International Journal of Computer and Information Sciences, Vol. 4, No. 1, 1975

A Decision Theory Method for the Analysis of Texture

Bruce H. McCormick[1] and Sadali N. Jayaramamurthy[1]

Received June 1974

A decision theory method (DTM) is presented for the analysis of texture. It is based on the principles of statistical decision theory. This method, combined with other procedures described, is versatile enough to deal with a wide range of problems involving either statistical or structural textures. Its ability to perform scene segmentation using textural information, and edge/border detection, and to provide discrimination/recognition of both spatially and chromatically textured scenes is demonstrated with examples. Generation of 2D filters that act as textural feature detectors is illustrated with the help of interval covering theory.

KEY WORDS: Statistical decision theory; event; likelihood ratio; operating characteristic curve; local categorizer; acceptance set; interval covering theory; interval complexes; texture analysis; textural feature detectors.

1. INTRODUCTION

There seems little doubt that texture plays an important role in visual perception. It has been convincingly argued(33) that textures do carry information of use in object detection and recognition. Many laboratory studies(7,8,13,33,41) have also shown that various types of textural information have measurable effects on the perception of the depth, slant, and shape of surface. In spite of its importance, the textural information found abundantly in almost every visual scene in the real world was not profitably used in earlier research projects involving computer vision. Instead the work was carried out in a highly contrived experimental environment, that is, working with line drawings with the assumption of smooth surfaces, sharp edges, and so on.

[1] Department of Information Engineering, University of Illinois at Chicago Circle, Chicago, Illinois.

This restriction upon the class of scenes admissible for machine analysis can be mainly attributed to the lack of a general method suitable for the interpretation of textural information and yet permitting ready machine implementation. In this context the following problems have recently received wide attention: scene segmentation using textural information,[2,29] texture discrimination/recognition,[2,29,37] texture description,[45] texture analysis,[22,33,35,39] and texture synthesis.[6,38]

In this paper we present a decision theory method for texture analysis. It will be shown that this method is sufficiently versatile and is adequately adaptive to deal with a wide range of texture problems.

2. REVIEW OF EARLIER WORK

2.1. Visual Texture

Everyone seems to understand what "texture" means, since we live in a world rich in textures. If viewed at an appropriate angle, texture can be seen in almost every scene. Yet it is one of those terms which has escaped a precise scientific definition. As a matter of fact, the host of visual scenes indicated by the term texture is so enormously large and varied in nature that it is a very difficult task to span the varied concepts of texture by a single definition. A definition like "texture is that property of material which indicates what it feels like if touched," and adjectives like "rough," "smooth," etc., deal with tactile textures. We are here exclusively interested in visual textures.

Pickett[32] observes that for any visual scene to be seen as texture, there should be "a large number of elements (spatial variations in intensity and/or wavelength)" and that the "elements and rules of spacing or arrangements may be arbitrarily manipulated provided a characteristic repetitiveness remains." He adds "provided there is sufficient detail shown in a small enough visual angle, a characteristic texture emerges even when the basic elements or spacings are randomly distributed." Thus the primary attributes of visual texture are "many variations" and "repetitive variations."

Similar aspects are found in other available definitions. But even after a suitable choice of a working definition, there remain many difficulties inherent in the task of texture extraction technology. Hawkins[16] enumerates these in great detail and comes to the conclusion that "texture classification may very well be one of the more difficult tasks in the field of image processing."

Visual texture is really very sensitive to external conditions, such as lighting and angle of view. The same scene may present a very different texture even for slight modification in the external conditions. Pictures in Brodatz's[5] book can be used to endorse the above statement. To simplify the already complicated problem, we assume that the scenes with which we are concerned are taken under similar conditions.

With these problems and having applications in such important fields as biomedical sciences and remote sensing of the environment, texture analysis is both an attractive and challenging area.

2.2. Classification of Textures

Textures can be subdivided into at least two categories: statistical textures and structural textures.[23] *Statistical textures* in a visual scene can be regarded as defined by a set of statistics extracted from a large ensemble of local measurements made on the scene. We need more information than this to define a *structural texture*; here the texture is considered to be defined by subpatterns which occur repeatedly within the overall pattern according to well-defined placement rules, as, for example, in a wallpaper design.

Though we find many textures which can be classified adequately under one of these limiting categories, there are still many textures for which a strict classification may be questionable.

2.2.1. Structural Textures

As its name suggests, the subpattern/placement model[38] appears to be the most appropriate model to deal with structural textures. This model is by far the one most widely used in the literature. The "subpattern" is sometimes referred to as the "unit cell." Usually, but not necessarily, the subpattern itself contains subsub patterns and so on.

If we accept this model, the description of texture is indeed simplified to developing a language to describe the unit cell(s) and to spelling out the rules of its placement over a given region. Similarly, synthetic generation can be employed to develop the unit cell(s) and distribute it over the given area according to (appropriate curvilinear) placement rules. The work of Trout,[45] Conroy,[6] and Rosenfeld[38] reflects a similar treatment of these problems. The success of such methods depends entirely on the choice of the vital parameters of the model, such as primitives, signs, unit cells, and placement rules. In most of these methods such choice is made on a trial and error basis. It is highly desirable, though, that these parameters be determined from the analysis of "parent" texture, i.e., the scene that is being described or reproduced. This is by no means an easy task, especially when dealing with natural textures.

2.2.2. Statistical Textures

Many investigators in this field have developed methods which are usually statistical in nature when working with natural textures. Rosenfeld *et al.* have attempted to discriminate textures by merely detecting the differences in averages of local properties. They demonstrate their scheme by detecting textural edges using the gray level as the local property.[37]

Muerle[29] also uses statistical analysis for the discrimination of textures. He divides the whole scene of analysis into many cells. Starting with a single cell, he expands the region by comparing the statistical distributions of the cell with those of its neighbors and adding the new cells to fragments that have similar distributions.

Bajcsy[2] also proceeds in a similar direction, that is, by dividing the region of analysis into "windows," extracting "texture descriptors" in each window, and patching windows that have "similar" descriptors. Her work mainly deals with textural scenes that occur in nature (natural textures), such as trees, clouds, water, grass, etc. "Texture descriptors" are evaluated in both the spatial domain and the frequency domain.

It may be noticed that most of the methods discussed so far concentrate on at most one problem involving either statistical textures or structural textures, but not both. The decision theory method which we introduce in this paper is very general and with simple extensions and in combination with other procedures we have developed[18] it is quite capable of dealing with several problems involving both classes of texture.

3. DESCRIPTION OF THE DECISION THEORY METHOD

3.1. Texture Recognition as a Statistical Decision Problem

To serve as an introduction to the method, let us initially attack the problem of texture recognition. Texture recognition can be basically viewed as a statistical decision problem where, as in a single trial of a psychophysical experiment, a cycle begins with the presentation of textural scenes and ends with the response of the decision maker. The type of responses expected of the decision maker varies with the problem. For example, in a typical recognition problem, given a textural scene T_Σ, a decision is to be made if it belongs to T^1 or T^0. Here T^1 represents a family of visual scenes consisting of a particular texture, or one of the hypotheses. The alternate is T^0, which does not contain that particular texture. The anticipated response here would be either 1 or 0, representing "signal" and "noise," respectively. This is the binary case and the decision problem can be extended to a case with multiple hypotheses.

3.2. Description of the Scheme

Ideally, each texture is composed of one single pattern, "the unit cell," and if one is able to extract it, the texture recognition problem would be reduced to comparing unit cells. But as we have already discussed, things need not be so simple: more than one unit cell may be present in a given scene, the placement rules may be difficult to extract, or no unit cell may be detected.

We avoid these problems by defining our own universe of local patterns of selected size, shape, and orientation. We search for the occurrences of these "known" patterns in the texture families presented for analysis. Using the elements of statistical decision theory, we extract disjoint sets of local patterns that characterize each family at the same time distinguishing it from the others. Extraction of these disjoint sets of patterns is the basic first step in the decision theory method (DTM). It is possible to determine whether or not the texture families are distinguishable when we use the local patterns defined by templates of selected size, shape, and orientation. If they are not, we can change the shape or increase the size or both to get better performance.

In what follows, we define some of the basic terms needed for further presentation of the method.

3.2.1. Local Patterns (Events)

We define a template of selected size, shape, and orientation centered around a point which samples the patterns from the scene of analysis. Each possible pattern is an n-tuple of gray levels of the nearest n neighbors of the given point and is represented as the n-dimensional vector, e.g., $e = (x_1, x_2, \ldots, x_n)$, and is regarded as an "event." All possible local patterns define the universe of events, E^U. For example, if the digital picture was quantized to h gray levels and the sampling template has n-pixels, then there are h^n elements in E^U. Figure 1 shows the number of elements in the universe for the templates shown.

3.2.2. Likelihood Ratio

The "events" just described now serve as the local evidence for the decision maker at the local level. The "likelihood ratio" of an event is a single number which is an indicator of the strength of evidence that the occurrence of that particular event would provide for the presence of the signal. More precisely, it is the ratio of the probability of the occurrence of the event in T^1 to that in T^0. This is estimated from the "training samples" provided from both the families of T^1 and T^0.

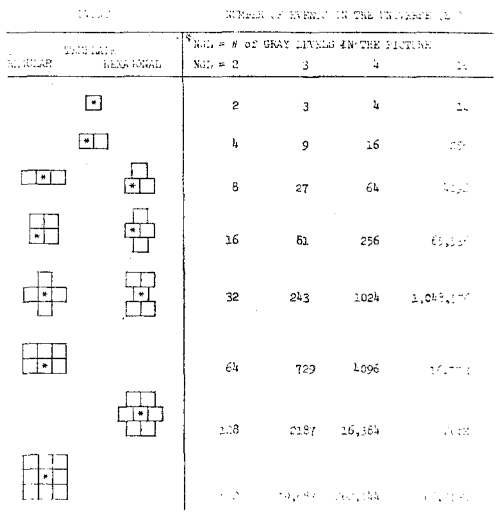

Fig. 1. Total number of events in the universe for various templates.

