Learning Texture Discrimination Rules in a Multiresolution System

H. Greenspan, R. Goodman, R. Chellappa, and C. H. Anderson

Abstract—We describe a texture analysis system in which informative discrimination rules are learned from a multiresolution representation of the textured input. The system incorporates unsupervised and supervised learning via statistical machine learning and rule-based neural networks, respectively. The textured input is represented in the frequency-orientation space via a log-Gabor pyramidal decomposition. In the unsupervised learning stage a statistical clustering scheme is used for the quantization of the feature-vector attributes. A supervised stage follows in which labeling of the textured map is achieved using a rule-based network. Simulation results for the texture classification task are given. An application of the system to real-world problems is demonstrated.

I. Introduction

We describe a hybrid texture analysis system that incorporates the advantages of learning paradigms, including statistical machine learning, knowledge-based systems and neural networks, in the context of multi-resolution feature extraction techniques. The main goal of the system is to learn a minimal representation for a given library of textures, based on which one can successfully classify and segment new mosaic test images into homogeneous textured regions. Of particular interest is to apply the system to noisy images arising in real-world computer-vision problems.

The main features of the system are the following: A multiresolution pyramid is used for a computationally efficient feature-extraction scheme. The important characteristics of the input domain are then learned from examples, with both unsupervised and supervised learning techniques utilized. An information theoretic technique enables the characterization of the most informative correlations between the input features and the texture class specification. The learned correlations are specified as discrimination rules which provide probability estimates for the output classes rather than just a hard-decision label. These probability estimates can be used for higher-level analysis, such as feedback for smoothing and the learning of an unknown class, the so called "pattern discovery" problem [1]. The learned rules are available to the user and can enhance his or her knowledge of the input domain and the classification task at hand. Finally, the learned rules can be mapped onto a rule-based neural network and thus the classification scheme is parallelizable and suitable for implementation using special purpose neural-network hardware.

The system consists of three major stages, as shown in Fig. 1. The first stage performs feature extraction and transforms the image space into an array of 15-dimensional feature vectors, each vector corresponding to a local window in the original image. The learning mechanism shown next derives a minimal subset of the above filters which conveys sufficient information about the visual input for its differentiation and labeling. We reduce the feature space both in the unsupervised and supervised stages of analysis. In the unsupervised stage a machine-learning clustering algorithm is used to quantize the continuous input features. The supervised learning stage follows in which labeling of the input domain is achieved using a rule-based network. Ultimately, a minimal representation for a library of patterns is learned in a training mode, following which the classification of new patterns is achieved. The texture-analysis task is defined next followed by the system description and simulation results.

II. The Texture Analysis Task

Visual texture is one of the most fundamental properties of a visible surface. It participates as one of the major modalities which help us in the understanding of our visual environment. As such it takes part in lower-level to higher-level tasks, from scene segmentation to object recognition. Texture-analysis methods can be utilized in a variety of application domains, such as remote sensing, automated inspection, medical image processing and advanced image-compression schemes. The different textures in an image are usually very apparent to a human observer (see Fig. 2), but no good mathematical definition can encapture the very diverse texture family. It is this lack of definition that makes automatic description or recognition of these patterns a very complex and as yet an unsolved problem.

Although researchers approach texture differently, most would agree that the texture family can be categorized into two main categories—*structured* and *unstructured*, more *stochastic* textures. Methods that can handle the more structured textures use structural models of texture which assume that textures are composed of texture primitives. The texture is produced by the placement of these primitives according to certain placement rules [2]. One needs to be able to define a priori a good set of primitives and placement rules (a tree grammar is commonly used) in order to characterize the textured input. This approach can handle very regular patterns. Some textures which can be handled in this manner are shown in Fig. 2 (top row).

Stochastic models, such as the Markov Random Field (MRF) models, are used as methods to handle unstructured or stochastic textures. Here the image is seen as an instance of a random process, defined via the model parameters [3]. The model parameters need to be estimated in order to define adequately the perceived qualities of the texture. This model-based technique can capture certain textures very well (see bottom row of Fig. 2), but they fail with the more regular textures as well as inhomogeneous ones.

The methods discussed above use the pixel-based domain as their input space. Other methods exist in the literature (e.g., [4], [5]) including this work, which compute texture features from filtered images, and use these filtered characteristics in the classification or segmentation tasks. For a review of the texture analysis field, its applications and the different methods available in the literature, see [6]. Although texture analysis has been a subject of intense study by many researchers, it is as yet an open challenge to achieve a high percentage classification rate on all the above textures within one framework. Such a challenge is our ultimate goal.

In this work, we wish to demonstrate the application of a learning system to the texture-analysis task. In this approach, the important characteristics of the input domain are learned from examples, rather

Manuscript received August 1, 1992; revised November 30, 1993. This work was supported in part by Pacific Bell, and in part by ARPA and ONR under grant no. N00014-92-J-1860. H. Greenspan was supported in part by an Intel fellowship. This research was carried out in part by the Jet Propulsion Laboratories, California Institute of Technology. Professor R. Chellappa's work was supported in part by an NSF Grant MIP 91-00655.

H. Greenspan and R. Goodman are with the Department of Electrical Engineering, 116-81 California Institute of Technology, Pasadena CA 91125.

R. Chellappa is with the Department of Electrical Engineering, Institute for Advanced Computer Studies and Center for Automation Research, University of Maryland, College Park, MD 20742 USA.

C. H. Anderson was with the Jet Propulsion Laboratory 168-522, California Institute of Technology, Pasadena CA 91109 USA. He is now with the Department of Anatomy and Neurobiology, Box 8108, Washington University, School of Medicine, 606 S. Euclid Ave, St. Louis MO 63110 USA.

IEEE Log Number 9403775.

