Virginia Polytechnic Institute and State University,
Blacksburg, VA 24061, U.S.A.

(*Received* 25 *November* 1981; *in revised form* 4 *March* 1982; *received for publication* 2 *April* 1982)

Abstract—Different image textures manifest themselves by dissimilarity in both the property values and the spatial interrelationships of their component texture primitives. We use this fact in a texture discrimination system.

An image is first segmented into closed regions called units. Then, a set of properties is calculated for each of the units. The units along with their respective properties constitute the primitives.

The discrimination between texture categories has two parts: the training phase and the classification phase. The primitives and the relationships which are obtained from representative training images are used to develop criteria for the classification phase. During classification, the primitives of the image under test are first used to assign a unit to one of several cluster types. Then, each primitive is assigned to the most likely texture class given its cluster type and the cluster types of its spatially adjacent neighbors.

The method is used on three images: a noisy checkerboard, a simulated texture and an aerial photograph.

Texture discrimination Image processing Pattern recognition Contextual classification Decision theory Aerial photographs Scene analysis.

1. INTRODUCTION

Texture discrimination, a common capability for the human vision system, continues to be a difficult problem for machines. In spite of the fact that a precise mathematical definition for texture does not currently exist, many approaches to texture discrimination have been suggested.

Some of the methods for texture discrimination have been statistically based[1–5] while others used a structural approach.[6–9] See Haralick[10] for a review of both approaches.

The reason for texture's importance hinges around one of the most important processes in scene analysis: partitioning an image into meaningful regions. For homogeneous regions, edge detection can accomplish this segmentation. Davis[11] reviews some techniques in edge detection.

This paper describes a method for discriminating between texture classes based on measurements made on typically small regions determined by an initial segmentation of the image into homogeneous regions. Each region and its measurements constitute a primi-
image, the symbolic image, and perhaps some in-
given the values of its measurements and those of its spatially adjacent neighbors. Some of the criteria required for the classification phase are obtained by the analysis of representative samples of texture classes processed during the training phase.

2. CLASSIFICATION PHASE

2.1. *Overview of the classification phase*

The structure of the classification phase is shown in Fig. 1. Image segmentation programs based on the facet model[12,13] are used to process the test image to obtain the segmented symbolic image, an image composed of closed regions, with each pixel having the index number of the region to which it belongs. Each region in the symbolic image is assigned a sequential and unique unit number. Then, the region adjacency graph (RAG), which contains the adjacent unit members for any given unit, is generated. Using the test image, the symbolic image, and aperhaps some intermediate images obtained from the image segmentation process, a property list containing a set of measurements for each unit in the symbolic image is generated. There are currently 26 properties measured for each region, which include:

1. the number of pixels;
2. the maximum gray level;
3. the minimum gray level;
4. the mean;
5. the variance;
6–7. the center of mass;
8. the sum of row × row;
9. the sum of col × col;
10. the sum of row × g (row, col);

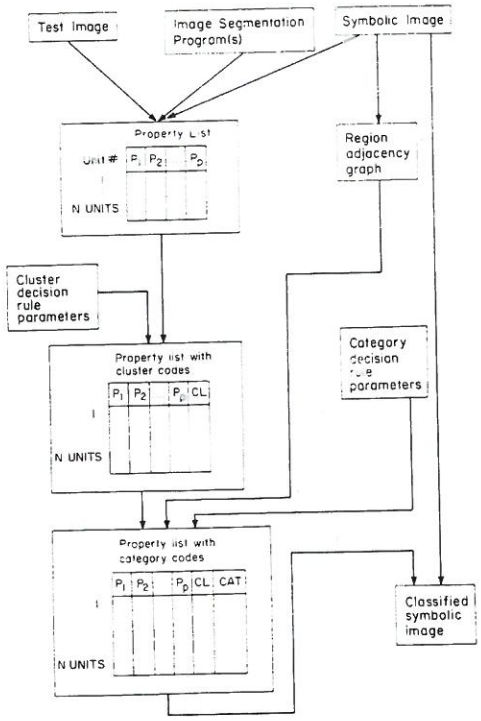

Fig. 1. Classification phase.

11. the sum of col × g (row, col);
12. the sum of row × col;
13–14. eigenvalues of second moment matrix;
15. eccentricity;
16. angle;
17. mean radius;
18. variance of radius;
19. the number of boundary pixels;
20. measure of circularity;
21. row partial derivative;
22. column partial derivative;
23. gradient of a region;
24. slope facet error;
25. third moment;
26. fourth moment.