Let E^1 and E^0 be the sets of events obtained from scanning T^1 and T^0 with a given template, respectively. Let $n_1(e_k)$ be the number of occurrences of event e_k in T^1, $n_0(e_k)$ the number of occurrences of event e_k in T^0, nT^1 the number of events in T^1, nT^0 the number of events in T^0, and $P(e_k \mid T^1)$ the probability of occurrence of e_k in T^1, i.e., the probability of e_k conditional on T^1:

$$P(e_k \mid T^1) = n_1(e_k)/nT^1$$

$P(e_k \mid T^0)$ is defined similarly.

Then the "likelihood ratio" of the event e_k is

$$\mathrm{LR}(e_k) = \gamma P(e_k \mid T^1)/P(e_k \mid T^0)$$

where γ is a normalization factor which compensates for the intrinsic probabilities.

The a posteriori probability, that is, the probability of truth of the hypothesis at a local level conditional on the occurrence of the event, can be shown to be

$$P(T^1 \mid e_k) = \mathrm{LR}(e_k)/[1 + \mathrm{LR}(e_k)]$$

3.2.3. *Decision Goals: Optimal Decision Making*

There are four outcomes in the binary signal detection problem: two errors and two correct decisions. A "false alarm" occurs if the choice is T^1

when it should have been T^0 and a "miss" is the converse. A "hit" is a correct choice of T^1, and the correct choice of T^0 is known as a "correct rejection." There may be different values associated with the correct decisions and different costs associated with the errors.

There can be many alternative decision goals. One of the goals, for instance, is to maximize the expected value. The "expected value" is the sum of the four terms, each representing the value or cost associated with an outcome weighted by its probability. Other decision goals could be to: maximize a weighted combination, maximize the percentage of correct responses, satisfy the Neyman–Pearson objective, and so on. These are discussed in detail by Green and Swets[12] and they prove that whatever is the decision goal, the likelihood ratio criterion is the optimal decision rule, that is, to choose T^1 if $\mathrm{LR}(e_k) > \beta$, where β is a positive number. β may vary for each decision goal.

With this background, we are ready to partially define a local categorizer ψ_R on the basis of a training set of information: Let E^U be the event space and

$$F^{1\beta} = \{e \mid e \in E^1 \cup E^0 \text{ and } \mathrm{LR}(e) > \beta\}$$
$$F^{0\beta} = \{e \mid e \in E^1 \cup E^0 \text{ and } \mathrm{LR}(e) \leqslant \beta\}$$
$$F^* = \{e \mid e \in E^U \backslash E^1 \cup E^0)\}$$

Then, define ψ_R by its acceptance set R, i.e.,

$$\psi_R(e) = 1 \qquad \text{if} \quad e \in R$$

where $F^{1\beta} \leqslant R \leqslant F^{1\beta} \cup F^*$ and $R \cap F^{0\beta} = \varnothing$.

Note that the determination of which events in F^* are in R has not been made at this point. These represent the "don't-care" events and their assignment will be deferred until later.

3.2.4. Operating Characteristics Curve

The operating characteristics (OC) curve (Fig. 2) is a useful device for observing and predicting the behavior of these categorizers. To make the curve, each event $\in E^1 \cup E^0$ is regarded as a two-component vector with $x = p(e \mid T^0)$ and $y = p(e \mid T^1)$. An ordering can be imposed on these vectors by sorting them in descending order by the likelihood ratios of the e's. The curve is generated by placing the tail of the first vector at the origin and then concatenating the rest in order (Fig. 2a). The OC displays several useful items of information in an easy to see form. For one thing, the training-set performance of a categorizer for each value of β is shown directly, since for each threshold, the y coordinate is equal to $\sum_{\{e \mid \mathrm{LR}(e) > \beta\}} p(e \mid T^1)$

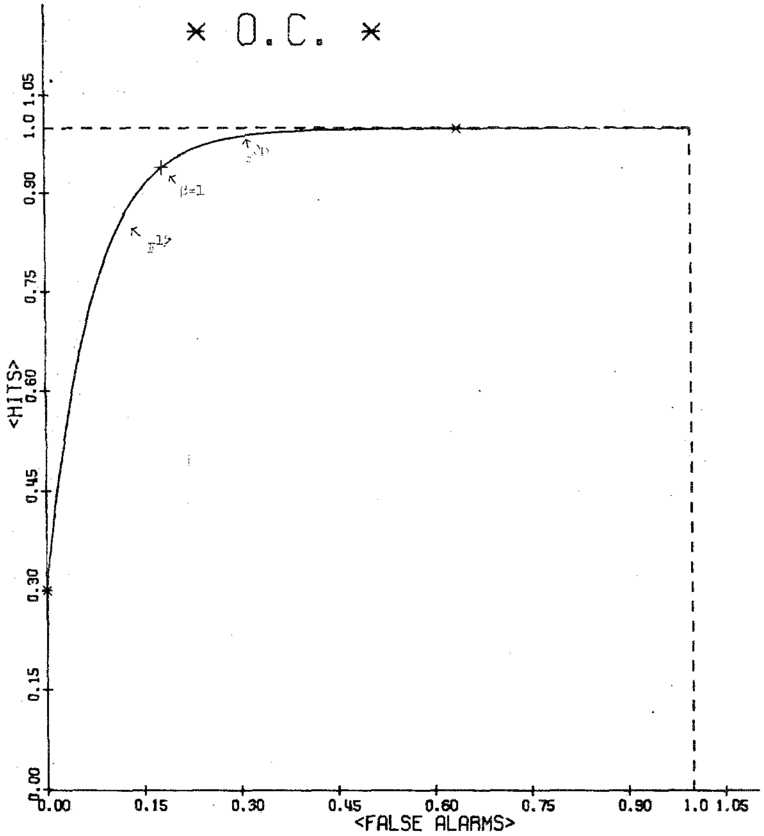

Fig. 2a. Operating characteristic curve.

and the x coordinate is equal to $\sum_{\{e|\mathrm{LR}(e)>\beta\}} p(e \mid T^0)$. The point on the OC corresponding to a given value of β is easy to find, since it is the head of the vector with slope equal to β. (Note that β has only a discrete number of values.) The OC also provides a measure of the inherent separability of the textures in the training set. The area A under the curve is equal to 0.5 if the textures are indistinguishable (all events occur with equal probability in both textures) and is equal to 1.0 if the textures are perfectly distinguishable (all events occur in one or the other texture but not both).

The OC curve can be displayed in another form, by plotting it on probability scales, that is, on axes scaled linearly for normal deviate (Fig. 2b). This type of representation will be useful when the density functions of the events extracted from T^1 and T^0 are of certain forms, such as Gaussian, exponential, etc. The OC curves will be straight lines for Gaussian and exponential distributions. In such cases the measure of separability can be

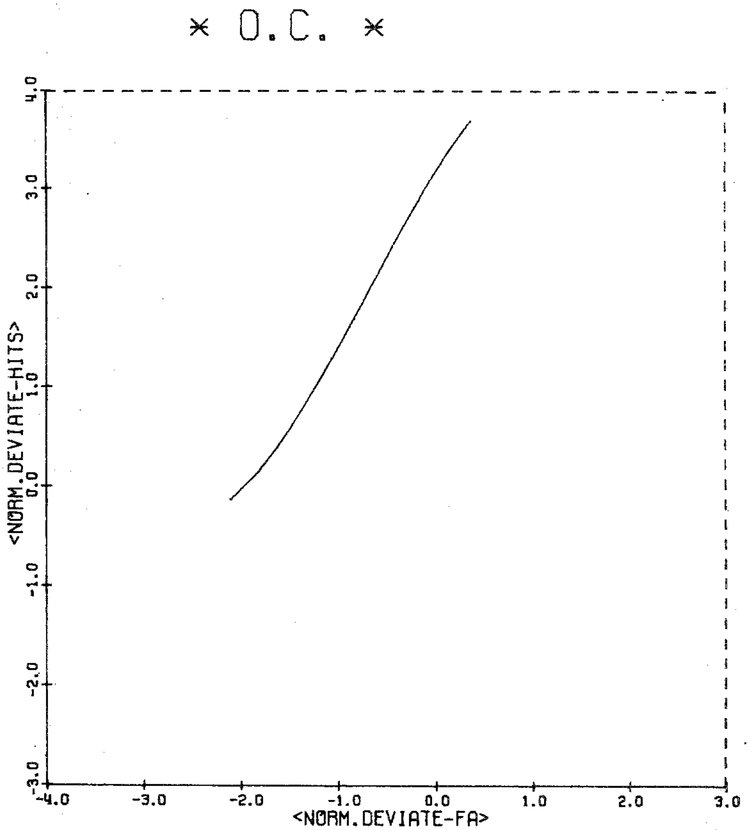

Fig. 2b. Operating characteristic curve on probability scales.

specified by a single parameter.[44] This measure, unlike A, permits the reconstruction of the OC curve from which it is derived. But the value of A has another property useful for conceptual purposes, in that it is equal to the percentage of correct choices that the system will make when attempting to select, from a pair of events, one extracted at random from T^1 and the other extracted at random from T^0, the one that belongs to T^1.[12,44]

3.3. Coloring the Scene of Analysis and Classification Scheme

It is sometimes convenient to have the following transformation of the scene. A point in the transformed plane is marked 1 (or dark) if the corresponding point in the scene of analysis has an event in the acceptance set R. Otherwise, it is marked 0 (light). The resultant picture is the "colored" image of the scene of analysis. An ideal case: colored images of samples of T^1 would be all "dark" and those from T^0 would all be "light."

In practical cases classification of the scene would be achieved based on the global characteristics, such as distribution of 1's and 0's in the "colored" scene. The thresholds needed for the classification can be estimated from training samples.

