Texture Analysis

Using Generalized Cooccurrence Matrices

L. Davis, S. Johns, and J. K. Aggarwal

TR - 95

April, 1979

L. Davis is with the Department of Computer Science, The University of Texas at Austin, Austin, Texas 78712.
S. Johns is with Abbott Laboratories, Dallas Texas.
J. K. Aggarwal is with the Department of Electrical Engineering, The University of Texas at Austin, Austin, Texas 78712.

The research reported herein was supported in part by grants AFOSR 77-3190 and NSF ENG 74-04986.

Abstract

We present a new approach to texture analysis based on the spatial distribution of local features in unsegmented textures. The textures are described using features derived from generalized cooccurrence matrices (GCM). A GCM is determined by a spatial constraint predicate, F, and a set of local features $P = \{(X_i, Y_i, d_i),\ i = 1,\ldots,m\}$ where (X_i, Y_i) is the location of the i^{th} feature, and d_i is a description of the i^{th} feature. The GCM of P under F, G_F, is defined by $G_F(i,j)$ = number of pairs p_k, p_ℓ such that $F(p_k, p_\ell)$ is true and d_i and d_j are the descriptions of p_k and p_ℓ, respectively. We discuss features derived from GCM's and present an experimental study using natural textures.

I. Introduction

Texture models play an important role in many image analysis systems. Textural features can be crucial for the segmentation of an image and can serve as the basis for classifying image parts. Experience with analyzing images containing textured regions (especially images of natural scenes) has often led to a qualitative distinction between two classes of textures -- micro-textures and macro-textures (see Hanson et al [1], among others).

Both micro-textures and macro-textures are ordinarily associated with an intuitive model of textures as being composed of "pieces" or "elements" whose shapes, sizes, and placement are the crucial factors in their analysis. Two common dimensions along which such textures are ordinarily described are dimensions of coarseness and directionality. Coarseness corresponds to the size of the texture elements, and directionality corresponds to both the orientation of the texture elements and to their spatial arrangement. It should be pointed out, of course, that there are other attributes of textures (e.g. homogeneity aspects) which can, and often should, be used in texture description. Also, this intuitive model is not adequate for the description of many textures -- for example, textures formed by diffusion processes cannot be naturally described as being composed of pieces. There are many other examples.

The distinction between micro-textures and macro-textures is based on the size of the underlying texture elements. For micro-textures, the texture elements are assumed to be small (e.g. diameter only several pixels), while for macro-textures, the texture elements are assumed to

be larger. Most micro-textures can be adequately described by a few features computed from a cooccurrence matrix [2] or a difference histogram [3]. However, as the texture elements become larger, the analysis of the texture using such techniques becomes more tenuous since statistics derived from cooccurrence matrices or difference histograms will depend more on the intensity transitions within texture elements than on the structural organization of the texture. Thus, we would expect that the texture descriptions based on features derived from cooccurrence matrices or difference histograms would tend to become less useful as the texture becomes more "macro".

Clearly, as we "zoom" in on a micro-texture, it becomes a macro-texture; and as we reduce the resolution with which we view a macro-texture, it becomes a micro-texture. Thus, the distinction between micro- and macro- textures is really a distinction between images of textures, and not between the physical textures underlying the images. However, models for the physical textures, while preferable to image models, are exceptionally difficult to obtain.

Only a few macro-texture analysis procedures have been described in the literature. Maleson and Feldman [4] suggest first segmenting the texture to obtain the texture elements, and then directly describing their shapes and their spatial distribution. Segmenting the textures, however, is itself a difficult problem. Furthermore, our ability to compute sophisticated shape descriptions is currently very limited, so that except for simple (e.g., convex) shapes, the shape description problem is not necessarily less complex than the texture description problem. So, although the approach described in [4] is very appealing,

since it attempts to directly capture our intuitive texture model, there are many practical situations where it might be difficult to apply.

An indirect approach towards the same end is to analyze the "primal sketch" (a rich edge map) of a texture region (see Marr [5]). Marr suggests applying certain similarity grouping operations to the primal sketch, while Zucker et al [6] suggests that, very often, histograms of aspects of the primal sketch (e.g. of edge magnitude, or orientation) can be used to discriminate between different textures. The approach sugggested in this paper lies between these two; it attempts to capture the spatial distribution of elements in a primal sketch, but without resorting to the computationally complex grouping operators suggested by Marr. This approach, based on "generalized cooccurrence matrices", is described in the following sections.

II. Generalized Cooccurrence Matrices

In order to analyze a texture, we will replace the texture by another image which indicates the positions of certain local properties in the original texture image -- e.g., of edges. Ordinarily, this second image is obtained by convolving a small set of masks with the texture image, and assigning local feature descriptions to the peaks, or local maxima, of the correlation surfaces. Each point corresponding to a local maxima has some description associated with it. For example, if the masks were edge masks, then the description might include the magnitude and orientation of the edge. The cooccurrence matrix, as ordinarily defined, will not serve as a useful tool for describing the distribution of these local maxima, because most local maxima will not have other local maxima at specific relative positions.

What we require is some generalization of a cooccurrence matrix which will capture some of the important spatial properties of the distribution of local maxima. We will define a generalized cooccurrence matrix as follows:

Let $P = \{p_i = (x_i, y_i, d_i)\}_{i=1}^{n}$ where (x_i, y_i) is the location and d_i the description of the i^{th} local maxima, p_i. We assume that the d_i have been quantized in some way. Let F be a spatial constraint predicate; e.g., $F(p_i, p_j, k) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < k$ is true iff the distance between p_i and p_j is less than k. Then the $(i,j)^{th}$ entry of the generalized cooccurrence matrix (GCM), G_F, is a count of the number of pairs (p_k, p_ℓ) with $F(p_k, p_\ell) = T$ and d_i and d_j the descriptions of p_k and p_ℓ, respectively.