Using the cluster decision rule parameters calculated in the training phase, each unit is assigned a cluster code. The updated property list, the region adjacency graph and the category decision rule parameters are used to calculate the most likely texture category. Each unit in the symbolic image can then be replaced with its texture category, resulting in the classified symbolic image.

The basic idea is illustrated in Fig. 2. An image is segmented to produce the symbolic image of Fig. 2a. The units are then numbered sequentially, resulting in Fig. 2b. The region adjacency graph contains the adjacency information for each unit. For example, for unit 9, the region adjacency graph says that the adjacent units are 6, 7, 8, 11 and 13. The cluster types for unit 9 and its neighbors are calculated. Then, the probability that unit 9 belongs to category 1 given the cluster types of its neighbors is calculated. This is repeated for each of the texture categories and unit 9 is assigned to the most likely category.

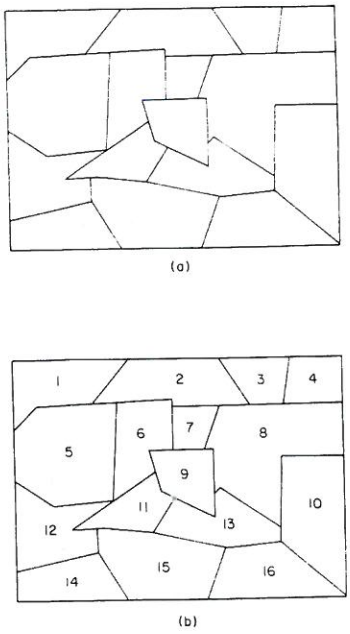

Fig. 2. (a) Segmented image. (b) Labeled symbolic image.

2.2. *Classification phase decision theory*

After an image has been segmented into units, each unit is first assigned a cluster type based on its measured properties and the ideal cluster types generated during the training phase. The most likely texture category is assigned to the unit under test according to (1) the cluster type of the unit under test and (2) the cluster types of the spatially adjacent neighbors to the unit under test. In this section we give a precise mathematical description of this procedure.

Suppose that we have a set of D units $u_1, u_2, \ldots, u_D$. For any given unit u_i, there is a set of N units which are spatially adjacent to u_i. We denote this set by $v(u_i) = \{v_1(u_i), v_2(u_i), \ldots, v_N(u_i)\}$. We first assign u_i as belonging to one of P clusters $\{R_1, \ldots, R_P\}$ based on its property measurements and the cluster decision rule (CDR) obtained from the training phase. Assume, for simplicity, that $CDR(U)$ returns the cluster number of the most likely cluster for unit U. Our goal is to assign u_i to the most likely texture category, CAT_j, given the cluster type of the unit under test and the cluster types of the spatially adjacent neighbors of u_i. Thus, our rule is to assign u_i to CAT_j if

$$P\{CAT_j \mid CDR(u_i), CDR[v(u_i)]\} > P\{CAT_k \mid CDR(u_i), CDR[v(u_i)]\} \quad j \neq k. \quad (1)$$

Using Bayes rule, (1) can be expressed as

$$P\{CAT_j \mid CDR(u_i), CDR[v(u_i)]\} = \frac{P\{CDR[v(u_i)] \mid CDR(u_i), CAT_j\} P\{CDR(u_i), CAT_j\}}{P\{CDR(u_i), CDR[v(u_i)]\}}. \quad (2)$$

Each term of (2) will now be considered separately in order to obtain a more mathematically tractable expression. We assume that the cluster name assignments for neighboring units are uncorrelated conditioned on $CDR(u_i)$ and CAT_j. This is an approximation to reality. With this assumption the first term of (2) can be expressed in terms of the product of the probability density functions of each cluster types as

$$P\{CDR[v(u_i)]|CDR(u_i), CAT_j\} = \binom{N}{n_1 n_2 \ldots n_P} \prod_{L=1}^{P} [P\{R_L|CDR(u_i), CAT_j\}]^{n_L} \quad (3)$$

where R_q is the q-th cluster type, n_q is the number of spatially adjacent neighbors whose cluster type is q, N is the total number of neighbors and P is the number of cluster types. The multinomial coefficient is calculated by