3.4. Multiple Textures

Thus far we have considered only binary signal detection (T^1 and T^0). The scheme can be extended to multiple textures in the same way that the binary signal detection is extended to that of M-ary signal detection. When there are M different classes of texture present ($T_1, T_2, \ldots, T_M$), we need to calculate at least $M-1$ "likelihood ratios" from which other likelihood ratios can be calculated. Here

$$\mathrm{LR}_{ij}(e_k) = \gamma_{ij} P(e_k \mid T_i)/P(e_k \mid T_j)$$

The universe of events will be partitioned into M-disjoint acceptance sets. An event e_k will be included in the R_i acceptance set, iff

$$\mathrm{LR}_{ij}(e_k) > 1 \qquad \text{for all} \quad i \neq j$$

In general for satisfactory results, for larger M, we need a larger universe of events, which means more computation time.

3.5. Texture Discrimination

We can mark the various textural regions in a composite scene if we are given the samples of textures that might be present in it. The procedure is the same as before: Extract disjoint acceptance sets of events representing each texture, and "color" the composite scene with each textural region marked with a different appropriate "color."

3.6. Analysis of Color Images

Thus far we have discussed only black and white pictures. In practice we come across many color images. The spectral (color) information appears to be very valuable and is used in many classificatory schemes involving natural biological images[46] and also in remote sensing technology for photointerpretation. We can perform the analysis of color images using the decision theory method by modifying the interpretation of the "event." Here the event is allowed to extract the spectral information of each point defined in the template. One way is to codify the "spectral signature" of each point; the event extracts the k-tuple of values $\{S_{\lambda 1}, S_{\lambda 2}, \ldots, S_{\lambda k}\}$, where λ_i

is a discrete frequency in the visible spectrum and k is finite. Thus the "event" is an $(n \times k)$-dimensional vector for an n-point neighborhood as defined by the template. After defining the event in this manner, we can proceed with the decision theory method for the analysis of chromatically textured scenes. Normally we use a triplet of values of S_λ for each point, corresponding to colors red, green, and blue in order to match the performance of the human eye.

3.7. Generation of Textural Feature Detectors Using the Concepts of Interval Covering Theory

An earlier version of the decision theory method with main emphasis on the generation of "intervals" that act as texture feature detectors was presented by Read and Jayaramamurthy.(34) To preserve the continuity of the present paper, we review the concepts of interval covering theory very briefly. For further details the reader is referred to Refs. 27, 28, and 34.

In the case of binary signal detection, the events from $F^{1\beta}$ can be considered a "true" set and those from $F^{0\beta}$ are considered a "false" set. The disjunctive normal form can be expressed as $\vee_i \xi_i(e_i)$, where ξ_i is a predicate that has output "true" when the input is a particular event, e_i, from $F^{1\beta}$, and output "false" if it is from $F^{0\beta}$. The symbol $\vee$ represents the logical "OR" of the predicates. Events from F^* are considered as don't-care events. McCormick and Michalski have developed "interval covering theory" as a generalization of switching theory(28) which permits the transplantation of much of the minimization machinery already in existence. In particular, Michalski's A^Q algorithm(27) for generation of quasiminimal covers can be used.

3.7.1. Notation

To explain the method, it is necessary to introduce a few items of notation from Ref. 27: E^U is the event space as before. That is, the set of all events

$$e = (x_1, x_2, ..., x_n) \qquad \text{where} \quad 0 \leqslant x_i \leqslant h - 1.$$

A *literal* ${}^{a_i}X_i^{b_i}$ is the set of all events $e \in E^U$ whose ith component lies between a_i and b_i:

$${}^{a_i}X_i^{b_i} = \{(x_1, x_2, ..., x_n) \mid a_i \leqslant x_i \leqslant b_i\}$$

An *interval* is a set-theoretic product of literals,

$$L = \bigcap_{i \in I} {}^{a_i}X_i^{b_i}, \qquad I \subseteq \{1, 2, ..., n\}$$

The interval represents a "box" in hyperspace which includes all events between $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$. Note that components not specified by the interval are free to take on any integer value in $[0, h-1)$.

An *interval cover* of the set $F^{1\beta}$ against $F^{0\beta}$ is defined as a union of intervals L_j such that

$$F^{1\beta} < \bigcup L_j < F^{1\beta} \cup F^*$$

Thus an interval cover contains all the events in $F^{1\beta}$, possibly some in F^*, but none in $F^{0\beta}$. However, the interval cover will represent this partitioning of the space of possible events much more concisely than just enumerating all the events in the acceptance set for T^1. Also, the interval cover can classify events which were not in the training set, because of the inclusion of F^* events in the "boxes."

Thus an interval cover can provide an efficient implementation of the local categorizer described earlier.

3.7.2. Generation of Interval Cover

A quasiminimal interval cover can be generated with the help of Michalski's A^Q algorithm, which is based on what is known as the "disjoint star method." Briefly, the procedure is as follows.

Consider the events in the sets $F^{1\beta}$, $F^{0\beta}$, and F^* as "ones," "zeros," and "don't cares," respectively. The cover is found by an iterative procedure which begins by picking the first "one" encountered in the set $F^{1\beta}$ and discovering all of the maximal intervals that include that "one" but no zeros (an interval "star"). One of these, the interval including the most "ones," is added to the covering set (initially empty). All of the "ones" included in the "star" are temporarily eliminated from $F^{1\beta}$. The iteration continues by selecting the first "one" encountered in $F^{1\beta}$. Eventually all the "ones" have been eliminated. If all the "ones" are included in the covering set, then, the cover is minimal. Otherwise, the cover is extended to include the neglected events, and may not be minimal.

3.7.3. Intervals as Textural Feature Detectors

Intervals were achieved as "boxes" into which events from the acceptance set R were efficiently packed. Since the events in the acceptance set occur more frequently in T^1 than in T^0, the intervals that are nothing but groups of such events have a tendency to define features that are more likely to be found in T^1 than in T^0. It has been found in a practical case that some intervals pick up features like vertical lines, horizontal lines, herringbone patterns, etc. which are readily perceived by human beings. Features extracted by other choices of intervals may be very poorly matched for human perception. Strategies for the choice of "good" intervals remain largely unexplored.

These "intervals" can be treated as 2D filters which detect textural features more common to T^1 than T^0. The normalized "output count" of the filters (normalized number of input "local patterns" from the scene of analysis that fall in the "pass band" of the filter) can be used as a feature vector. In this multidimensional space, we will be dealing with clusters of the scenes from T^1 and T^0 and for classification purposes we can resort to any popular cluster analysis method.

It may be noted that the generation of "interval covers" was possible for the binary case only. Extension of this procedure for the case of multiple textures is reserved for further investigation.

As we have seen, the success of the decision theory method depends heavily on the choice of appropriate template. The OC curve helps us select a "good" template among several possible choices. Sometimes, when we do not meet early success, this search can be very lengthy. Hence there is a need for a method that would help us determine the size, shape, and orientation of the template, preferably by making use of the information extracted from the scene of analysis. One such method is described in the appendix.

4. APPLICATION OF THE DECISION THEORY METHOD

4.1. Texture Recognition

We illustrate the applicability of the decision theory method to the problem of texture recognition with an example. Here T^1 represents a family of textures which consist of grids of different sizes with varying amounts of noise added (Fig. 3a). Some random pictures shown in Fig. 3b belong to the family T^0. All these samples are digitized binary pictures containing 32×32 pixels. We selected a 3×2 template to extract sets of events E^1 and E^0 from these training samples. For convenience, we define the following disjoint sets of events, which we frequently refer to later:

$$F^1: \quad \{e \mid e \in E^1 \text{ and } e \notin E^0\}$$

$$F^0: \quad \{e \mid e \in E^0 \text{ and } e \notin E^1\}$$

$$F^\Phi: \quad \{e \mid e \in (E^1 \cup E^0) \backslash (F^1 \cup F^0)\}$$

The events in F^Φ are arranged in descending order according to their likelihood ratios. The OC curve for this example is shown in Fig. 3c. We selected $\beta = 1$ in order to maximize the percentage of correct responses,[12] and

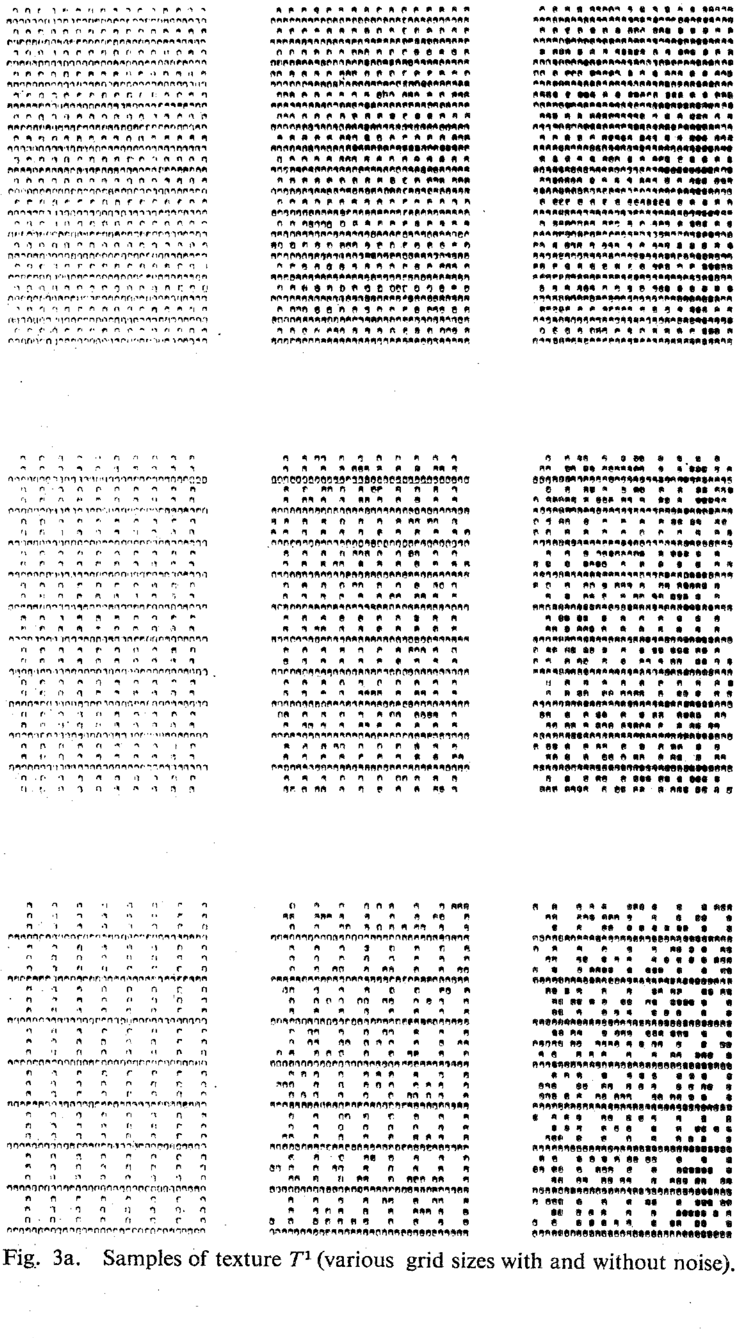

Fig. 3a. Samples of texture T^1 (various grid sizes with and without noise).