We can identify several spatial constraint predicates which should often give rise to useful GCM's:

1) $F_1(p_i,p_j,k)$ is true if the distance between p_i and p_j is less than or equal to k.
2) $F_2(p_i,p_j)$ is true if p_j is the nearest neighbor of p_i.
3) $F_3(p_i,p_j,\theta,\Delta)$ is true if p_j lies in the shaded area displayed in Figure 1.

One can, of course, combine these predicates: $F_1(p_i,p_j,k) \wedge F_2(p_i,p_j)$ finds a nearest neighbor of p_i only if it is within distance k; $F_2(p_i,p_j) \wedge F_2(p_j,p_i)$ counts only mutual nearest neighbors.

We can illustrate the computation of GCM's with the aid of Figure 2. In Figure 2, we have coded the orientations of the local maxima of some gradient operation as follows: H - horizontal, V - vertical, L - left diagonal (\), R - right diagonal (/), blank - no local maxima at that point. Figures 3 and 4 show two GCM's for the texture of Figure 2. In Figure 3, we have used $F = F_1(p_i,p_j,2)$, while in Figure 4, we have used $F = F_3(p_i,p_j,\theta(p_i),0) \wedge \forall\ p_j'[F_3(p_i,p_{j'},\theta(p_i),0) \rightarrow d(p_i,p_{j'}) \geq d(p_i,p_j)]$ -- i.e., we look in a direction orthogonal to the orientation of p_i for the closest other local maxima.

Consider now the $(R,H)^{th}$ position of the GCM of Figure 3. It represents the sum of the number of horizontal points within distance 2 of right-diagonal points. We can see how this number was arrived at by considering the R's in the order that they would be encountered in a left-right, top-down scan of the image of Figure 2. The first R has no neighbors within (Manhattan) distance 2, the second has 1, the third 1,

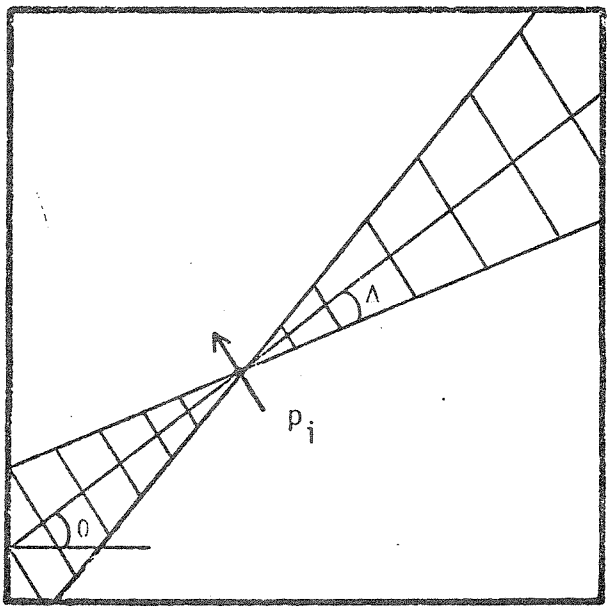


Figure 1. $F_3(p_i, p_j, \theta, \Delta)$

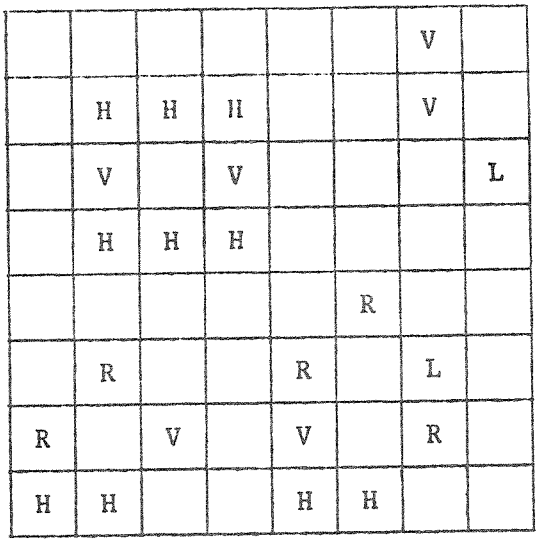


Figure 2

	H	V	L	R
H	22	11	0	5
V	11	6	1	4
L	0	1	0	3
R	5	4	3	4

Figure 3. GCM using $F = F_1(p_i, p_j, 2)$.

	H	V	L	R
H	2	5	0	3
V	1	4	0	2
L	1	0	0	1
R	3	1	1	0

Figure 4. Directional, nearest neighbor GCM.

the fourth 2 and the fifth 1 for a total of 5. Since $F_1(p_i,p_j,k)$ if and only if $F_1(p_j,p_i,k)$, the GCM of Figure 3 is symmetric.

Now consider the $(R,H)^{th}$ position of the GCM in Figure 4. To find the nearest neighbor of any R point along a line orthogonal to R, we simply start a symmetric, bi-directional scan along that line starting from R. If two nearest neighbors are found, both contribute to the GCM. So, we see that the third, fourth and fifth R of Figure 4 have H's which satisfy the predicate used to construct the GCM of Figure 4, and so the $(R,H)^{th}$ position of the GCM has value 3. Note that since $F_3(p_i,p_j,\theta(p_i),0)$ does not by itself necessarily imply $F_3(p_j,p_i,\theta(p_j),0)$, the GCM of Figure 4 is not symmetrical.

III. Operations on GCM's

In this section we will discuss a variety of operations which can be performed on GCM's for the purpose of computing texture descriptors. Haralick et al [2] describe many statistics which can be computed from grey level cooccurrence matrices. Some of the operations which we will describe are similar to the ones presented there. It should be pointed out however that the interpretation of the results of the operations depends on a) the local operator used to generate P (i.e., an edge detector, spot detector, etc.), and b) the particular F used to compute the GCM. In what follows, therefore, we will identify several different operations which can be applied to GCM's and discuss the interpretations of the results of these operations for various local operators and spatial constraint predicates. We will use G_F to denote the GCM produced using the spatial constraint predicate F.

a) Generalized contrast - $C(G)$

$$C(G_F) = \sum_{d_i,d_j} D(d_i,d_j)\, G_F(d_i,d_j)$$

where $D(d_i,d_j)$ is some measure of the difference between the i^{th} and j^{th} descriptors. For conventional cooccurrence matrices, where d_i corresponds to intensity i, $D(d_i,d_j) = |i - j|$. C reflects the homogeneity of the spatial distribution of local features with respect to F.

