TEXTURE DISCRIMINATION USING REGION BASED PRIMITIVES

Ronald Lumia, Robert M. Haralick, Linda Shapiro

Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

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 regions. In each region, a set of properties is calculated. The regions along with their respective properties constitute the primitives.

The discrimination between categories has two parts: the training phase and the classification phase. Primitives and relationships obtained from representative training images are used to develop criteria for the classification phase. During classification, the primitives of the test image are grouped into clusters based on a minimum distance classifier. Then, each primitive is assigned to the most likely texture class given its cluster name and the cluster names of its spatially neighboring primitives.

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 while others have centered around a structural approach. (See Haralick [1] for a review of statistical and structural approaches to texture.)

The reason for texture's importance hinges around one of the most important processes in scene analysis, dividing an image into meaningful regions. One way to accomplish this segmentation is through edge detection [2]. An equally valid approach to region segmentation employs texture discrimination: different textures define the different areas.

This paper will describe a method used to discriminate between texture classes based on measurements made on typically small regions determined by an initial segmentation. Each region and its measurements constitute a primitive. A region is assigned to the most likely texture class 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.

CLASSIFICATION PHASE - OVERVIEW

The structure of the classification phase is shown in Fig. 1. Image segmentation programs based on the sloped facet model [3] are used to process the test image to obtain the symbolic image, an image composed of regions. Each region in the symbolic image is assigned a unit number, and a region adjacency graph (RAG), which contains the adjacent unit numbers for any given unit, is generated.

Using the test image, the symbolic image, and perhaps 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. Using the cluster decision rule parameters calculated in the training phase, each unit is assigned a cluster code (CL). The updated property list, the RAG, 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.

CLASSIFICATION PHASE

A test image is transformed into a symbolic image whose regions are called units. The classification problem is to assign each unit to the most likely texture category. A unit is first classified, based on measured properties, to belong to a cluster type which has been defined in the training phase. All of the units which are spatially adjacent to the unit under consideration are also assigned cluster names. From this information, the most likely texture category for the unit, given the cluster name (clust) of the unit and the cluster names of its neighbors (neigh), can be calculated. Using Bayes rule, the conditional probability density can be expressed as

$$P\{cat_i \mid clust_j, neigh\} = \frac{P\{neigh \mid clust_j cat_i\} P\{clust_j, cat_i\}}{P\{clust_j, neigh\}} \quad (1)$$

For a given unit, the probability in equation (1) is calculated. Then, the unit is classified as category i (cati) if

$$P\{cat_i \mid clust_j, neigh\} > P\{cat_k \mid clust_j, neigh\} \quad \forall i \neq k \quad (2)$$

The terms in equation (1) will be considered separately to obtain a more mathematically tractable expression.

We assume that the cluster name assignments for neighboring units are uncorrelated. This seems quite reasonable. If two adjacent units were to exhibit very similar characteristics and thus be correlated in terms of cluster names, then the initial segmentation was wrong; the two units should have been segmented as one unit in the symbolic image. With this assumption, the first term of equation (1) can be expressed in terms of the product of the probability density functions for each cluster type as

$$P\{neigh \mid clust_j cat_i\} = \binom{N}{n_1 n_2 \ldots n_c} \prod_{k=1}^{c} [P\{neighbor \in clust_k \mid clust_j, cat_i\}]^{n_k} \quad (3)$$

where neighbor denotes a neighbor spatially adjacent to the unit under consideration, n^k is the number of spatially adjacent neighbors of the unit under consideration whose cluster name is k, N is the total number of neighboring units, and c is the number of cluster types. The multinomial coefficient is calculated by

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

The next term of equation (1) can be expressed as

$$P\{clust_j, cat_i\} = P\{clust_j \mid cat_i\} P\{cat_i\} \quad (5)$$

The denominator of equation (1) will be identical for all categories. It adds no information to the classification process and can, therefore, be ignored in the discriminate function resulting in

$$P\{cat_i \mid clust_j, neigh\} = K P\{neigh \mid clust_j, cat_i\} P\{clust_j \mid cat_i\} P\{cat_i\} \quad (6)$$

where K is a constant. Thus, equations (2), (3), (4), and (6) are required for unit classification.

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

1) $P\{cat_i\}$

   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 $P\{cat_i\}$ in equation (6) is ignored and its value is absorbed into the constant K.

2) $P\{clust_j \mid cat_i\}$

   From some representative sample images, the number of occurrences of cluster j which occur for units known to be from category i can be counted. The normalized result is an estimate of this required conditional probability density function.

3) $P\{clust_j \mid clust_k, cat_i\}$

   This term is estimated from the training data by calculating the normalized occurrence matrix of cluster names for category i.