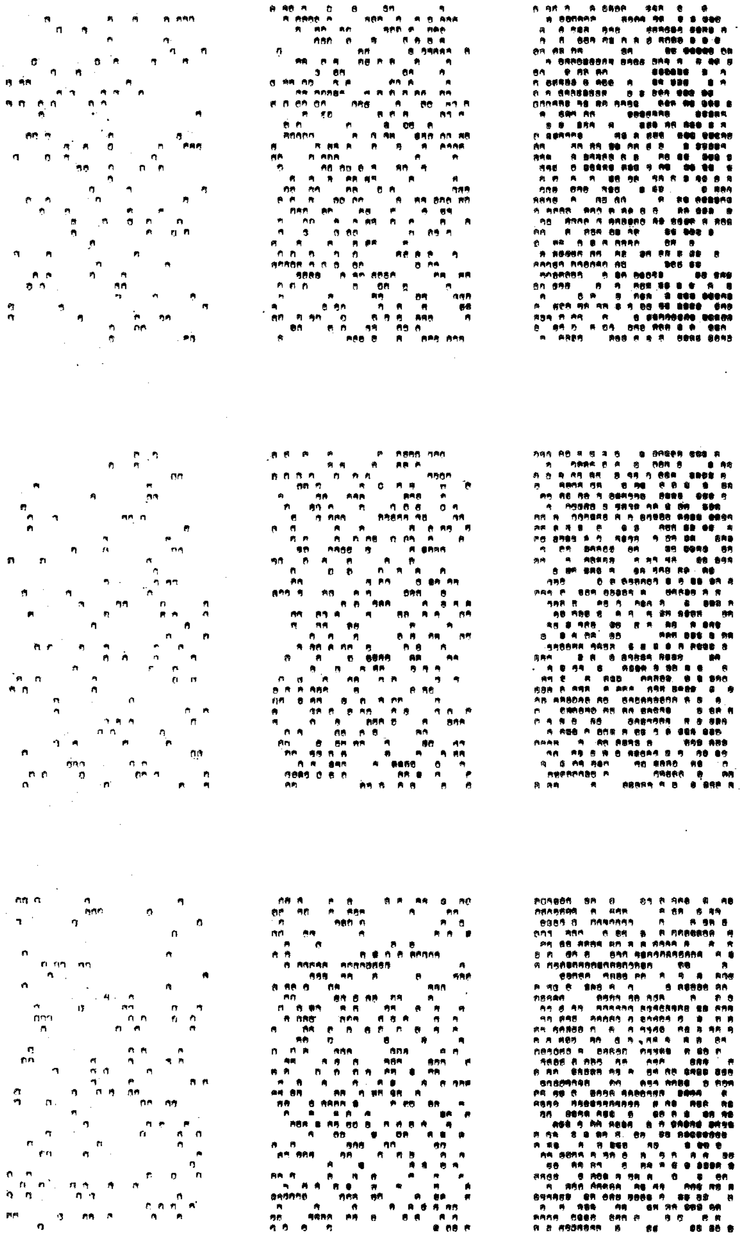

Fig. 3b. Samples of texture T^0 (random pictures).

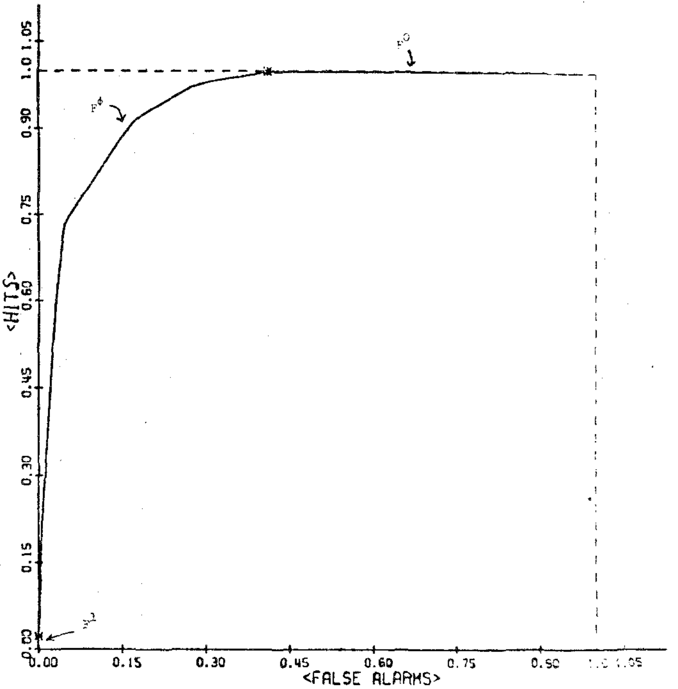

Fig. 3c. Operating characteristic curve for textures shown in Figs. 3a and 3b.

extracted the sets $F^{1\beta}$ and $F^{0\beta}$. As described in Section 3.3, we have colored the training samples and calculated the thresholds on the percentages of 0's and 1's that enable us to correctly classify a maximum number of training samples. Using these thresholds, we attempt to classify the test samples shown in Fig. 3d and 3e. The results of the classification (Fig. 3f) show that all samples in the first set of test samples have been correctly identified. In the second set, as test samples we used grids of larger sizes with added noise. Here, too, the classification is correct except in the presence of excessive noise.

A similar experiment was conducted for recognition of textures of chromatin samples and artifact samples from Pap smears, and is described in detail in Ref. 34.

4.2. Border Extraction

When two textured regions meet, the events picked up along the border can be expected to be different from those that occur in the core of either regions, for the simple reason that they are made up from parts of both the

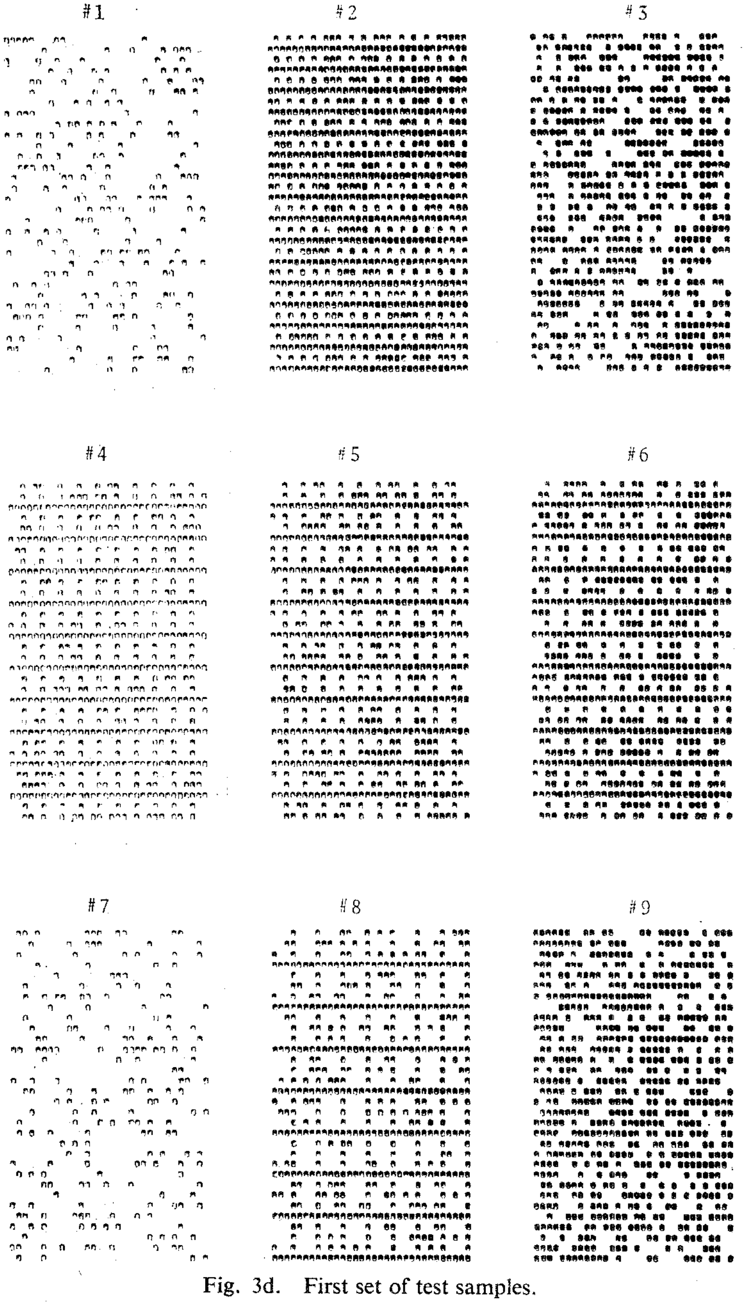

Fig. 3d. First set of test samples.