Examples

1) local operator: edge detector; $F = F_1(p_i,p_j,k)$ -- all neighbors of p_j within distance k of p_i. The d_i are just the orientations

of the local maxima, $-\pi/2 \leq d_i \leq \pi/2$; $D(d_i,d_j) = \sin(d_i - d_j)$.
For such GCM's, C will be high when most p_i are "surrounded" by p_j with different orientations; C will be lowest if all pairs p_i,p_j within distance k of one another have the same orientation. Therefore C will be a measure of anisotropy - the lower C, the more anisotropic the texture.

2) local operator: Laplacians of various sizes - best sizes are assigned to points by a process similar to the one described for edges in [3]; $F = F_1(p_i,p_j,k)$; the d_i are the sizes of the Laplacian; $D(d_i,d_j) = |i - j|$. The purpose of using Laplacians of several sizes is to collect information on the size of the texture elements. High values of C indicate a spatial mixture of elements of various sizes. C is lowest when all pairs p_i,p_j within distance k of one another have the same size. Thus, in this case, C measure the spatial homogeneity of the texture with respect to the size of the texture elements.

3) same as 1-2, but with $F = F_2(p_i,p_j)$-- p_j being the nearest neighbor of p_i. C can be interpreted in much the same way in both cases. One would use F_2 in the absence of any good a priori information on the size and density of the texture elements.

4) same as 1-2, but with $F = F_3(p_i,p_j,\theta,\Delta) \wedge F_1(p_i,p_j,k)$.
A texture may have a strong directional homogeneity only in a specific direction: Figure 5 displays a spatial distribution of directions which exhibits strong anisotropy through C only when $F = F_3(p_i,p_j,45°,0) \wedge F_1(p_i,p_j,k)$ is used. Using $F = F_1(p_i,p_j,k)$ would result in a C indicating relatively high isotropy.

R	V	H	L	R	V
V	H	L	R	V	H
H	L	R	V	H	L
L	R	V	H	L	R
R	V	H	L	R	V
V	H	L	R	V	H

Figure 5. Strong anisotropy in 45° direction.

V	H	L	R	V	H
V	H	L	R	V	H
V	H	L	R	V	H
V	H	L	R	V	H
V	H	L	R	V	H
V	H	L	R	V	H

Figure 6. Texture with misleadingly high U.

b) Uniformity:

$$U = \sum_{d_i,d_j} G_F(d_i,d_j)^2$$

U measures the uniformity of the entries in G_F. It is lowest when all entries are equal.

Examples

1) local operator: edge detector; $F = F_1(p_i,p_j,k)$; $d_i = O_i$. Low values of U indicate that a) all directions occur with nearly equal frequency, and b) the spatial distribution of directions is relatively random. U is more difficult to interpret than C for this GCM, inasmuch as the highly directional texture resulting in the local maxima with orientations as shown in Figure 6 has relatively low U.

2) same as (1), but with a directionally sensitive F: allows us to distinguish between textures giving rise to gradient orientations such as Figure 6 and truly "uniform" textures.

c) Ratios of features derived from different GCM's.

Ratios of features derived from different GCM's can also yield useful texture descriptors. Ordinarily the GCM's will be based on similar or identical local operators; the differences will arise from changing F.

Examples

1) local operator: edge detector; feature: generalized contrast; $F = F_1(p_i,p_j,k)$, $F' = F_1(p_i,p_j,k+\Delta k)$. In this case we compute the ratio $C_{FF'} = C_{F'}/C_F$, which reflects the extent of the anisotropy of the underlying texture. We noted before that a high value of C will ordinarily

correspond to an anisotropic texture. A high value of $C_{FF'}$ indicates that as we increase the neighborhood size of F_1, the texture appears more anisotropic; a value of $C_{FF'}$ near 1 indicates that the anisotropy of the texture remains constant as we change k. Note that a plot of C_F versus k would indicate the point at which the texture become anisotropic.

2) same as above, but with $F = F_1(p_i,p_j,k) \wedge F_3(p_i,p_j,\theta,\Delta)$ and $F' = F_1(p_i,p_j,k) \wedge F_3(p_i,p_j,\theta',\Delta)$. Here, $C_{FF'}$ indicates whether the texture shows more directional consistency in direction θ than θ'. Note that we could fix $\theta = \theta'$, but let Δ change between F and F'. In this case we get a measure of the robustness of the directional consistency in the general direction θ. Note again, we can plot $C_{FF'}$ as a function of θ, of Δ, or of θ and Δ and discover useful information about the directionality of the underlying texture.

d) Detecting clusters in GCM's. In [6], Zucker et al examined the histograms of local maxima for peaks in order to segment an image composed of several textures into the individual texture regions. In much the same way, we could search GCM's for peaks in order to segment an image. Segmentations based on peaks or clusters discovered in GCM's should be more reliable than segmentations derived from histogram analysis.

e) Texture edge detection. Suppose that we have a region which is composed of two textures. How can this be discovered? One possible approach is to compare the GCM's of different parts of the region. There are at least two levels at which this can be done:

1) The GCM's can be compared directly through some metric defined on the matrices. In fact, Chen and Pavlidis [7] have very recently suggested this approach for grey-level cooccurrence matrices to segment images containing textured regions using a pyramid data structure.

2) Features derived from the GCM's can be compared. We might, for example, look at the difference in contrast, or compare contrast "variograms."