$$\binom{N}{n_1 n_2 \ldots n_p} = \frac{N!}{n_1! n_2! \ldots n_p!}. \quad (4)$$

The next term of (2) can be expressed as

$$P\{CDR(u_i), CAT_j\} = P\{CDR(u_i)|CAT_j\} P\{CAT_j\}. \quad (5)$$

The denominator of (2) will be identical for all classes. As a result, it adds no information to the classification process and can therefore be ignored in the discriminant function. This results in

$$P\{CAT_j|CDR(u_i), CDR[v(u_i)]\} = KP\{CDR[v(u_i)]|CDR(u_i), CAT_j\} P\{CDR(u_i)/CAT_j\} P\{CAT_j\} \quad (6)$$

where K is a constant. Thus equations (1), (3), (4) and (6) are sufficient to classify unit u_i.

3. THE TRAINING PHASE

It is clear from the derivation of (6) that some information is obtained from measurements on the image under test, while other information must be determined by training on known images.

In order to classify a unit as a member of a texture category, the following information is required:

(1). $P\{CAT_j\}$, *the* a priori *probability of category* j. The *a priori* probability of a texture class can be obtained in three ways. First, there may be some texture categories for a given problem which are known *a priori* to be more likely than others. Secondly, this probability may be calculated from the representative training data. Lastly, when no information is available, all categories may be assumed equally likely. In this last case, the $P\{CAT_j\}$ in equation (6) is ignored and its value is absorbed into the constant K.

(2). $P\{CDR(u_i)|CAT_j\}$, *the conditional probability of cluster* $CDR(u_i)$ *given category* j. From some representative sample images, the number of occurrences of each cluster which occur for units known to be from category j can be counted. The normalized result is an estimate of this required conditional probability density function.

(3). $P\{R_q|CDR(u_i), CAT_j\}$, *the conditional probability of cluster* R_q *given that a neighbouring region has cluster type* $CDR(u_i)$ *and category* j. This term is estimated from the training data by calculating the normalized co-ocurrence matrix[14] of cluster types for category j.

This training phase, shown in Fig. 3, is required in order to calculate the parameters necessary for the classification phase, as described in the previous section. The initial processing for the training phase parallels that of the classification phase. From a training image we generate a segmented symbolic image, its region adjacency graph and a region property list.

The main difference between the training phase and the classification phase occurs when the user manually selects regions from the training image which are representative samples of each texture category. On the basis of the user selection, the property list is tagged so that category index i is associated with all units manually declared to belong to the training set for texture class i.

The second training step is clustering. Given all of the sample units and their measurement property vectors belonging to a texture class, a minimal spanning tree can be generated with the subsequent clustering resulting in a relatively small number of cluster types. In other words, a set of prototype clusters for each class is generated. In the classification phase a

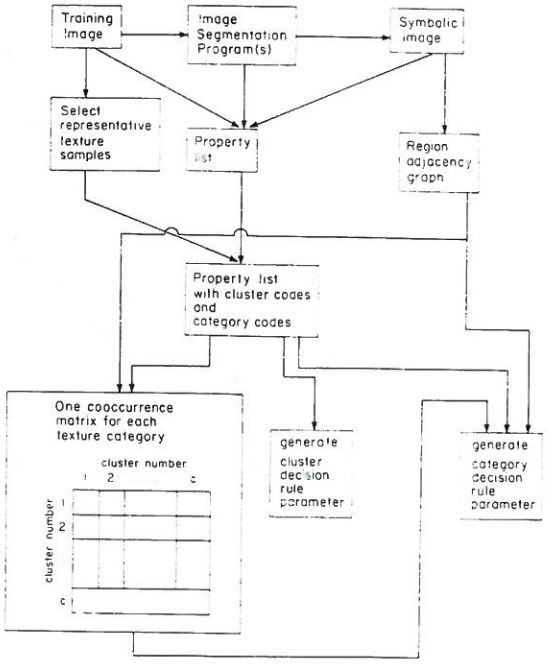

Fig. 3. Training phase.

measurement property vector associated with a unit is used to assign a unit to one of the clusters, based on a criterion such as minimum Euclidean distance to the cluster mean. This constitutes the cluster decision rule.

In the third training step, the property list and the region adjacency graph are used to generate a co-occurrence matrix of cluster vs neighboring cluster for each texture class. The (i, j)-th element of each co-occurrence matrix is the number of times the adjacent units in the training data were labeled cluster i and cluster j for that texture class. Each matrix is then normalized to provide the conditional probabilities of cluster type given cluster type and category.