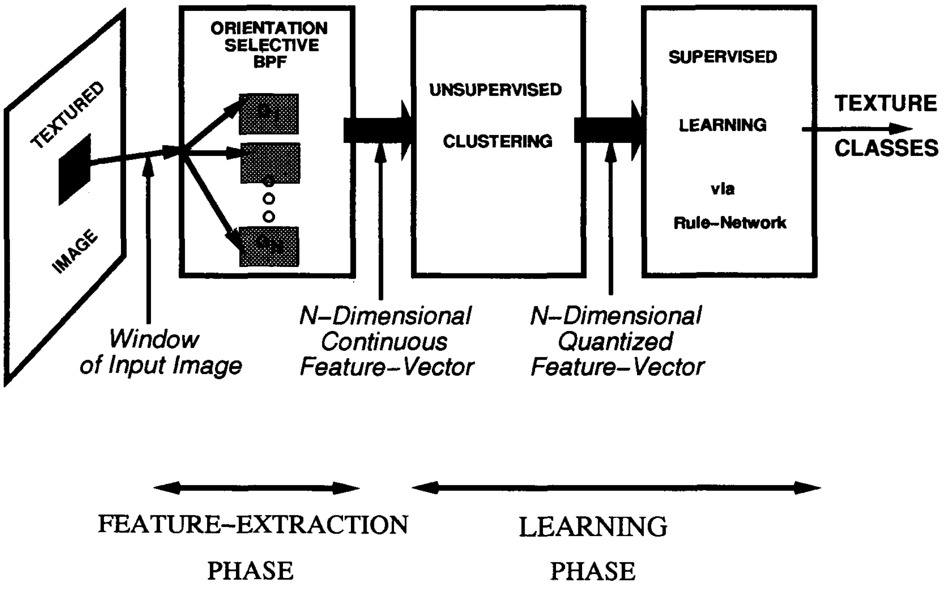
TEXTURED
IMAGE
Window of Input Image
ORIENTATION SELECTIVE BPF
N-Dimensional Continuous Feature-Vector
UNSUPERVISED CLUSTERING
N-Dimensional Quantized Feature-Vector
SUPERVISED LEARNING via Rule-Network
TEXTURE CLASSES
FEATURE-EXTRACTION PHASE
LEARNING PHASE

Fig. 1. System block diagram.

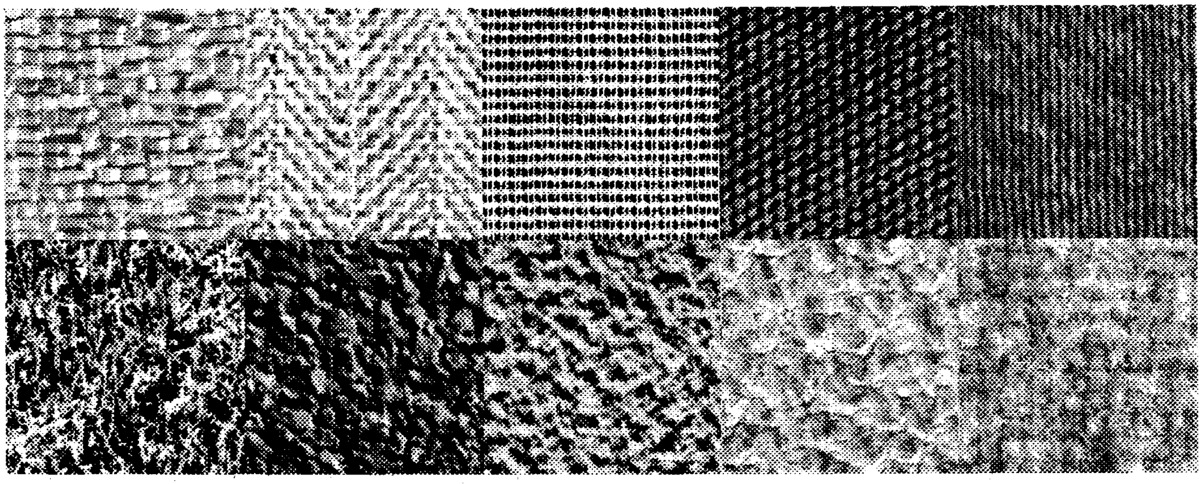

Fig. 2. Structured (top row) and Unstructured (bottom row) texture examples. Top row (left to right): raffia, herringbone weave, canvas1, canvas2, jeans. Bottom row (left to right): grass, cork, handmade paper, pigskin, cloth.

than specified a priori via parametric model-based schemes such as the structural or stochastic models mentioned above. In the following sections the system is described.

III. Feature Extraction Stage

In the texture-analysis task there is both biological and computational evidence supporting the use of a bank of orientation-selective bandpass filters for the feature extraction phase [4], [5], [7]. Orientation and frequency responses are extracted from local areas of the input image and the statistics of the coefficients characterizing the local area form the representative feature vector. In this work, we use the log-Gabor pyramid [8], or the Gabor wavelet decomposition, to define an initial finite set of filters.