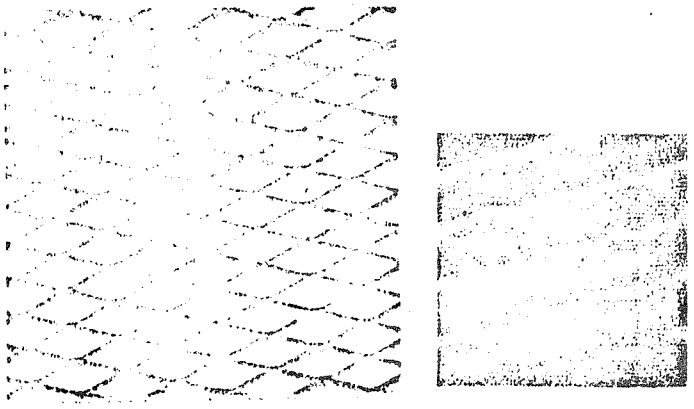
In the next section we report the results of a classification experiment which uses both features computed from GCM's and features computed from grey level cooccurrence matrices (GLCM's).

IV. CLASSIFICATION EXPERIMENT

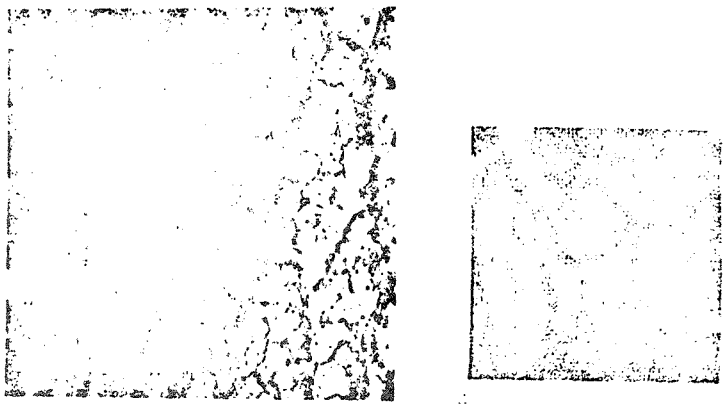
This section describes a classification experiment comparing the performance of various features computed from GCM's to the performance of conventional grey level cooccurrence matrices. Two different grey level cooccurrence matrices and four different GCM's were computed. From these, a set of four features was computed from each of the six matrices. These features were used singly and in pairs as features for a leave-one-out classifier. The results of the classifier are used to compare the performance of GCM's and grey level cooccurrence matrices.

The data set used in this study consists of five different classes of textures: pebbles, tree bark, orchard, iron grating, and scrap iron. Six samples of each class were used. The original photographs are displayed in Figure 7.

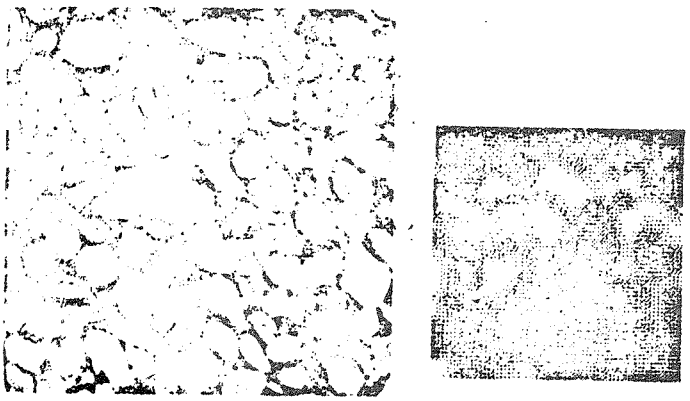
Whenever different samples of the same texture type are taken from different frames, a normalizing process should be performed to eliminate the effect of any difference in conditions (e.g. lighting, film, or developing) that exists between frames. In this study, all six samples of each texture were taken from the same negative. However, in order to prevent the different texture types from being distinguished strictly on the basis of first-order statistics such as mean grey level, each sample was transformed so as to have a uniform grey level histogram. This process has the additional effect of increasing the contrast of the image as well as enhancing the edges. Examples of the resulting images, after grey scale normalization, are also shown in Figure 7. The normalization procedure is described in Rosenfeld and Kak [3].



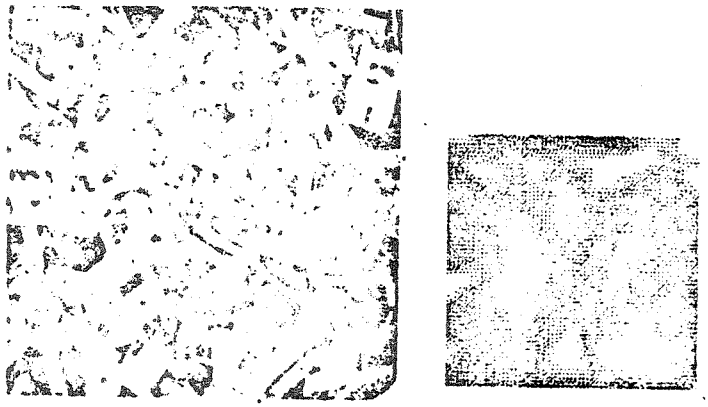
a) grating



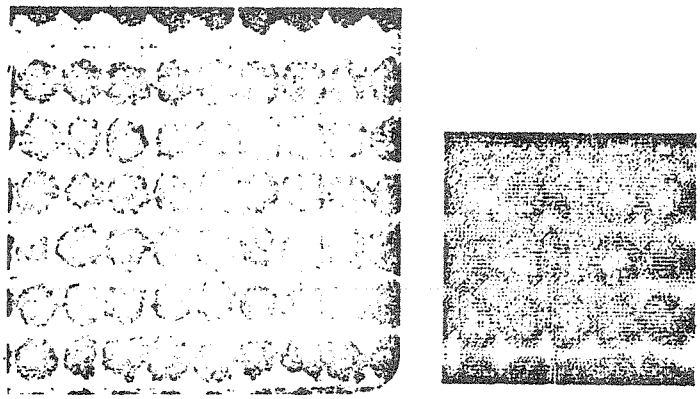
b) tree bark



c) pebbles



d) scrap metal



e) orchard

Figure 7. Original images and examples of normalized, digitized samples.