TRAINING PHASE

The training phase, shown in fig. 2, 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, a symbolic image, a RAG, and a property list are generated. The main difference in the training phase occurs when the user manually selects regions from the training image which are representative of each texture category.

The property list is updated so that category number i is written into the appropriate column for all units previously declared to belong to the training set for texture class i. Given all of the sample units 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 cluster names for each class is generated. Thus, the properties associated with each unit can be assigned to one of the prototype cluster names based on a criterion such as minimum Euclidean distance. The prototype cluster vectors along with the minimum Euclidean distance classifier constitute the cluster decision rule.

This process rule is now applied to each set of the training texture classes with the results stored in the property list. Using the property list and the RAG, a cooccurrence matrix of cluster code vs cluster code can be calculated for each texture class. The $(i,j)^{th}$ element of each matrix is the number of times that adjacent units in the training data were labeled cluster i and cluster j for that texture class. Each matrix is then normalized by row.

Thus, the cluster decision rule parameters, which indicate the prototype values for cluster types, and the category decision rule parameters, $P\{cat_i\}$, $P\{clust_j|clust_k,cat_i\}$ $P\{clust_j|cat_i\}$, have been generated. This is all of the information required for the classification phase.

CONCLUSION

This paper has described a method to assign regions to texture categories based on measured property values and spatial interrelationships of the texture primitives. The preliminary results appear quite encouraging.

The method appears to be sufficiently general for a wide class of images. Furthermore, the flexibility of the processing methods should facilitate evaluation of alternative procedures. For example, using the same training and test images, different methods for the image segmentation process can be compared and evaluated. Further investigation into the property measurements which provide the most information is required.

REFERENCES

[1] R. M. Haralick, "Statistical and Structural Approaches to Textures," Proceedings of the IEEE, Vol. 67, No. 5, May, 1979, pp. 786-804.

[2] L. S. Davis, "A Survey of Edge Detection Techniques," Computer Graphics and Image Processing, Vol. 4, Sept. 1975, pp. 248-270.

[3] R. M. Haralick, "Edge and Region Analysis for Digital Image Data," Computer Graphics and Image Processing, Vol. 12, 1980, pp 60-73.

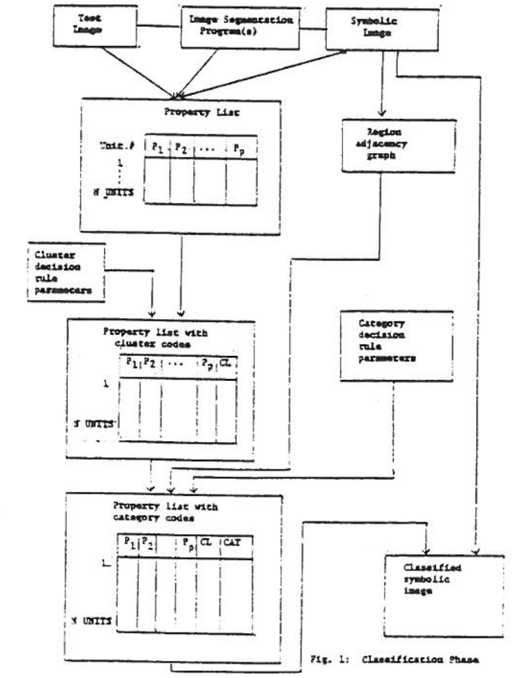

Fig. 1: Classification Phase

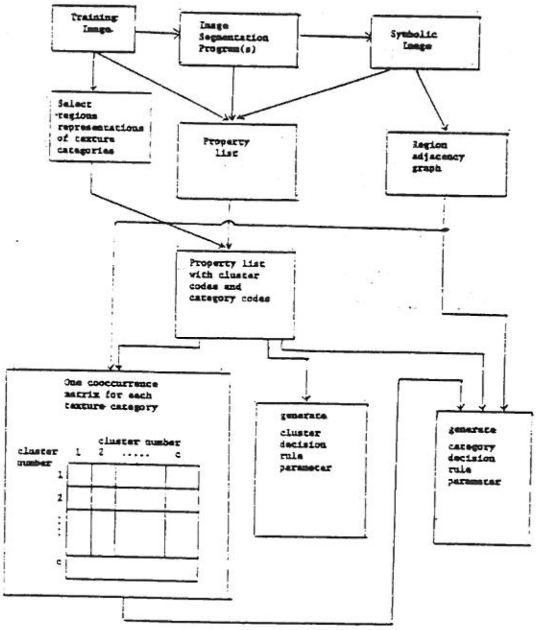

Fig. 2: Training Phase