Thus, the cluster decision rule parameters and the category decision rule parameters, $P\{CAT_j\}P\{R_1|CDR(u_i), CAT_j\}$ and $P\{\text{CDR } (u_i)|CAT_j\}$, are all generated. This is all of the information required for the classification phase.

4. EXPERIMENTAL RESULTS

In this Section we discuss the application of the context classifier on three images: a noisy checkerboard pattern, an artificially generated texture and an aerial photograph. In each case, the goal is to separate different textures. Of course, the types of textures we wish to separate depend on the picture content.

Figure 4 illustrates the experiment for the noisy checkerboard. The upper left corner shows the original image. There are two classes we would like to separate: the first class contains the larger squares and the second class contains the smaller squares. After smoothing to reduce noise, a gradient operator and threshold were used to obtain the edges. The edges defining the closed regions of the symbolic image are shown in the lower left corner of Fig. 4. The training consisted of choosing some large squares for one class

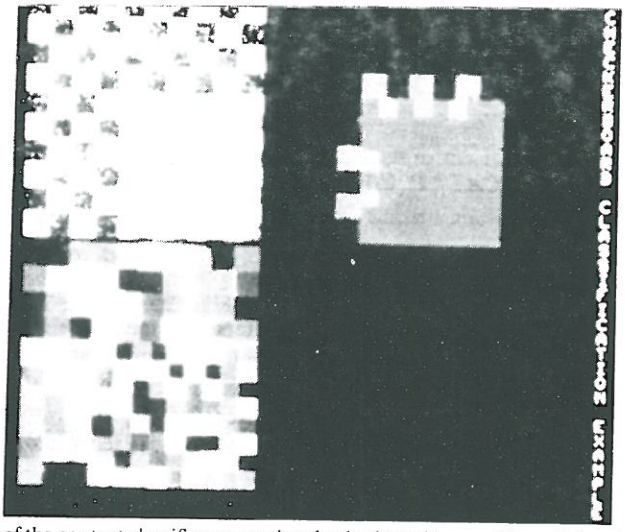

Fig. 4. Application of the context classifier on a noisy checkerboard image. Upper left, original image; lower left, symbolic image; upper right, classification result.

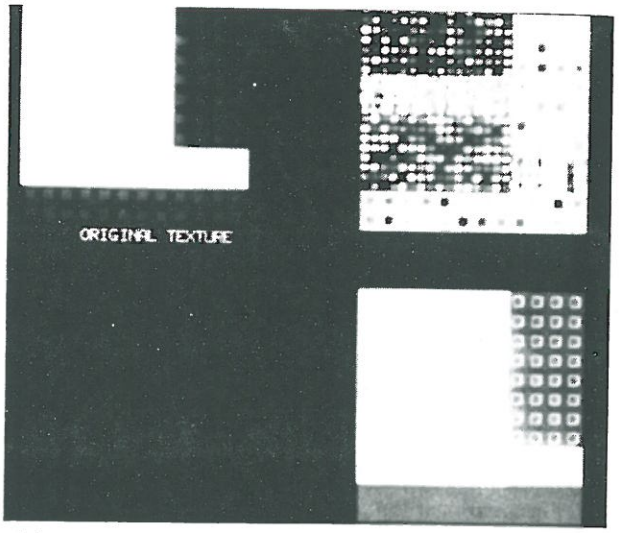

Fig. 5. Application of the context classifier on an artificial texture. Upper left, original image; upper right, symbolic image; lower right, classification result.

and some small squares for the other. Each unit of the symbolic image is then classsified. The upper right corner of Fig. 4 shows the classification results. It is clear that the class of larger squares was separated from the class of smaller squares, with the exception of some units where the two classes meet. These units were rejected by the classifier. Rejection occurs when (1) a unit's properties are grossly dissimilar to the prototype cluster types for all textures clategories or (2) when the probability of the most likely texture class is below a threshold.

The rejected areas between the large squares and small squares in the classified picture resulted from an inadequate edge/region determination. The failure of the gradient method to segment this picture underscored the necessity to use more sophisticated edge detection schemes.

Another artificial texture is shown in the upper left corner of Fig. 5. Next to the texture is its segmentation, (its symbolic image) and below the symbolic image is the classification result. In the training phase, we chose one piece of each texture as its representative sample. During classification, the background of the rightmost texture (which was all one unit) was rejected. This resulted because the training set's background was significantly different from the background of the untrained region for that class and was therefore considered grossly dissimilar.