On each of the 30 texture samples, six different cooccurrence matrices were computed. Two of these were grey level cooccurrence matrices (GLCM's) and the remaining four were GCM's.

The first GLCM (GLCM1) was based on a point and its immediate neighborhood. The second GLCM (GLCM2) was based on the four points labeled i, j, k, and ℓ in Figure 8.

Four different GCM's were computed. The descriptor used was edge orientation, quantized to four orientations. Two different edge detectors were used and for each of these two different spatial constraint predicates were used. Both of the edge detectors calculate gradients for the four primary orientations: 0, 45, 90, and 135. The orientation with the largest gradient magnitude is chosen as the most likely edge orientation at that point. Edge orientations are computed at every point in the image except for points near the edge of the image that do not have complete neighborhoods.

The first edge detector was the Kirsch operator [8]. The second edge detector computes gradients in a similar way, only on a 5x5 neighborhood. This edge detector is less sensitive to noise. Here, the neighborhood of a point is split into slightly more complex regions as shown in Figure 9.

The edge detectors just discussed compute an edge orientation at every point in the image. For macro-textures, it is highly unlikely that there are true edges at all of these points. If texture elements are several pixels across, then those pixels interior to the elements should not be labeled as edge points. In order to eliminate most of the

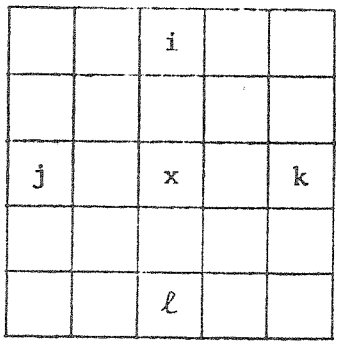


Figure 8. Points used for computing GLCM2.

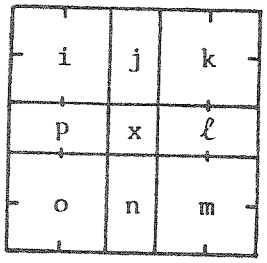


Figure 9. Neighbors for the 5×5 edge detector.

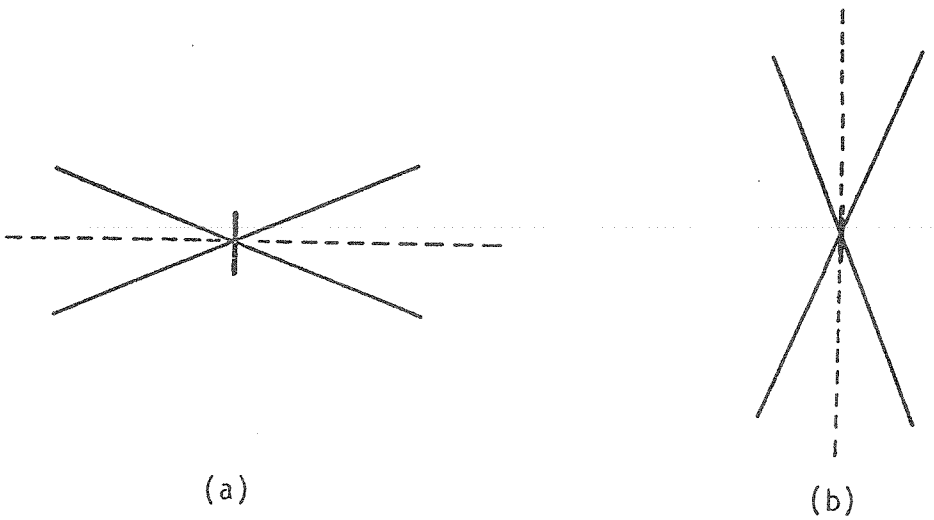


Figure 10. Predicates used for GCM's.

edge responses of interior points, non-maxima suppression was performed over the results of the edge detector (see Rosenfeld and Kak [3]).

In this study, non-maxima suppression was performed as follows. Associated with each point in the image is some edge orientation E and gradient magnitude M. Passing through each point are two line segments orthogonal to E at that point. All edges on these line segments whose magnitudes are less than M are suppressed. The length of the line segments is a parameter of the non-maxima suppression. For this experiment all non-maxima suppression was done with line segments three pixels in length.

The GCM's in this experiment are 4x4 matrices since edge orientations were only computed for the four principle directions. For each of the two edge detectors two different spatial constraint predicates were used. For both predicates there are two cone-shaped regions projecting out from each local maxima to a distance of five pixels. The difference between the two spatial constraint predicates is the orientation of the cone-shaped regions. For one of the predicates the cones are oriented orthogonal to the edge (Figure 10a) and for the other, the cones are oriented in the direction of the edge (Figure 10b).

For each maxima in one of the cones some entry in the GCM is incremented. If the edge at the apex of the cones is a vertical edge, for example, and a local maxima with a horizontal edge orientation is located within one of the cone-shaped regions, then the (vertical, horizontal)'th entry in the GCM is incremented.

The final phase of the study consists of a classification experiment. This study uses a leave-one-out classifier. In this method, every sample

in the data set except one is used as the training set (it is assumed that the true class of each sample is known). The remaining sample is then classified using the statistics derived from the training set. Next, the test sample is exchanged with one of the samples in the training set. The appropriate class statistics are updated to reflect the changes in the training set. The entire process is repeated until each sample in the data set has been treated as an unknown and classified. This method of classification has the advantage of achieving maximum significance in both class statistics and classification performance results.