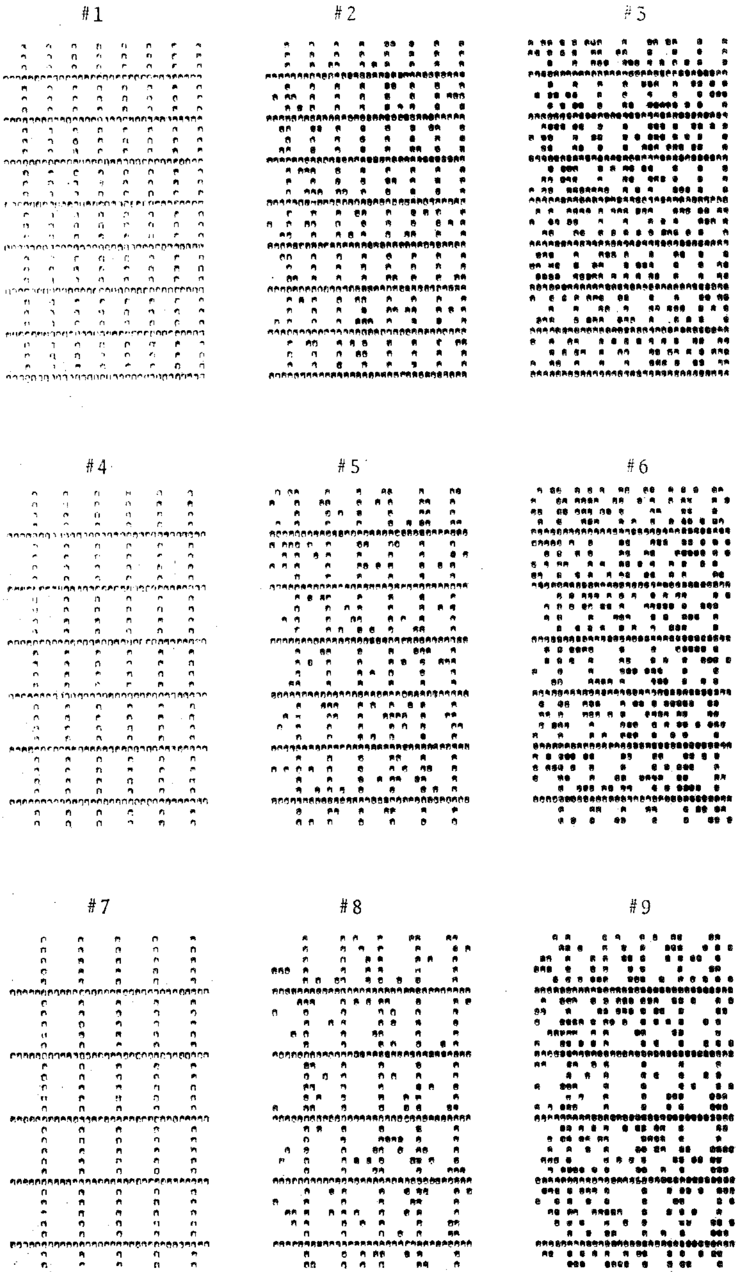

Fig. 3e. Second set of test samples.

```
TEST SET : 1
ENTER CLASSIFY
*** CLASSIFICATION ***
SAMPLE # 1 ===     TEXTURE 0
SAMPLE # 2 ===     TEXTURE 1
SAMPLE # 3 ===     TEXTURE 0
SAMPLE # 4 ===     TEXTURE 1
SAMPLE # 5 ===     TEXTURE 1
SAMPLE # 6 ===     TEXTURE 1
SAMPLE # 7 ===     TEXTURE 0
SAMPLE # 8 ===     TEXTURE 1
SAMPLE # 9 ===     TEXTURE 0
ENTER ROC
```

```
TEST SET : 2
ENTER CLASSIFY
*** CLASSIFICATION ***
SAMPLE # 1 ===     TEXTURE 1
SAMPLE # 2 ===     TEXTURE 1
SAMPLE # 3 ===     TEXTURE 0
SAMPLE # 4 ===     TEXTURE 1
SAMPLE # 5 ===     TEXTURE 1
SAMPLE # 6 ===     TEXTURE 0
SAMPLE # 7 ===     TEXTURE 1
SAMPLE # 8 ===     TEXTURE 1
SAMPLE # 9 ===     TEXTURE 0
ENTER ROC
```

Fig. 3f. Results of classification.

regions. We may say that the "border texture" is different from the textures on either side. We can use this fact to force all the events in the border into a set. Later, when this set is colored, the border will appear distinctly.

Let t_1 and t_0 be the protosamples of textures present in a combined scene T^1. Let us consider the union of t_1 and t_0 as T^0. We proceed as before and obtain F^1, F^Φ, and F^0 sets. It is intuitively clear that F^0 is a null set because t_1 and t_0 are part of T^1. F^Φ contains all events in both t_1 and t_0. F^1 contains only the events that occur in T^1 exclusively, which are nothing but the border events; therefore when we color the F^1 set, we obtain the location of the border (Figs. 4a and 4b).

4.3. Extraction of Texture Regions

Given protosamples of textures present in a composite scene, we should be able to mark each texture region separately. For this purpose we analyze the protosamples and extract disjoint acceptance sets for each texture as described in Section 3.4. We then analyze the composite scene and mark the events belonging to each acceptance set differently. The result of such an experiment on a composite scene (Fig. 5a) is shown in Fig. 5b. Here we notice that each texture region is marked uniformly and differently.

We repeated this experiment with natural textures, this time using the protosamples of the textures of nucleus, cytoplasm, and the background of the brain cells. The result of "coloring" the composite scene is shown in Fig. 6. This is the raw output picture and it is possible to "clean" it, by removing, for example, points that do not have a certain number of neighbors belonging to the same class.

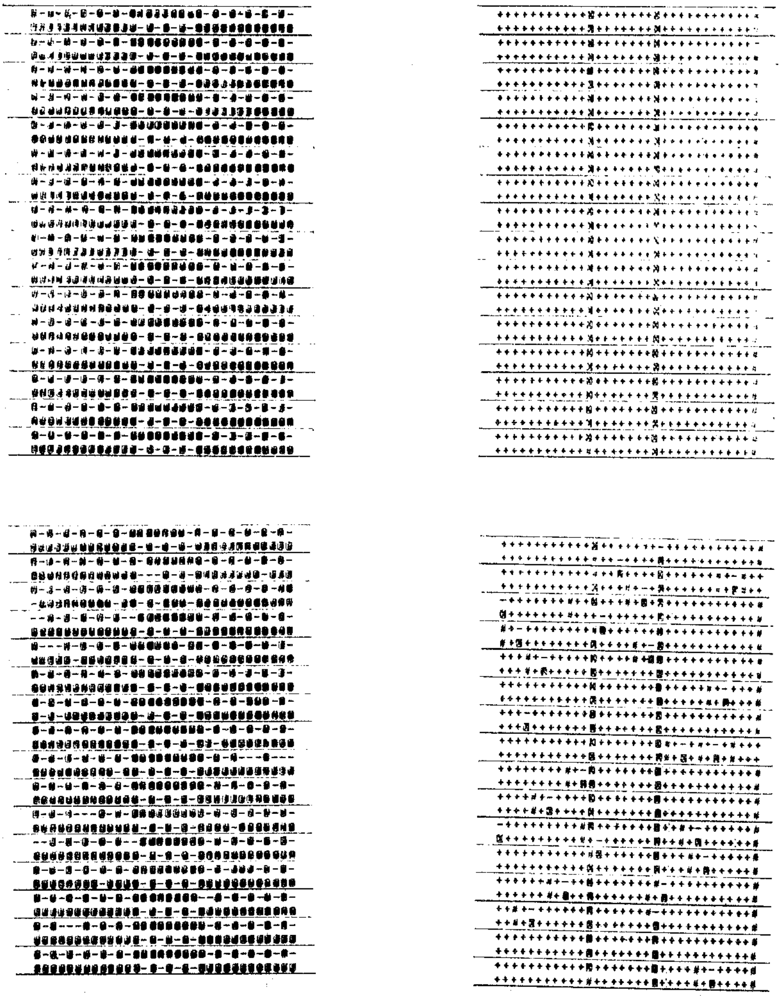

Fig. 4a. Border extraction: example 1. The bottom picture contains "salt and pepper" noise.

4.4. Iteration

After obtaining F^1 and F^0 for a given pair of textures T^1 and T^0, if we go back and color the original scenes as described in Section 3.3, we obtain T_c^1 and T_c^1. This output pair is more easily distinguishable than the original pair. As a matter of fact for an ideal case, T_c^1 and T_c^0 should be uniform and in completely different colors. When this is not the case, as happens in general, we can repeat the whole process using the colored pair as input pictures.

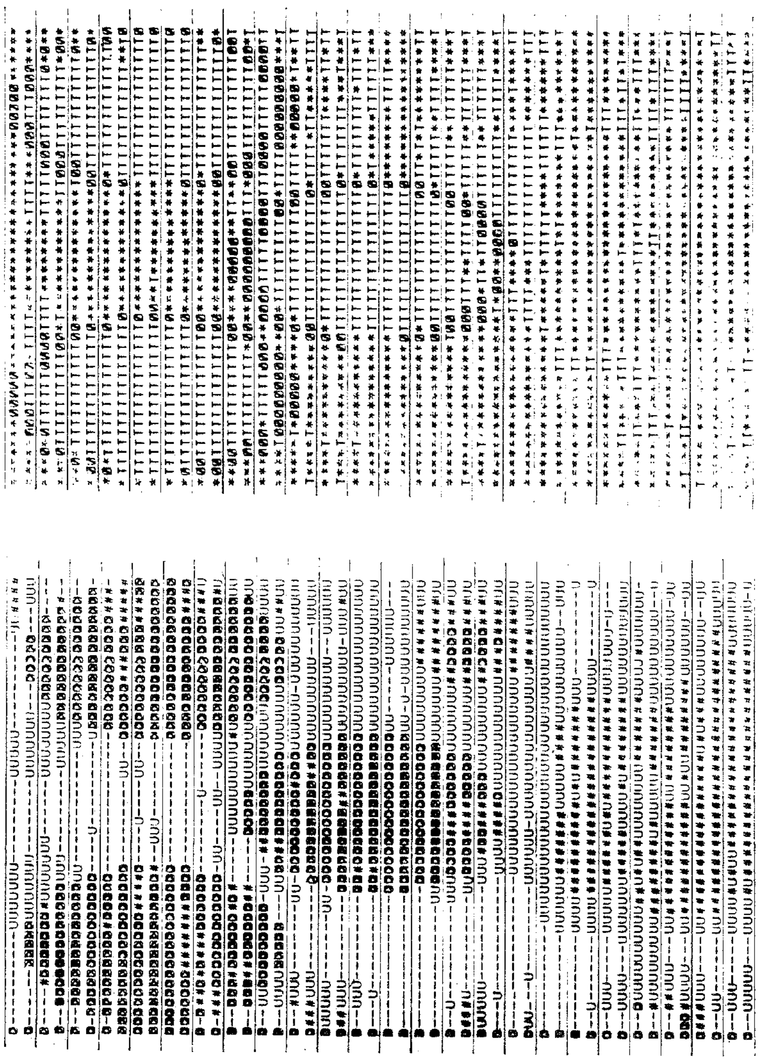

Fig. 4b. Border extraction: example 2.