A computationally efficient filtering scheme is used based on a pyramidal approach. In a pyramid representation the original image is decomposed into sets of low-pass and band-pass components via Gaussian and Laplacian pyramids, respectively [9]. The Gaussian pyramid consists of low-pass filtered (LPF) versions of the input image, with each stage of the pyramid computed by low-pass filtering of the previous stage and corresponding subsampling of the filtered output. The Laplacian pyramid consists of band-pass filtered (BPF) versions of the input image, with each stage of the pyramid constructed by the subtraction of two corresponding adjacent levels of the Gaussian pyramid. We use the Filter-Subtract-Decimate (FSD) Laplacian pyramid [10], which is a variation on the Burt and Adelson Laplacian pyramid [9]. In the following we refer to the input image as G_0, the LPF versions are labeled G_1 thru G_N with decreasing resolutions and the corresponding BPF versions are labeled L_0 thru L_N respectively. A recursive procedure allows for the creation of the FSD pyramid, as follows:

$$G_{n+1}^0 = W * G_n; L_n = G_n - G_{n+1}^0;$$
$$G_{n+1} = \text{Subsampled } G_{n+1}^0. \quad (1)$$

The LPF, W, is Gaussian in shape, normalized to have the sum of

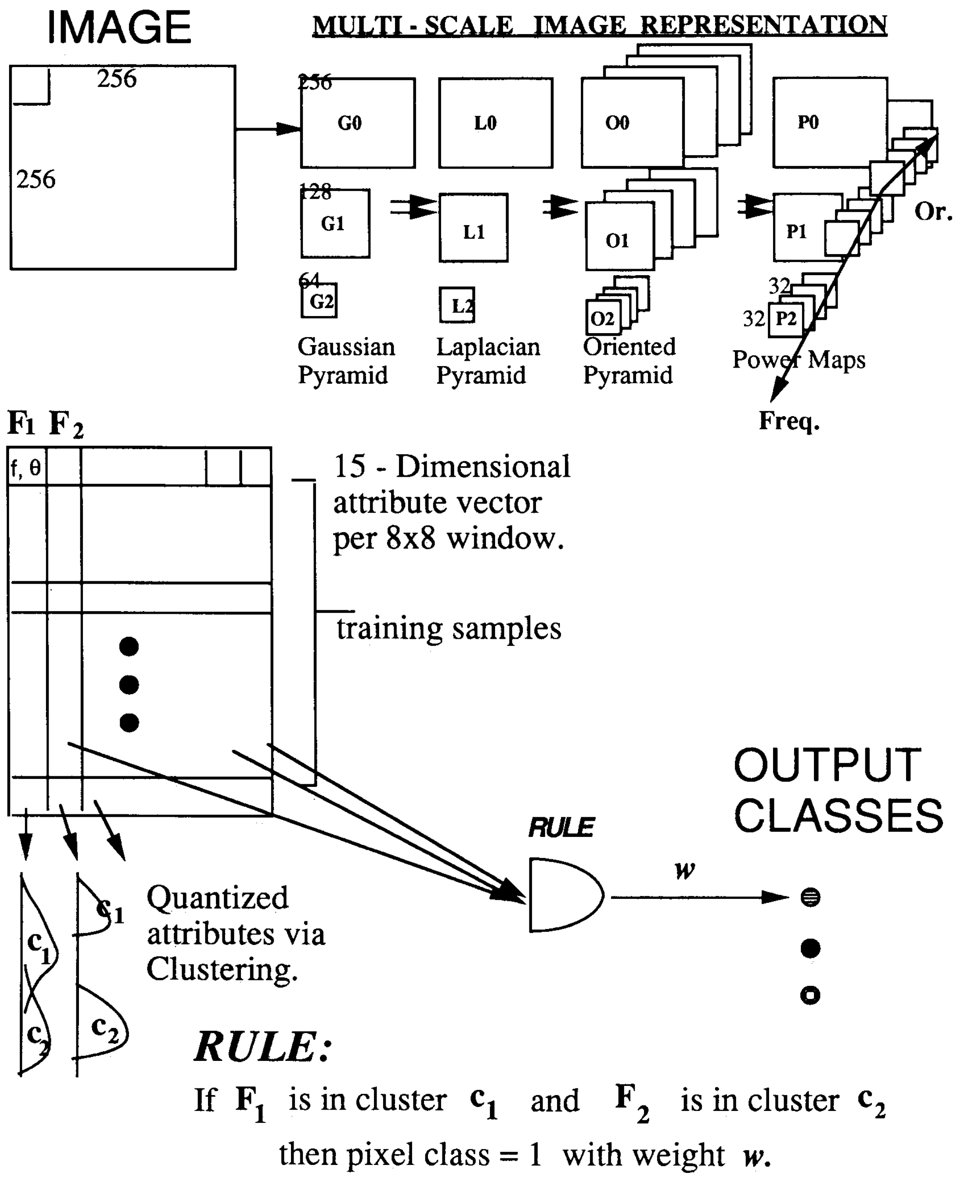
IMAGE
256
256
MULTI - SCALE IMAGE REPRESENTATION
256
G0
L0
O0
P0
128
G1
L1
O1
P1
Or.
64
G2
L2
O2
32
32
P2
Gaussian Pyramid
Laplacian Pyramid
Oriented Pyramid
Power Maps
Freq.
F1 F2
f, θ
15 - Dimensional attribute vector per 8x8 window.
training samples
OUTPUT CLASSES
RULE
w
c1
c2
c1
c2
Quantized attributes via Clustering.
RULE:
If F1 is in cluster c1 and F2 is in cluster c2
then pixel class = 1 with weight w.

Fig. 3. Detailed sketch of the system.

its coefficients equal to 1. The values used in this work for W, which is a 5-sample separable filter, are (1/16, 1/4, 3/8, 1/4, 1/16).