Ideally, the samples from each class form distinct clusters in the feature space. An unknown sample is classified as belonging to the class corresponding to the nearest cluster since that class has features most similar to those of the unknown sample. Most often Euclidean distance is used as a measure of the "nearness" of a cluster, although other measures of similarity are also possible. When Euclidean distance is used, some kind of normalization of the data must be performed to prevent features with large numerical values from dominating the distance calculations.

The normalization technique used in this study is to subtract the mean and divide by the standard deviation. More formally, let $\bar{f}$ be the feature values for some unknown sample and let $\bar{\mu}_c$ and $\bar{\sigma}_c$ be the mean and standard deviation for class c. The distance from the feature point of the unknown sample to the class c mean is denoted d_c and defined as:

$$d_c = \sqrt{\frac{(\bar{f}-\bar{\mu}_c)^2}{\bar{\sigma}_c}}$$

This normalization procedure takes into account the spread of values within a cluster. For example, in the one-dimensional case, if $(f-\mu_i) = (f-\mu_j)$ but $\sigma_i > \sigma_j$, the sample would be classified as belonging to class i since i has more variability. Similar reasoning applies to higher-dimensional cases.

For each of the five classes, the distances from the unknown sample to the class mean is computed. The class for which the distance is minimum is the class to which the unknown sample is assigned. The results of this classification are recorded in a confusion matrix.

The six cooccurrence matrices computed in this experiment consist of two GLCM's and four GCM's:

GLCM1 - 8 neighbors

GLCM2 - 4 neighbors of Figure 8

GCM1 - 3x3 edge detector; Spatial Constraint Predicate (SCP) of Figure 10a

GCM2 - 3x3 edge detector; SCP of Figure 10b

GCM3 - 5x5 edge detector; SCP of Figure 10a

GCM4 - 5x5 edge detector; SCP of Figure 10b

Four features were computed on each of the six cooccurrence matrices. Angular second moment, entropy, and correlation (see Haralick [2]) are the same for both GLCM's and GCM's. The definition of contrast, however, includes a difference term that represents the difference between two descriptors. For the GLCM's, the descriptors are grey level intensities and the difference between two grey levels i and j is taken to be $|i-j|$. For the GCM's, the descriptors are edge orientations and the difference

between two edges is the sine of the angle between the two orientations.

The main results of the classification experiment are condensed into Figures 11-16. For each of the four single features (the entries on the diagonal) and each of the six pairs of features (upper triangle) there is a table entry denoting the percentage of samples correctly classified using those features. From these tables we can see that GCM's performed significantly better than GLCM's (an average of 60% vs. 44% for single features and 68% vs. 43% for pairs of features).

In the scanner plots that follow, the following notation is used for the five classes of textures:

G = iron grating

O = orchard

B = tree bark

P = pebbles

M = scrap metal

The confusion matrices corresponding to the three best pairs of features for both GLCM's and GCM's are presented in Figures 17-22. Figures 23-24 contain sample scatter plots, one for GLCM1 and one for GCM4. The differences in clustering of the classes are readily seen in the scatter plots. The scatter plot for the GLCM, Figure 23, shows that the samples are highly overlapped in both dimensions, and only the grating samples form any kind of definite cluster. However, Figure 24 indicates very good clustering for three of the classes. Pebbles and scrap metal are confused in both GCM's and GLCM's, but they are very similar even to the human eye.

Of the four features used, contrast was the most useful in

	CON	ASM	ENT	COR
CON	70	73	73	80
ASM		73	70	60
ENT			63	60
COR				43

Fig. 11. Classification Results for GCM1

	CON	ASM	ENT	COR
CON	63	70	73	67
ASM		63	70	63
ENT			73	60
COR				37

Fig. 12. Classification Results for GCM2

	CON	ASM	ENT	COR
CON	57	53	60	63
ASM		60	63	63
ENT			67	70
COR				40

Fig. 13. Classification Results for GCM3

	CON	ASM	ENT	COR
CON	73	83	87	60
ASM		63	63	70
ENT			73	70
COR				43

Fig. 14. Classification Results for GCM4

	CON	ASM	ENT	COR
CON	47	50	50	47
ASM		57	50	43
ENT			37	37
COR				40

Fig. 15. Classification Results for GLCM1

	CON	ASM	ENT	COR
CON	43	43	43	43
ASM		57	33	40
ENT			33	40
COR				40

Fig. 16. Classification Results for GLCM2

CORRECT \ CLASSIFIED AS	G	O	B	P	M
G	3	3	0	0	0
O	0	4	2	0	0
B	0	2	3	0	1
P	0	0	2	1	3
M	0	0	1	1	4

Fig. 17. Confusion Matrix for ASM vs. ENT (GLCM1)

CORRECT \ CLASSIFIED AS	G	O	B	P	M
G	5	1	0	0	0
O	1	2	3	0	0
B	0	2	3	0	1
P	0	0	3	1	2
M	0	0	1	1	4

Fig. 18. Confusion Matrix for CON vs. ASM (GLCM1)

CORRECT \ CLASSIFIED AS	G	O	B	P	M
G	5	1	0	0	0
O	0	3	3	0	0
B	0	2	2	0	2
P	0	0	2	1	3
M	0	0	1	1	4

Fig. 19. Confusion Matrix for CON vs. ENT (GLCM1)

CORRECT \ CLASSIFIED AS	G	O	B	P	M
G	5	0	1	0	0
O	0	6	0	0	0
B	0	0	5	1	0
P	0	0	0	5	1
M	0	0	1	2	3

Fig. 20. Confusion Matrix for CON vs. COR (GCM1)

CORRECT \ CLASSIFIED AS	G	O	B	P	M
G	5	0	1	0	0
O	0	6	0	0	0
B	0	0	6	0	0
P	0	0	1	3	2
M	0	0	1	0	5

Fig. 21. Confusion Matrix for CON vs. ASM (GCM4)