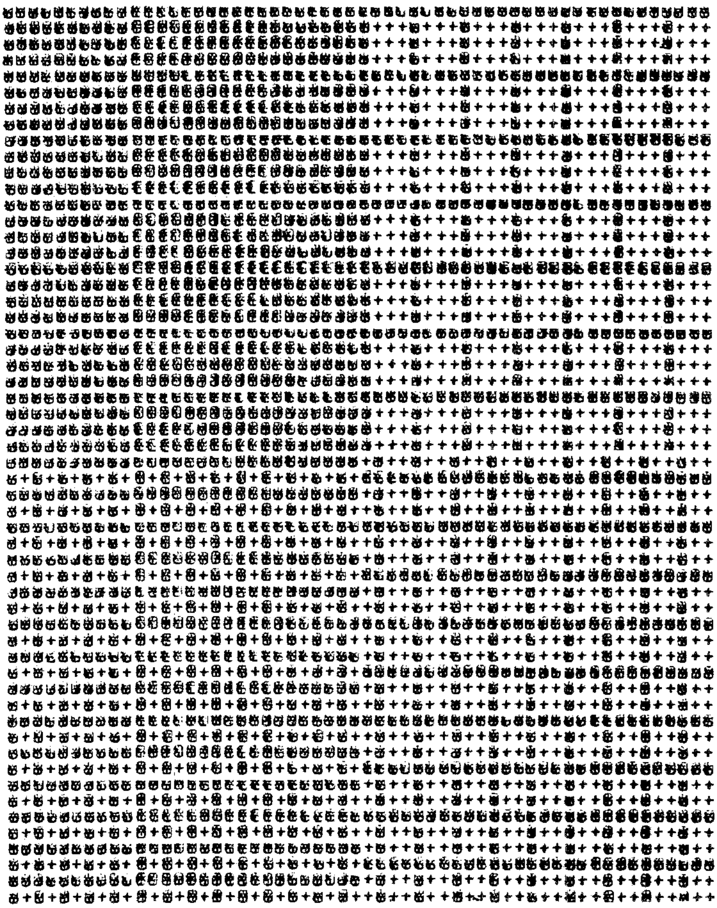

Fig. 5a. A scene with multiple textures.

Let us define an operator C which operates on a pair of scenes:

$$\{T^1, T^0\} \xrightarrow{C} \{T_c^{\ 1}, T_c^{\ 0}\}$$

By repeated application of operator C,

$$\{T^1_{c^{n-1}}, T^0_{c^{n-1}}\} \xrightarrow{C} \{T^1_{c^n}, T^0_{c^n}\}$$

we can go on iterating until we obtain a satisfactory separation in the output pair (Figs. 7a and 7b). The improvement in OC curve toward the ideal case with each iteration can be seen in Fig. 7c.

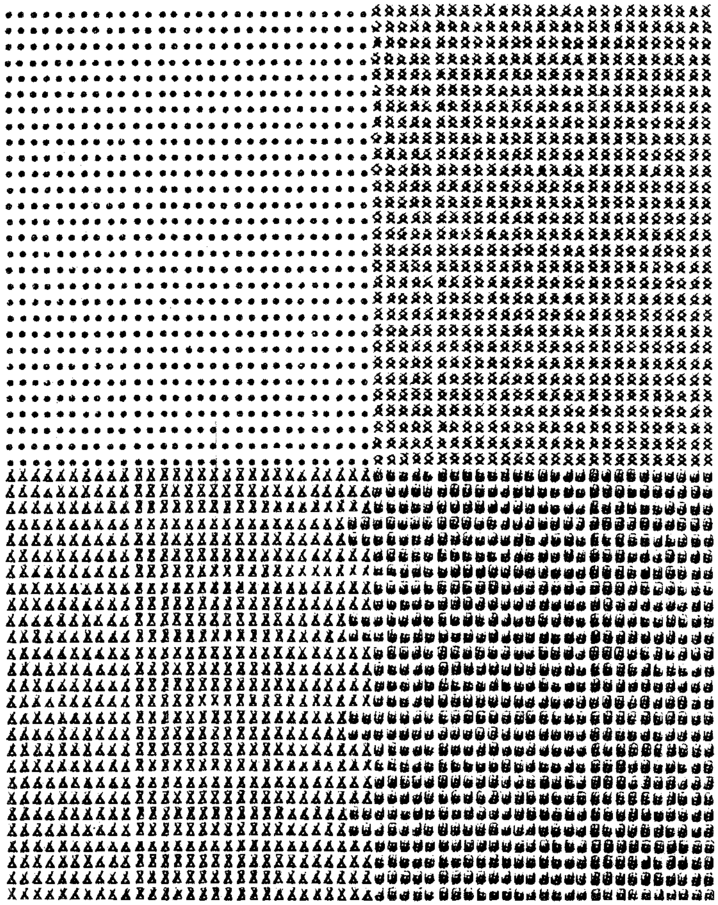

Fig. 5b. Filtered image of the scene shown in Fig. 5a (each texture region marked differently).

It is clear that each point in the output picture is a function of neighbors around a given point in the original picture. As a result of iteration, the points in the output plane depend (maybe weakly) on a larger and larger area in the original scene. That is to say, we are operating on the original pictures at a more and more global level with each iteration.

The test picture also will be colored iteratively using the results obtained at each iteration and can be subjected to the filtering process with the filters obtained at the last iteration that has given satisfactory separation of the original pair. At this point a decision will be made as to which scene the test scene belongs.

Fig. 6. Filtered image of the brain cells (the background, cytoplasm, and nuclei have been marked differently).

4.5. Interval Complexes

We obtained eight interval complexes with T^0 containing regular herringbone textures and T^1 containing a set of textures that do not contain a herringbone pattern. We used binary pictures and a 3×3 template in this analysis. The intervals are shown in Fig. 8a. These intervals, acting as 2D filters, are supposed to block regular herringbone patterns. Figure 8b shows the multiple texture scene. It contains a herringbone pattern slightly above the lower right-hand corner. Figure 8e shows the union of the output of all eight filters for the scene of analysis. Figures 8c and 8d show the output of individual filters for the same input scene. It can be seen that features like horizontal lines and vertical lines have been passed through.

Derivation of interval complexes while experimenting with biological textures is explained in detail in Ref. 34.

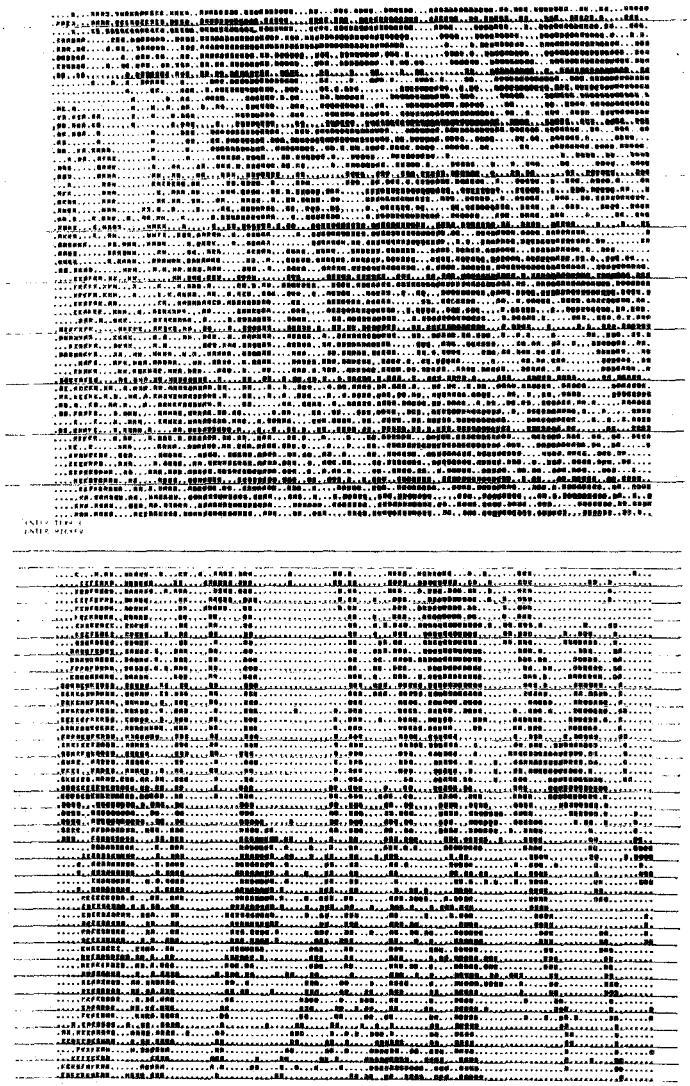

Fig. 7a. Textures of straw (T^1) and wood grain (T^0).

5. SUMMARY AND CONCLUSION

In this paper we have presented a method for the analysis of texture suitable for computer implementation.