In order to extract the orientationally tuned band-pass filtering responses, the oriented pyramid is formed next. The oriented pyramid is the result of modulating each level of the Laplacian pyramid with a set of oriented sine waves, followed by the same LPF operation used above, and corresponding subsampling, as defined in (2)[1]:

$$O_{n\alpha} = \text{LPF}[e^{(i\vec{k}_\alpha \cdot \vec{r})} L_n[x, y]] \qquad (2)$$

where $\vec{r} = x\vec{i} + y\vec{j}$ (x and y are the indices of the Laplacian image),

$$\vec{k}_\alpha = (\pi/2)[\cos\theta_\alpha \vec{i} + \sin\theta_\alpha \vec{j}]$$

and

$$\theta_\alpha = (\pi/4)\alpha; (\alpha = 0, 1, 2, 3).$$

Here, n is the scale coefficient and α represents the orientation coefficient. A three-scale pyramid is utilized ($n = 0, 1, 2$). Each level of the pyramid is multiplied with four sine waves at four orientations (0, 45, 90, 135 degrees). The orientation and frequency bandwidth of each orientationally tuned bandpass filter is thus 45 degrees and 1 octave, respectively. For a theoretical analysis of the pyramid filters' characteristics the reader is referred to [11], [12].

It is the local statistics of the oriented pyramid's coefficients which characterize the image local-area response to the different orientations and frequencies. A measure of power associated with each filtered map is defined next as the nonlinear operation given below:

$$P_{n\alpha} = |O_{n\alpha}|. \qquad (3)$$

[1] This filtering operation is not the standard one found in the literature. Most modulate the filter and then perform a convolution with the image. Here we propose a reversal in the order of operations for a computationally efficient filtering scheme.

TABLE I
CLASS-CONFUSION MATRIX

	raffia	herring	canvas1	canvas2	jeans	grass	cork	hmpaper	pigskin	cloth
raffia	**91.02**	0	0	0	0	0	0	0	8.98	0
herr	13.28	**84.76**	0	1.95	0	0	0	0	0	0
canvas1	0	0	**100**	0	0	0	0	0	0	0
canvas2	0	0	0	**100**	0	0	0	0	0	
jeans	0	0	0	0	**100**	0	0	0	0	0
grass	0	0	0	0	0	**100**	0	0	0	0
cork	0	0	0	0	0	7.03	**92.58**	0.39	0	0
hmpaper	4.68	0	0	0	0	0	4.29	**91.01**	0	0
pigskin	3.51	0	0	0	2.34	0	0	0	**86.72**	7.42
cloth	1.17	0	0	0	0	0	0	0	1.56	**97.26**

The mean values of the power maps, at three scales and four orientations, together with the power of the nonoriented component L_n at each scale, produce 15 feature maps which we use to represent the input image. Gaussian pyramids are formed on each of the power maps, $P_{n\alpha}$. The Gaussian levels of the $P_{n\alpha}$ which have the same size as the lowest spatial frequency power map are used to form the feature vector. These generate reduced-size feature maps, with the features corresponding to the average of the power of the image's local response to specific orientation and frequency ranges. Using 15 equal-sized feature maps generated in this way, each $8 * 8$ local window of the input image gets mapped to a 15-dimensional feature vector as the output of the feature extraction stage (see Fig. 3).

IV. THE LEARNING SYSTEM

The goal of the learning system is to use the feature representation described above to discriminate between the input patterns, or textures. Both unsupervised and supervised learning stages are utilized. A minimal set of features are extracted from the 15-dimensional attribute vector, which convey sufficient information about the visual input for its differentiation and labeling. We thus use the learning system to extract the most important features.

A. Unsupervised Clustering

The unsupervised learning stage can be viewed as a preprocessing stage for achieving a more compact representation of the filtered input. The goal is to quantize the continuous valued features which are the result of the initial filtering, thus shifting to a more symbolic representation of the input domain.

The output of the filtering stage consists of 15 continuous-valued feature maps. Thus, each local area of the input image is represented via a 15-dimensional feature vector. An array of such vectors, viewed across the input image, is the input to the learning stage. A detailed sketch of the system is presented in Fig. 3.

We wish to detect characteristic behavior, across the 15-dimensional feature space, for the family of textures to be learned. In this work, each dimension out of the 15-dimensional attribute vector is individually clustered. All samples are thus projected onto each axis of the space and one-dimensional clusters are found using the K-means clustering algorithm [13]. This statistical clustering technique consists of an iterative procedure of finding K means in the sample space, following which each input sample is associated with the closest mean in Euclidean distance. The means, labeled $0, 1, 2, \cdots, K-1$ arbitrarily, correspond to discrete codewords. Each continuous-valued input sample gets mapped to the discrete codeword representing its associated mean. The output of this preprocessing stage is a 15-dimensional quantized vector of attributes which is the result of concatenating the discrete-valued codewords of the individual dimensions. Each dimension can be seen to contribute a probabilistic differentiation onto the different classes via the clusters found. As some of the dimensions are more representative than others, it is the goal of the supervised stage to find the most informative dimensions for the desired task (with the higher differentiation capability) and to label the combined clustered domain.

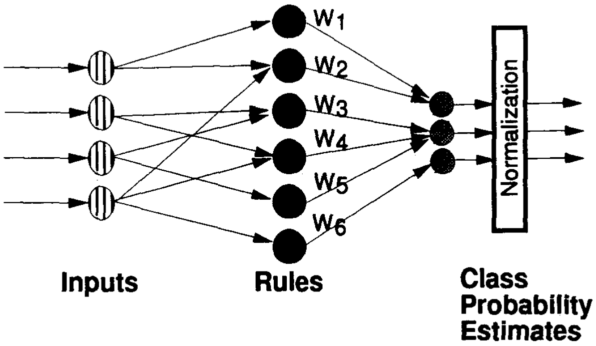

Fig. 4. Rule-based network.

In the algorithm implemented here, K was chosen as the number of output classes we wish to learn. A different value of K could have been chosen with comparable results [11]. The fact that this is a preprocessing step, prior to the rule-based network classification, reduces substantially the difficulty of picking an appropriate number of clusters.