CORRECT \ CLASSIFIED AS	G	O	B	P	M
G	6	0	0	0	0
O	0	6	0	0	0
B	0	0	6	0	0
P	0	0	1	3	2
M	0	0	1	0	5

Fig. 22. Confusion Matrix for CON vs. ENT (GCM4)

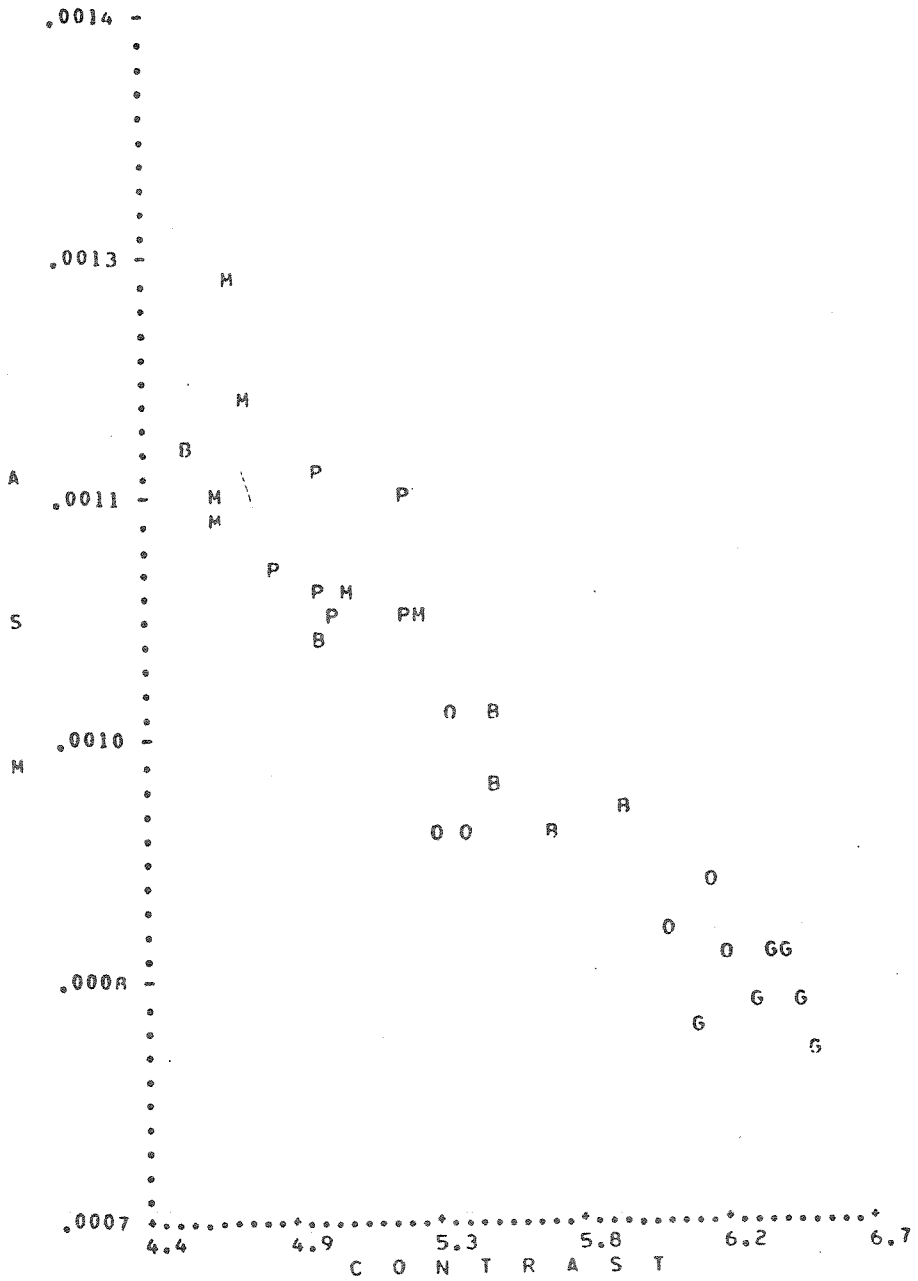


Figure 23. GLCM1: CON vs. ASM

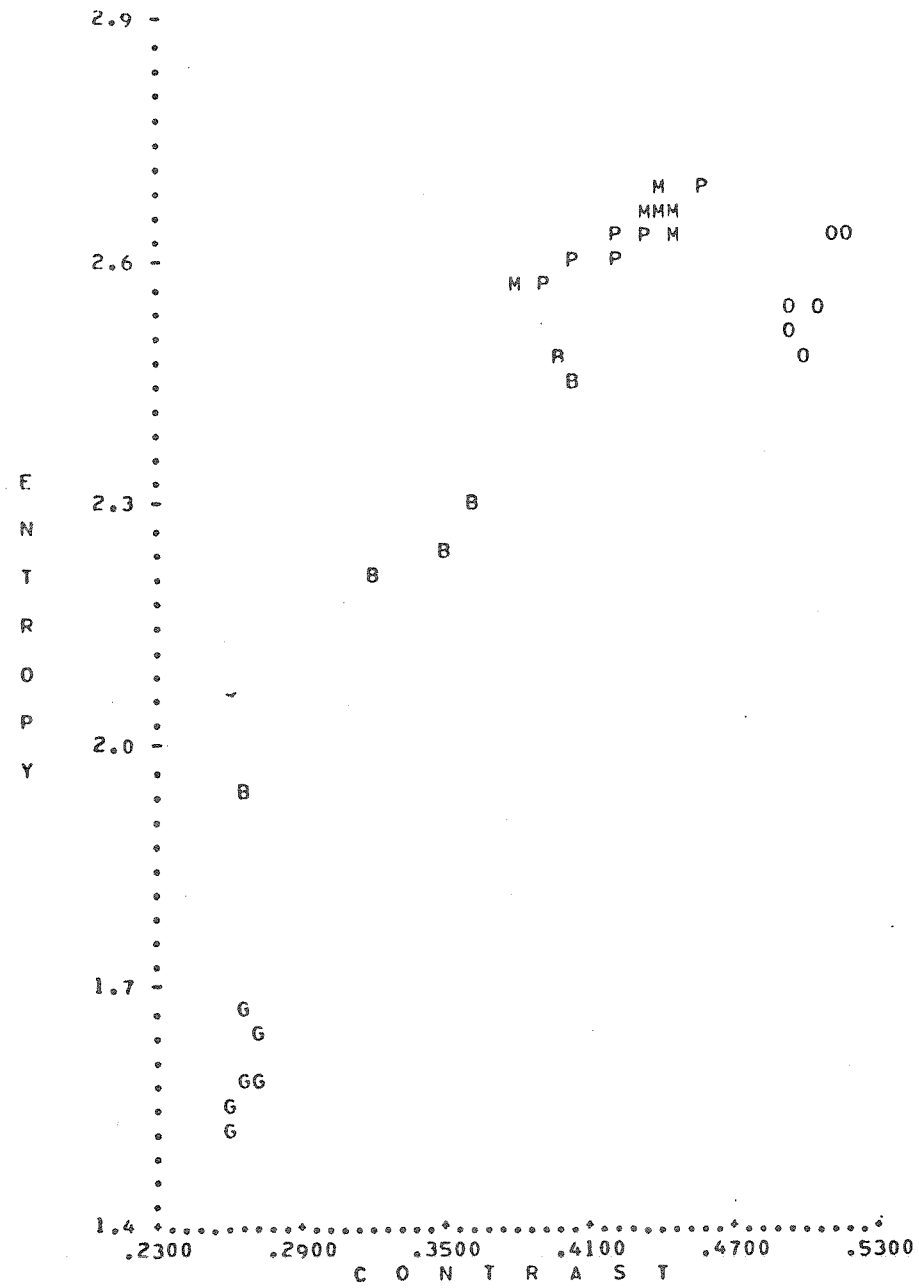


Figure 24. GCM4: CON vs. ENT

discriminating between textures. Of the six best pairs of features over GCM's and GLCM's, contrast was one of the features in five of those pairs. When the local descriptions are grey levels as in the GLCM's, contrast reflects both the coarseness of the texture and the contrast of the edges. In this experiment's data set, edge contrast was very similar between classes, so contrast is primarily a measure of texture coarseness. This explains why orchard samples were confused with grating samples and similarly pebbles with scrap metal. Orchard and grating are similar in element size as are pebbles and scrap metal.

The interpretation of contrast computed on GCM's is not as straightforward. Many texture characteristics such as shape, size, and spatial arrangement of the texture elements (as well as the SCP) affect contrast values. For example, as we have seen, contrast is low for elongated textures such as tree bark using the SCP of Figure 10b. We have seen that the samples of grating and orchard are similar in terms of grey level cooccurrences and element size. They are largely distinguishable on the basis of the shape of the elements (diamond versus circular) and their arrangement (interlocked versus rows of elements). The contrast measure of the GCM's seems to measure these characteristics quite satisfactorily, since the orchard and grating samples are more tightly clustered within their classes and much more widely separated between their classes than for the GLCM's (see Figure 24).

Another important observation to note is that bark samples are better classified using GCM4 than for GCM3. The difference is in the spatial constraint predicate. The most distinguishing characteristics of the bark samples are the elongation and directionality of the texture

elements. The spatial constraint predicate of GCM4 (looking in the direction of the edge) tends to measure elongatedness better than GCM3 since an important characteristic of textures containing elongated elements is the way edges are found in linear configurations.

V. SUMMARY