We have demonstrated that the decision theory method is versatile

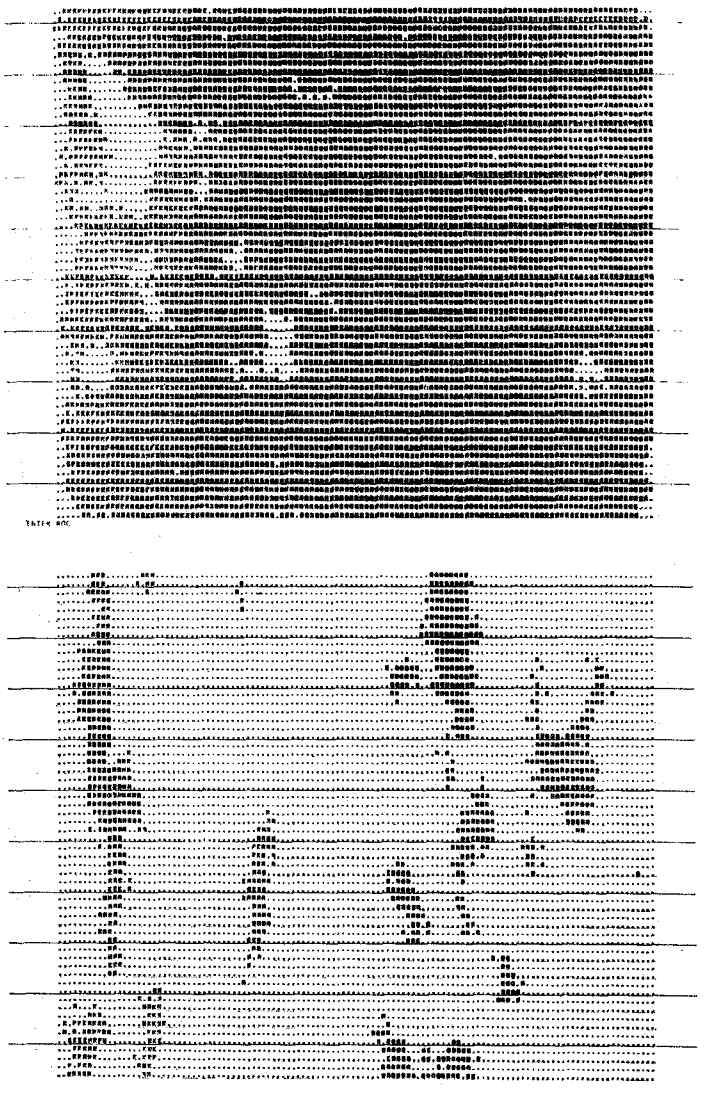

Fig. 7b. Filtered images of Fig. 7a at the end of five iterations.

and can deal with various problems concerning visual texture, such as scene segmentation using textural information, extraction of texture borders, and discrimination/recognition of both spatially and chromatically textured scenes. This method, coupled with the application of interval covering theory, has been shown to be capable of generating 2D filters which act as texture feature detectors for the analysis of the input scene.

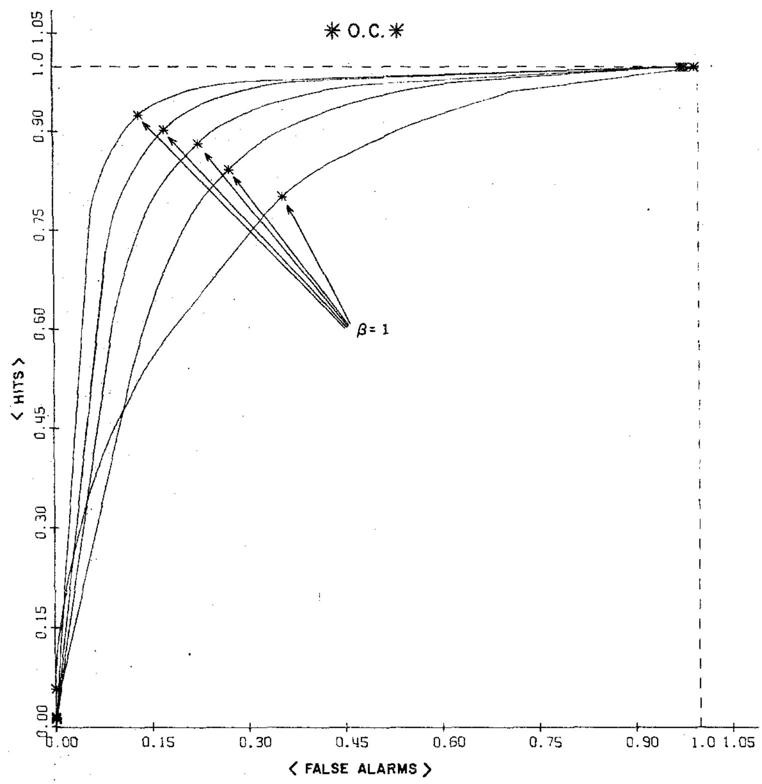

Fig. 7c. Operating characteristic curves for five iterations for textures shown in Fig. 7a.

By virtue of the present interpretation of the "event," the decision theory method of analysis of texture is based on the statistics of local patterns only. It is possible to extend the analysis to the global level by modifying the interpretation of an event. Here each variable in the event is allowed to represent, instead of a local property such as gray level of a point, some measure of a global property. Examples of global variables include the number of edges per unit area, the gray level distributions, and spatial frequency spectrum. These measures are then quantized and digitized. Each variable can assume different discrete range of values.

When a "global event" is defined in this manner we can perform analysis of the scene at global level. The "likelihood ratio" criterion used in this way then provides a better "similarity criterion" than those found in other methods performing similar analysis.[2,16,29]

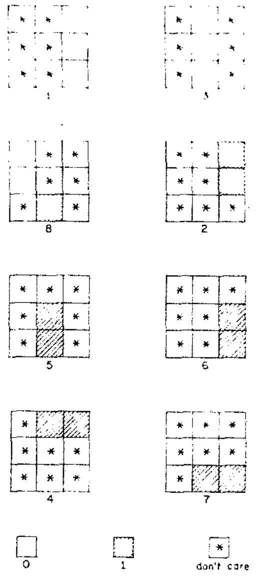

Fig. 8a. Interval complexes (2D filters) for T^1 = random texture and T^0 = herringbone pattern.

APPENDIX. A METHOD FOR DETERMINING THE SIZE AND SHAPE OF THE TEMPLATE

A model has been suggested[24] that views the pixels of a digitized 2D texture scene as a two-way seasonal time series. We show here that the parameters of this model are helpful in determining the appropriate template to be used in the decision theory method of analysis.

We describe the time series model very briefly. For further details the reader is referred to Refs. 18, 24, and 26.

Let

$$\ldots, Z_{t-2}, Z_{t-1}, Z_t, Z_{t+1}, \ldots \tag{A.1}$$

be the time series, which is denoted as $[Z_t]$. Here the Z's represent the pixel values in a discretized TV scan of a texture scene.

A series of values a_t is assumed to be generated from a white noise process with mean zero and variance σ_a^2.

B is the backward shift operator such that

$$BZ_t = Z_{t-1}$$

Hence $B^s Z_t = Z_{t-s}$.

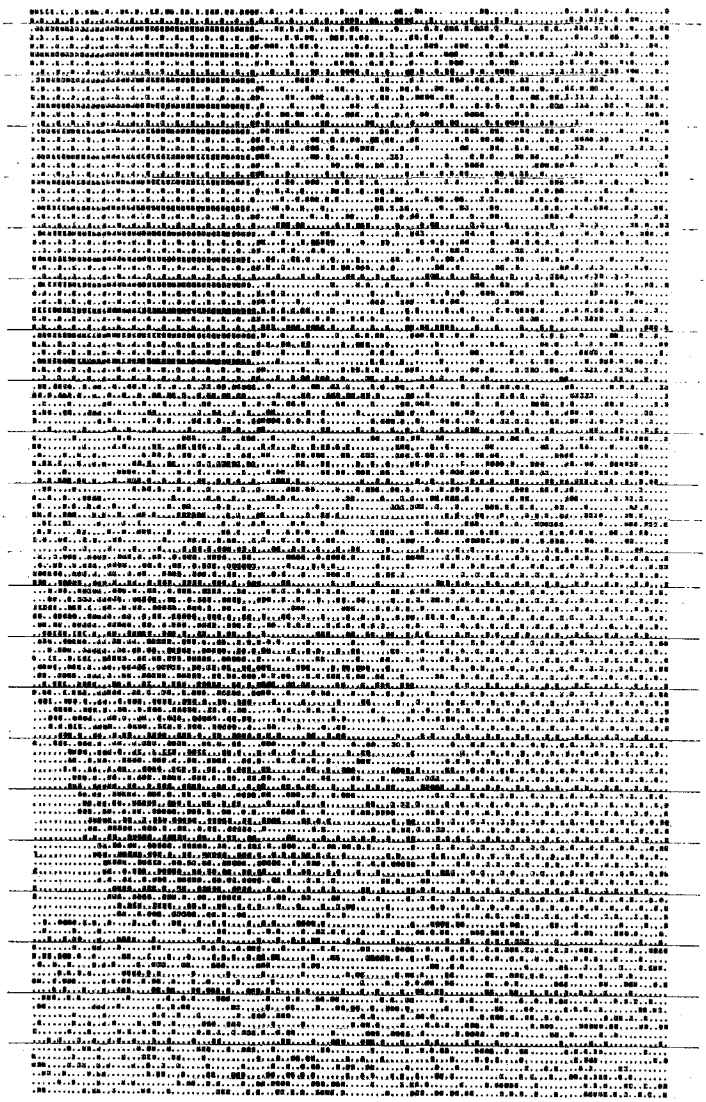

Fig. 8b. Multiple textural scene (input scene).