B. Supervised Learning via a Rule-Based System

The goal of the supervised stage is to classify the input image, while finding the most informative input dimensions, or attributes, for the desired task, thus reducing the dimensionality of the representation. We wish to learn a classifier which maps the output features of the unsupervised stage to the texture class labels. Any classification scheme could be used. However, we utilize rule-based information-theoretic approach (ITRULE) which is an extension of a first-order Bayesian classifier, because of its ability to output probability estimates for the output classes. The classifier defines correlations between input features and output classes as probabilistic rules of the form: If $Y = y$ then $X = x$ with probability P. Here, $Y = (Y_1, \cdots, Y_N)$ represents the attribute vector and X is the set $(x_1, \cdots, x_m)$ of m possible output classes. In this work, $N = 15$ and m is the number of texture classes learned. Given an initial

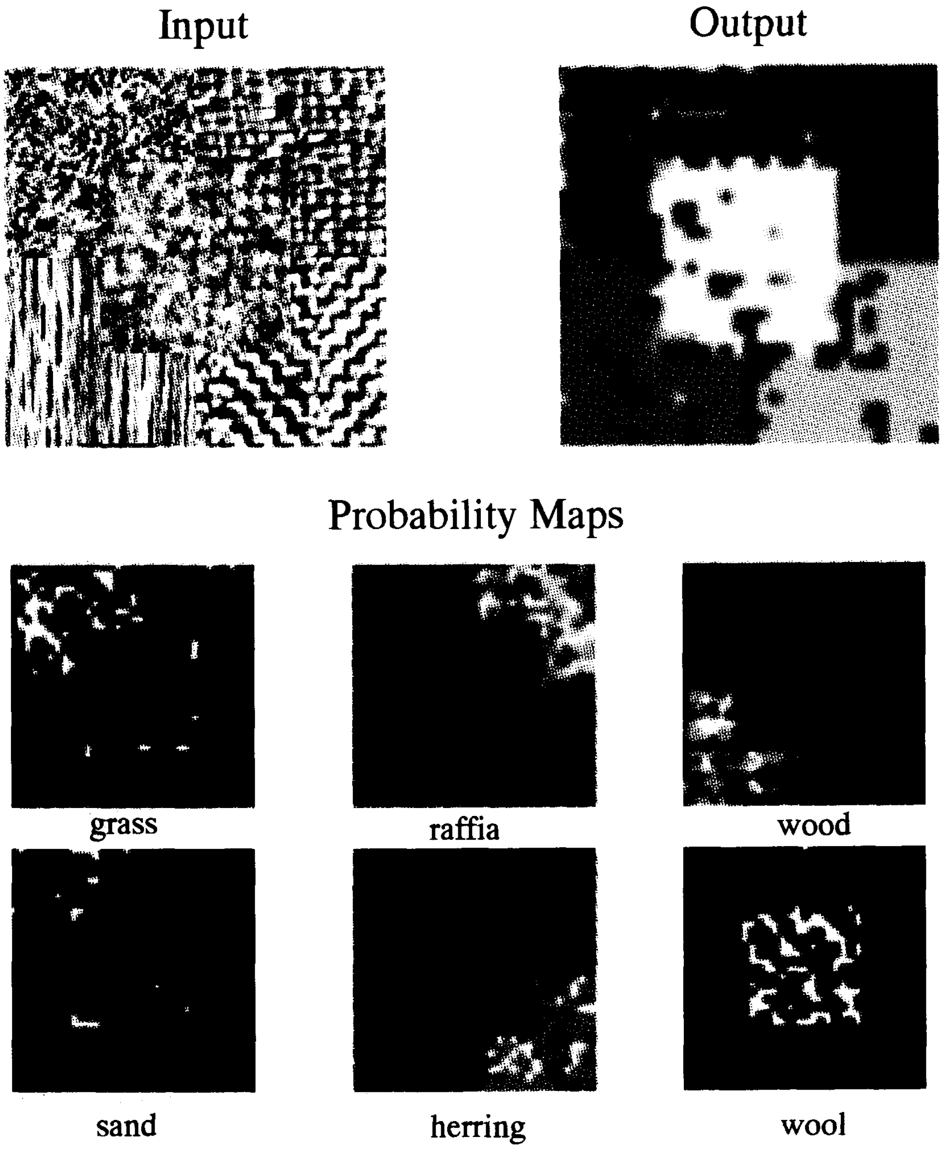

Fig. 5. Five class natural texture classification. Input mosaic is presented (top left), followed by the labeled output map (top right) and probability maps (bottom).

labeled training set of examples, where each example is of the form $(Y_1 = y_1, \cdots, Y_N = y_n, X = x_i)$, the system learns a classifier such that when presented with future test attribute vectors, it estimates the posterior probability of each class. A data-driven supervised learning approach utilizes an information theoretic measure to learn the most informative links or rules between features and class labels. Such a measure was introduced as the J measure [14] defined as

$$J(X = x; y) = p(Y = y)\left(p(x \mid y)\log\left(\frac{p(x \mid y)}{p(x)}\right) + (1 - p(x \mid y))\log\left(\frac{1 - p(x \mid y)}{1 - p(x)}\right)\right). \quad (4)$$

Here, the information content of a rule is represented as the average amount of information that attribute values y give about the class X. The J measure has several desirable properties as a rule information measure. It is comprised of two main terms. The first is $p(Y = y)$, the probability that the particular set of attribute values will occur. The second term is the cross-entropy of X and X given y. This is a measure of the goodness of fit between the *a posteriori* belief about X and the *a priori* belief. Maximizing the product of the two terms is equivalent to simultaneously maximizing both the simplicity of the specific correlation vector, Y, and the goodness of fit to the perfect predictor of X. The J-measure is used in a search algorithm to search the space of all possible rules relating the attributes to the class, X, and produce a ranked set of the most informative rules which classify X. For details about the rule-extraction algorithm see [15].