The results of the experiment described in this paper have demonstrated that GCM's are useful in discriminating between macro-textures that are not satisfactorily distinguishable using features derived from conventional cooccurrence matrices. Classification of thirty texture samples from five different classes yielded 80+% accuracy for feature pairs derived from GCM's versus 50-57% accuracy for those derived from GLCM's. Although the results are clearly promising, much work remains to be done. A larger data set, both in the number of classes and in the number of samples/class, would improve the significance of the results.

This study dealt with only one local description and two spatial constraint predicates. The power of GCM's lies in carefully matching the local operator and spatial constraint predicates to be able to measure particular characteristics of textures. Experimentation needs to be done with other local operators, such as spot and line detectors, and other spatial constraint predicates. Furthermore, some of the other ideas discussed in Section III (ratios, clustering) should be investigated.

A final suggestion concerns the study of "continuous" GCM's. These are defined in the same way as the discrete GCM's used in this study, with the following exception. A local operator does not deterministically return a single descriptor, but returns instead a vector of "probabilities" indicating the likelihoods of each of the possible descriptions. The appropriate entries of the GCM are not simply incremented by one. Instead, the product of the corresponding descriptor probabilities is added to the current entry value.

References

1. Hanson, A., E. Riseman, and P. Nagin, "Region growing in textured outdoor scenes," COINS TR-75C-3, 1975.

2. Haralick, R., K. Shanmugam and I. Dinstein, "Texture features for image classification," IEEET-SMC,3, 1973, 610-622.

3. Rosenfeld, A., and A. Kak, Digital Picture Processing, Academic Press, New York, 1976.

4. Maleson, J., C. Brown, and J. Feldman, "Understanding Natural Texture," Proc. DARPA Image Understanding Workshop, Palo Alto, CA., 1977, 19-27.

5. Marr, D., "Early processing of visual information," Phil. Trans. Roy Soc. B, in press.

6. Zucker, S., A. Rosenfeld, and L. Davis, "Picture segmentation by texture discrimination," IEEET-C,24, 1975, 1228-1233.

7. Chen, P., and T. Pavlidis, "Segmentation by texture using a split-and-merge algorithm," IEEE Workshop on Pattern Recognition and Artificial Intelligence, Princeton, NJ, 1978.

8. Kirsch, R., "Computer determination of the constituent structure of biological images," Computers and Biomedical Research, 4, 1971.