The last image, shown in Fig. 6, is an aerial image of a complex urban area. The results of the image

Fig. 6. Application of the context classifier on an aerial image of a complex urban area. Original image.

Fig. 7. Application of the context classifier on an aerial image of a complex urban area. Segmented (symbolic) image. Detected edges are overlaid onto the original image.

segmentation are shown in Fig. 7 where the edges are overlaid onto the original image. Two texture classes are sought: large buildings and small buildings. Examples from the training regions for the classes are shown in Fig. 8 and Fig. 9, respectively, where the detected edges are overlaid on top of the original image. We see an excellent correspondence between the detected edges and the original image using a new edge detection procedure which will be discussed in another paper. The results from the classification phase are shown in Fig. 10, where dark grey represents large buildings, light gray indicates small buildings and white is a rejection. The results show that most of the smaller buildings were classified correctly and many of the nonbuildings (roads, parks) were justly rejected. Some of the larger buildings are correctly classified while others are not. Several units were erroneously classified as large buildings. This problem is caused by units being assigned to cluster types which are not sufficiently similar. The Euclidean distance classifier for cluster types needs to be modified to overcome this deficiency.

The computer used for this work was a VAX 11/780, which was run on a time share basis but dedicated to image processing. The system consisted of 1.75 MBytes of RAM and 900 MBytes of disk storage. The emphasis was placed on writing programs which worked, rather than creating programs which ran in

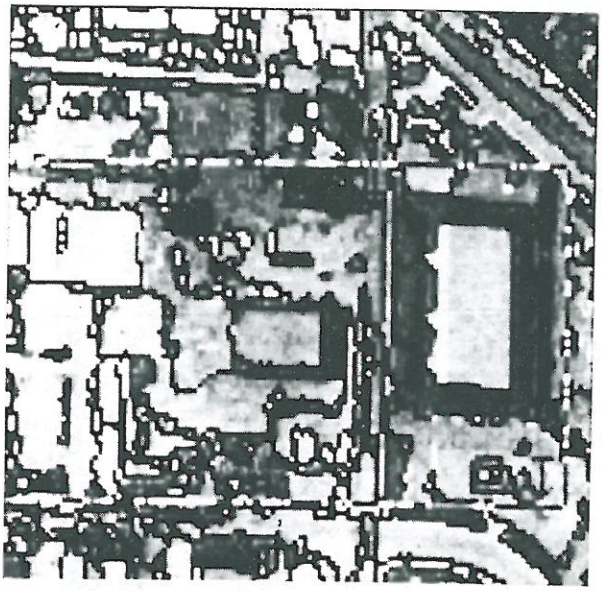

Fig. 8. Application of the context classifier on an aerial image of a complex urban area. Example of a training region for large buildings. Selected edges are overlaid onto the original image.

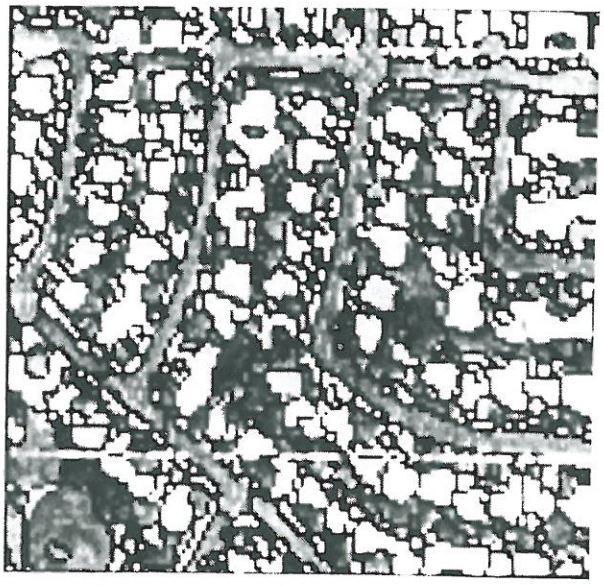

Fig. 9. Application of the context classifier on an aerial image of a complex urban area. Example of a training region for small buildings. Detected edges are overlaid onto the original image.