The most informative set of rules via the J measure is learned in a training stage, following which the classifier uses them to provide an estimate of the probability of a given class being true. When presented

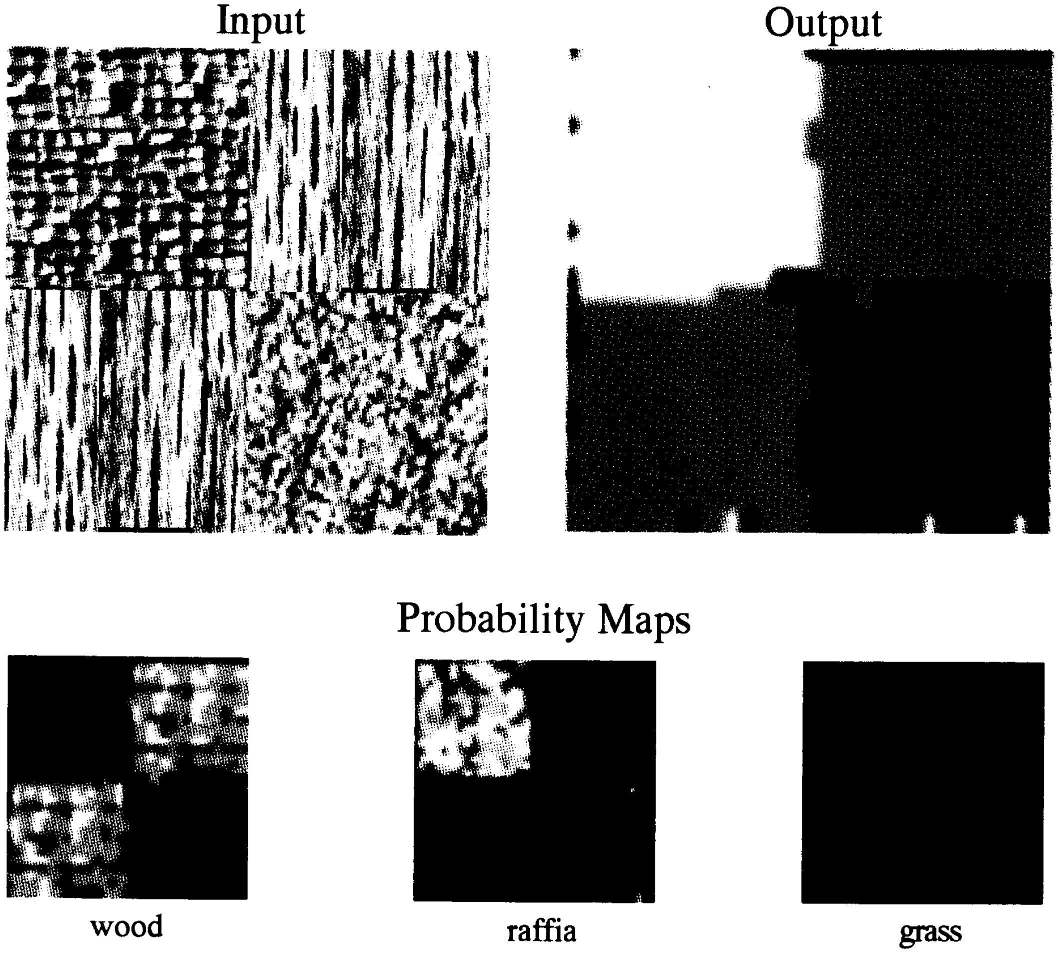

Fig. 6. Generalization to an unknown class. Presented is the input mosaic (top left), followed by the output labeled map (top right) and the probability maps (bottom).

with a new input evidence vector, Y, a set of rules can be considered to "fire." These are a subset of the correlations learned which the input attribute vector matches. Using Bayes' rule, the classifier estimates the log posterior probability of each class given the rules that fire [15]:

$$\log p(x \mid \text{rules that fire}) = \log p(x) + \sum_j W_j \tag{5}$$

with

$$W_j = \log\left(\frac{p(x \mid y)}{p(x)}\right),$$

where $p(x)$ is the prior probability of the class x, and W_j represents the evidential support for the class as provided by rule j. In the absence of any rules firing, the estimate of each class is given by the bias value, namely the log of the prior probability of the class. Given a set of rules which fire, each rule contributes a weight to its corresponding output class. A positive weight implies that the class is true, while a negative weight implies it is false. The W_js provide the user with a direct explanation of how the classification decision was arrived at. Each class estimate can now be computed by accumulating the "weights of evidence" incident on it from the rules that fire. This can be done in a parallel manner. The largest estimate is chosen as the initial class label decision. The probability estimates for the output classes can now be used for feedback purposes for spatial smoothing and further higher-level processing.

The rule-based classification system can be mapped into a three-layer feed-forward architecture as shown in Fig. 4. The input layer contains a node for each attribute. The hidden layer contains a node for each rule and the output layer contains a node for each class. Each rule (second layer node j) is connected to a class via the multiplicative weight of evidence W_j. This hybrid rule-based neural model combines the explicit knowledge representation in the form of rules with the parallel implementation of neural-network architectures.

V. Simulation Results

We present the result of applying the above-described system to textured images. The system was tested on both structured and unstructured natural textures, taken from the Brodatz library of natural textures [16], as well as on natural scenery.