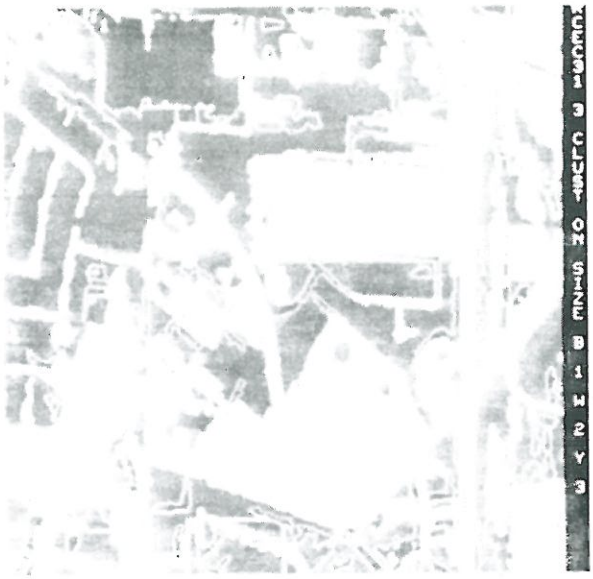

Fig. 10. Application of the context classifier on an aerial image of a complex urban area. Classification result. Dark gray, large buildings; light gray, small buildings; white, rejected.

Table 1. Processing times for the context classifier

	Noisy checkerboard (sec)	Artificial texture (sec)	Aerial photograph (hr: min)
Training	1867.4	4260.2	31:45
Classification	246.3	1881.7	17:22

the most efficient manner. Consequently, a large amount of processing time could eventually be recovered by rewriting the programs. The processing times for training and classification for each of the images are shown in Table 1.

5. CONCLUSION

The segmenting procedure used in our experiments was extremely simple: it was a connected components labeling algorithm performed on non-edge pixels. The fact that this technique worked as well as it did is not a strength for the connected components technique. It is a strength for the edge finding technique. More work is needed to determine optimal ways to clean edges and close gaps. The resulting segmentation should then be an input to a region grower process so that small regions can merge with similar neighbouring regions, if there are any.

Buildings are recognizable by their shape: square corners and straight line sides. They are also recognizable by their context: industrial area, suburban area, etc. Our experiments made no attempt to identify buildings by the shape alone. Such techniques need to be explored.

Our experiments did try to identify the context, not by identification of the texture in an arbitrary square window, but by using a region-co-occurrence-based approach that used no predetermined artificial boundaries. More work is needed to make this kind of approach more computationally efficient and to give it a higher degree of identification accuracy.

REFERENCES

1. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene analysis, *IEEE Trans. Comput.* **C-20,** 562–569 (1971).
2. R. Sutton and E. Hall, Texture measures for automatic classification of pulmonary disease, *IEEE Trans. Comput.* **C-21,** 667–676 (1972).
3. A. Rosenfeld and E. Troy, Visual texture analysis, Technical Report 70-116 of the University of Maryland (1970).
4. R. M. Haralick, K. Shanmugan and I. Dinstein, On some quickly computable features for texture, *Proceedings of the 1972 Symposium on Computer Image Processing and Recognition* Vol. 2, pp. 12-2-1–12-2-10, Columbia, MO (1972).
5. J. Weszka, C. Dyer and A. Rosenfeld, A comparative study of texture measures for terrain classification *IEEE Trans. Syst. Man Cybernet.* **SMC-6,** 269–285 (1976).
6. L. Carlucci, A formal system for texture languages, *Pattern Recognition* **4,** 53–72 (1972).
7. S. Y. Lu and K. S. Fu, A syntactic approach to texture analysis, *Comput. Graphics Image Process.* **7,** 303–330 (1978).
8. S. Zucker, Toward a model of texture, *Comput. Graphics Image Process.* **5,** 190–202 (1976).
9. S. Tsuji and F. Tomita, A structural analysis for a class of textures, *Comput. Graphics Image Process.* **2,** 216–231 (1973).

10. R. M. Haralick, Statistical and structural approaches to texture, *Proc. IEEE*, **67**, 786–804 (1979).
11. L. S. Davis, A survey of edge detection techniques, *Comput. Graphics Image Process.* **4**, 248–270 (1975).
12. R. M. Haralick and L. Watson, A facet model for image data, *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, pp. 489–497, Chicago, IL (1979).
13. R. M. Haralick, Edge and region analysis for digital image data, *Comput. Graphics Image Process.* **12**, 60–73 (1980).
14. L. Davis, S. Johns and J. K. Aggarwal, Texture analysis using generalized co-occurrence matrices, *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, pp. 313–318, Chicago, IL (1978).