An example of a five-class natural texture classification is shown in Fig. 5. The mosaic is comprised of grass, raffia, herringbone weave, wood and wool (center square) textures. The input mosaic is presented (top left), followed by the labeled output map (top right) and the corresponding probability maps for a prelearned library of six

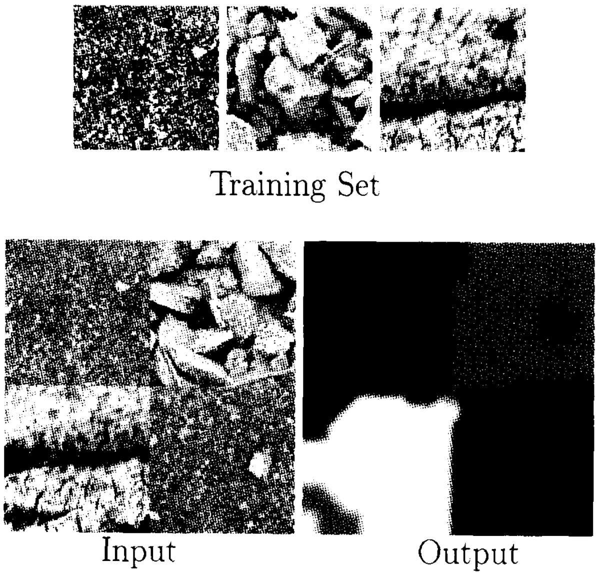

Fig. 7. Natural-scene analysis: 3-texture case. The training set, composed of gravel, rock and wood textures, is presented (top). It is followed by an input test image (bottom left) and the corresponding output label map (bottom right).

textures (grass, raffia, wood, sand, herringbone weave, and wool, left to right, top to bottom, respectively). The input poses a very difficult task which is challenging even to humans. Based on the probability maps (with white indicating probability closer to 1) the very satisfying result of the labeled output map is achieved. The five different regions have been identified and labeled correctly (in different shades of gray) with the boundaries between the regions very strongly evident. It is worth noting that the probabilistic approach enables the analysis of both structured textures (such as the wood, raffia and herringbone weave) and unstructured textures (such as the grass and wool).

Table I presents the class-confusion matrix for the 10 textures of Fig. 2. Very high percentage classification rates are achieved for both the structured (top row) and the unstructured (bottom row) textures. The training set consists of one 128* 128 image for each texture. Three other images are used in testing. Four runs were made, each with a different training input, and the averages of these four runs are listed in the table. Note that the classification rates are based on labeling correctly $8 * 8$ windows . This defines a very high-resolution classification strategy which enables segmentation as well as recognition.

Fig. 6 demonstrates the capability of the system to generalize to the identification of an unknown class. In this task a presented pattern, which is not part of the prelearned library, is to be recognized as such and labeled as an unknown area of interest. This task is termed "pattern discovery" and its application is widespread, from identifying unexpected events to the selection of areas of interest in scene exploration studies. Learning the unknown is a difficult problem in which the probability estimates prove to be valuable. Our criterion for declaring an unknown class is when the sum of W_j's (5) is negative for each class; i.e. there is negative evidence for each prelearned class. In the presented example, a three texture library was learned, consisting of the wood, raffia and grass textures. The input consists of wood, raffia and sand (top left). The output label map (top right) which is the result of the analysis of the respective probability maps (bottom) exhibits the accurate detection of the known raffia and wood textures, with the sand area labeled in black as an unknown class. This conclusion was based on the negative weights of evidence for each of the prelearned classes—indicated as zero probability in the corresponding probability maps. We have thus successfully analyzed the scene based on the existing source of knowledge.

The application of the texture discrimination system to natural scene analysis has been pursued [17] and is demonstrated in the following two examples. In Fig. 7. the three texture classes of gravel, rock and wood were learned (top) and a new mosaic test image was presented for recognition and labeling (bottom left). Here the images were taken using a 35 mm camera at the Jet Propulsion Laboratory (JPL). Note that the training image patches are different from the testing images. The input test image is successfully segmented and labeled as can be seen in the result label map (bottom right). Note that black represents a class label in this figure. An example of an airborne image classification is presented in Fig. 8. In this example the classes learned are bush (output label dark gray), ground (output label gray) and a structured area, such as a field present or the man-made structures (white). Here, the training was done on 128* 128 image examples (1 example per class). The input image is 800* 800. In the result presented (right) we see that the three classes have been found and a rough segmentation into the three regions is achieved. Note in particular the detection of the bush areas and the three main structured areas in the image, including the man-made field, indicated in white. The above results demonstrate the network's capability for generalization and robustness to noise in complex real-world images.

VI. Summary and Discussion

We have presented a texture-analysis system in which learning of texture-discrimination rules is achieved in a multi-resolution environment. We have thus combined a learning paradigm with pyramidal feature-extraction techniques.

We have demonstrated the ability of the learning approach to contribute in a variety of applications. High-percentage classification rates are achieved for both structured and unstructured (stochastic) textures. The classification results presented in this work are competitive in performance with other techniques widely used in the literature. The main advantages of the learning approach are the ability to handle all types of textures within one framework, and to produce probability estimates for the output classes. A minimal feature set is learned and the classification rules are available for the user's information. The system can thus enhance the user's knowledge of the input domain via its own extracted rule knowledge base. Note that a segmentation of the image is achieved via the recognition process.

The output probability maps give more information about the decision process than do the hard-decision output common in other methods. We have demonstrated the generalization capability of the system, based on the probability maps, to the identification of an unknown class, so-called "pattern discovery." An application to natural scene analysis, with initial attempts at remote-sensing image analysis, are shown. These initial results are very encouraging and indicate the robustness of the system in coping with noisy real-world applications.

We have recently achieved classification accuracy rates of 97% on large complex databases of 30 textures [11], [18], demonstrating the scalability of the system performance. Scale and rotation-invariant recognition is the topic of current investigation. Encouraging rotation-invariant recognition results can be found in [18].

Acknowledgment

We are grateful to P. Smyth for useful discussions and suggestions regarding the ITRULE algorithm and the manuscript as a whole.

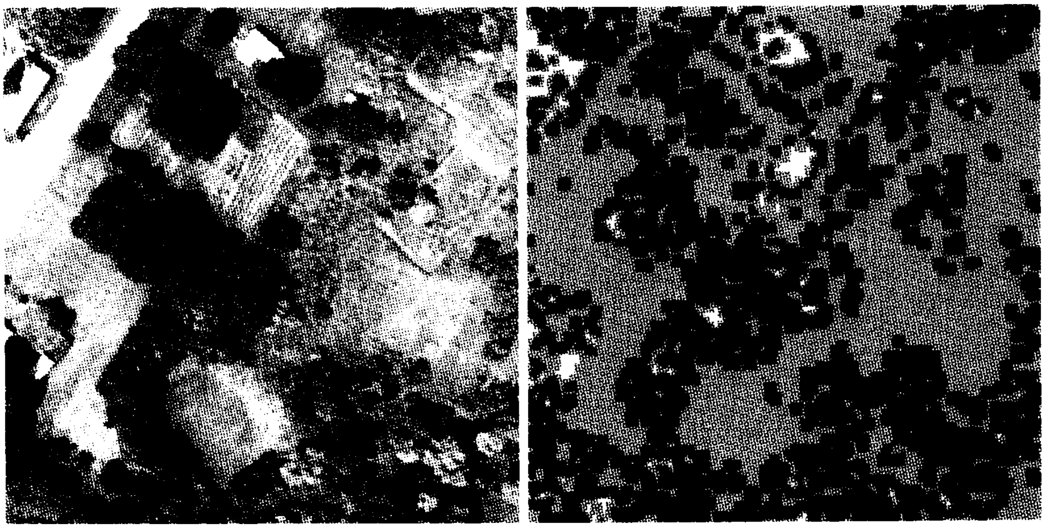

Fig. 8. Aerial image analysis. The input test image is shown (left) followed by the system output classification map (right). Dark gray indicates a bush area, light gray is a ground cover region and white indicates man-made structures.

References

[1] H. Greenspan, R. Goodman, and R. Chellappa, "Combined neural network and rule-based framework for probabilistic pattern recognition and discovery," in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson and R. P. Lippman, Eds. San Mateo, CA: Morgan Kaufmann, 1992, pp. 444–452.

[2] F. M. Vilnrotter, R. Nevatia, and K. E. Price, "Structural analysis of natural textures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 76–89, 1986.

[3] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 959–963, 1985.

[4] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *J. Optical Soc. Amer. A*, vol. 7, pp. 923–932, 1990.

[5] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55–73, 1990.

[6] M. Tuceryan and A. K. Jain, *The Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. World Scientific Publishing Co., 1992, ch. 11.

[7] J. G. Daugman, "Uncertainty relation for resolution, spatial frequency, and orientation optimized by 2D visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, pp. 1160–1169, 1985.

[8] D. H. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. Amer. A*, vol. 4, pp. 2379–2394, 1987.

[9] P. J. Burt and E. A. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, 1983.

[10] C. H. Anderson, "A filter-subtract-decimate hierarchical pyramid signal analyzing and synthesizing technique," U.S. Patent 4 718 104, 1987.

[11] H. Greenspan, "Multi-resolution image processing and learning for texture recognition and image enhancement," Ph.D. thesis, California Inst. of Technol., 1994.

[12] H. Greenspan, S. Belongie, P. Perona, R. Goodman, S. Rakshit, and C. H. Anderson, "Overcomplete steerable pyramid filters and rotation invariance," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, Seattle, WA, June 1994, pp. 222–228.

[13] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley, 1973.

[14] P. Smith and R. Goodman, "An information theoretic approach to rule induction from databases,' *IEEE Trans. Knowledge Data Eng.*, vol. 4, no. 4, pp. 301–316, 1992.

[15] R. M. Goodman, C. Higgins, J. Miller, and P. Smyth, "Rule-based networks for classification and probability estimation," *Neural Computation*, vol. 4, no. 6, pp. 781–804, 1993.

[16] P. Brodatz, *Textures*. New York: Dover, 1966.

[17] H. Greenspan and R. Goodman, "Remote sensing image analysis via a texture classification neural network," in *Advances in Neural Information Processing Systems 5*, C. L. Giles, S. J. Hanson and J. D. Cowan, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp. 425–432.

[18] H. Greenspan, S. Belongie, P. Perona, and R. Goodman, "Rotation invariant texture recognition using a steerable pryamid," in *Proc. 12th Int. Conf. on Pattern Recognit.*, Oct. 1994, to appear.

Geometric Primitive Extraction Using a Genetic Algorithm

Gerhard Roth and Martin D. Levine

***Abstract*—Extracting geometric primitives from geometric sensor data is an important problem in model-based vision. A minimal subset is the smallest number of points necessary to define a unique instance of a geometric primitive. A genetic algorithm based on a minimal subset representation is used to perform primitive extraction. It is shown that the genetic approach is an improvement over random search and is capable of extracting more complex primitives than the Hough transform.**

***Index Terms*— Genetic algorithms, geometric primitive extraction, Hough transform, template matching, minimal subset.**

Manuscript received August 1, 1992; revised December 9, 1993. This work was supported in part by NSERC Grant G1840. Recommended for acceptance by Associate Editor B. Bhanu.

G. Roth is with the Institute for Information Technology of the National Research Council of Canada, Ottawa, ON, K1A OR6 Canada; e-mail: roth@iit.nrc.ca.

M. D. Levine is with the Center for Intelligent Machines, McGill University, Montreal, PQ, H3A 2A7 Canada.

0162-8828/94$04.00 © 1994 IEEE