About the Author—RONALD LUMIA was born in Bay Shore, New York, on 27 September 1950. He received the BS degree in Electrical Engineering from Cornell University, Ithaca, New York, in 1972, and the MS and PhD degrees from the University of Virginia, Charlottesville, in 1977 and 1979, respectively.

From 1972 to 1975, he was employed as a Design Engineer with Teradyne, Inc. From 1975 to 1978, he worked as a Graduate Research Assistant at the University of Virginia on the application of charge-coupled device discrete analog processors to pattern recognition systems. From 1978 to 1980, he was with the faculty of Ecole Superieure d'Ingenieurs en Electrotechnique et Electronique, Paris, France. In 1980, he joined the faculty of Virginia Polytechnic Institute and State University, Blacksburg, where he is presently an assistant professor of Electrical Engineering. His research interests include image processing, pattern recognition and speech analysis.

Dr. Lumia is a member of Eta Kappa Nu and Tau Beta Pi.

About the Author—ROBERT M. HARALICK was born in Brooklyn, New York, on 30 September 1943. He received the BS degree from the University of Kansas in 1966. He has worked with Autonetics and IBM. In 1965, he worked for the Center for Research, University of Kansas, as a research engineer and in 1969, when he completed his PhD at the University of Kansas, he joined the faculty of the Electrical Engineering Department, where he served as a Professor from 1975 to 1978. In 1979, Dr. Haralick joined the faculty of the Electrical Engineering Department at Virginia Polytechnic Institute and State University, where he is now a Professor and director of the Spatial Data Analysis Laboratory. Dr. Haralick has done research in pattern recognition, multi-image processing, remote sensing, texture analysis, data compression, clustering, artificial intelligence and general systems theory. He is responsible for the development of GIPSY (General Image Processing System), a multi-image processing package which runs on a minicomputer system.

He is a member of the Institute of Electrical and Electronic Engineers, the Association for Computer Machinery, Sigma Xi, the Pattern Recognition Society and the Society for General Systems Research.

About the Author—LINDA G. SHAPIRO was born in Chicago, Illinois, in 1949. She received the BS degree in Mathematics from the University of Illinois at Urbana-Champaign in 1970 and the MS and PhD degrees in Computer Science from the University of Iowa, Iowa City, in 1972 and 1974, respectively.

She was an Assistant Professor of Computer Science at Kansas State University, Manhattan, from 1974 to 1978. She is currently an Associate Professor at Virginia Polytechnic Institute and State University, Blacksburg. Her research interests include scene analysis, pattern recognition, spatial information systems, computer graphics and data structures. She has completed an undergraduate textbook on data structures with R. Baron.

Dr. Shapiro is a member of the IEEE Computer Society, the Association for Computing Machinery, the Pattern Recognition Society and the American Association for Artificial Intelligence. She is currently co-editor of the newsletter for the IEEE Technical Committee on Machine Intelligence and Pattern Analysis.

About the Author—TING-CHUEN PONG was born in Hong Kong on 27 June 1957. He received the BS degree in Mathematics and Physics from the University of Wisconsin, Eau Claire, in 1978 and the MS degree in Computer Science in 1981 from Virginia Polytechnic Institute and State University, Blacksburg. He is currently a doctoral candidate in the Department of Computer Science at Virginia Polytechnic Institute and State University.

About the Author—OSCAR A. ZUNGA was born in Callao, Peru, on 29 September 1953. He received the BS degree in Electrical Engineering from the University Nacional de Ingenieria, Lima, Peru, in 1975 and the MS degree in Electrical Engineering from the University of Kansas, Lawrence, Kansas, in 1979. He is currently working towards the PhD degree in Electrical Engineering at Virginia Polytechnic Institute and State University, Blacksburg. During 1978, he was involved in research in image data compression at the University of Kansas. Since 1979 he has been a research assistant at VPI & SU where he is presently involved in research in image processing and computer vision.

About the Author—FAR-PEING WANG was born in Taiwan on 3 August 1951. He received the BS degree in Mathematics in 1974, from Chung-Young University, Taiwan, and the MS degree in Mathematics in 1976, from Wright State University. He is currently a PhD student in the Computer Science Department at Virginia Polytechnic Institute and State University.