∇ is the backward difference Operator such that

$$\nabla Z_t = Z_t - Z_{t-1} = (1 - B)\, Z_t$$

$$\nabla^m Z_t = (1 - B)^m\, Z_t \qquad \text{and} \qquad \nabla_s Z_t = (1 - B^s)\, Z_t$$

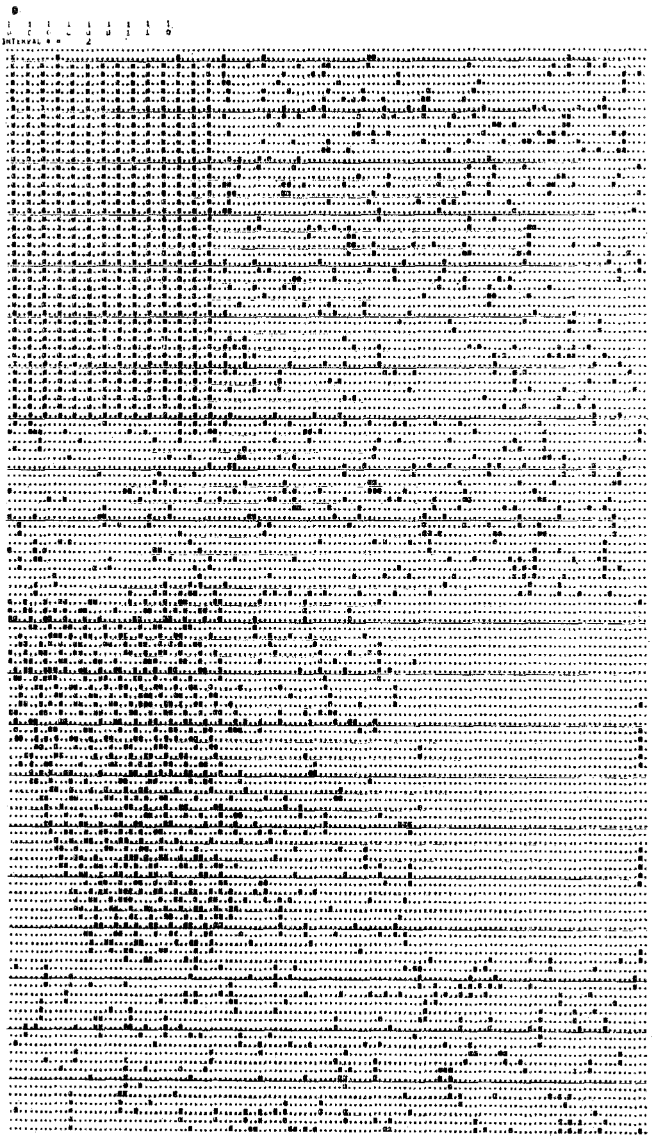

Fig. 8c. Output of one of the filters with the scene shown in Fig. 8b as input.

If the time series $[Z_t]$ shows a seasonal behavior (with period s), then it can be represented by the following multiplicative model:

$$\tilde{\Phi}_p(B)\, \tilde{\tilde{\Phi}}_P(B^s)\, \nabla^d \nabla_s{}^D Z_t = \tilde{\theta}_q(B)\, \tilde{\tilde{\theta}}_Q(B^s)\, a_t$$

where $\tilde{\Phi}_p(B)$ and $\tilde{\theta}_q(B)$ are polynomials in B of order p and q and are known

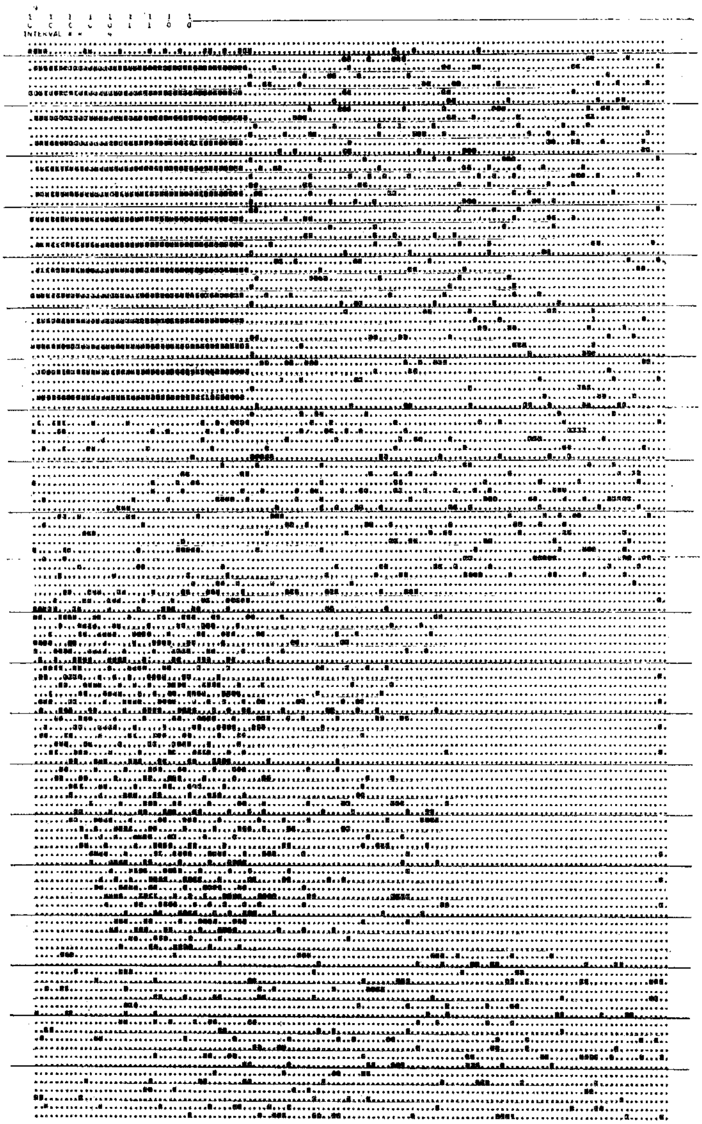

Fig. 8d. Output of another filter for same input.

as "autoregressive" and "moving average" operators, respectively; and $\tilde{\Phi}_P(B^s)$ and $\tilde{\theta}_Q(B^s)$ are polynomials in B^s of order P and Q and are known as "seasonal autoregressive" and "seasonal moving average" operators, respectively.

This model can be easily extended to take care of multiple seasonalities.

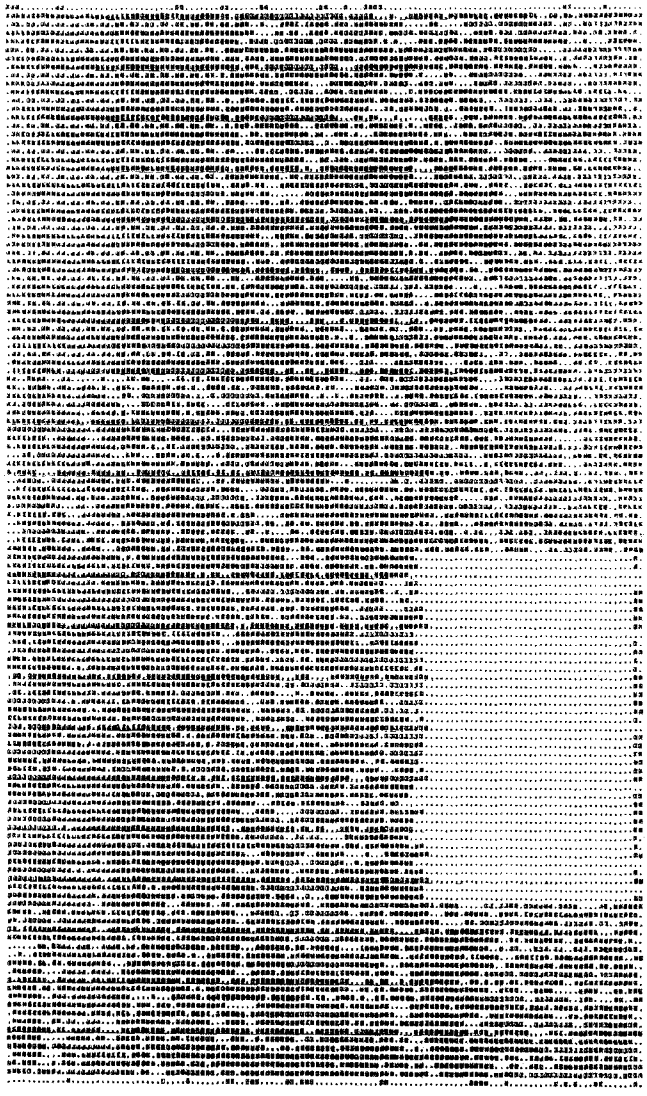

Fig. 8e. Union of the output of all eight filters for the input scene shown in Fig. 8b.

To illustrate, we fit the above model to some natural textures. The texture scenes shown in Figs. 9a and 9b belonging to texture families T^1 and T^2 are represented respectively by the following time series models:

$$(1 + 0.15B^s)\, \nabla_s Z_t = (1 - 0.25B)(1 - 0.5B^s)\, a_{t1} \tag{A.2}$$

$$(1 - 0.80B^s)\, Z_t = (1 + 0.25B)(1 - 0.5B^s)\, a_{t2} \tag{A.3}$$

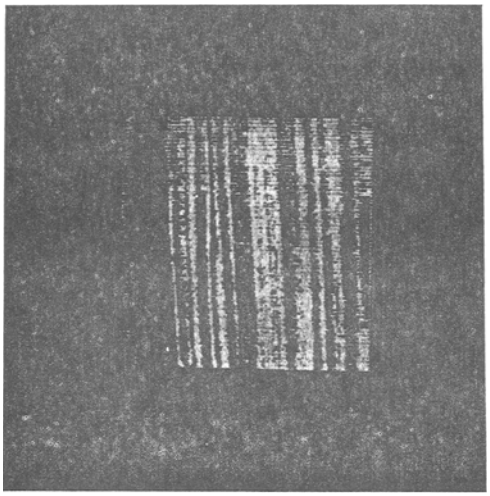

Fig. 9a. Cheesecloth texture.

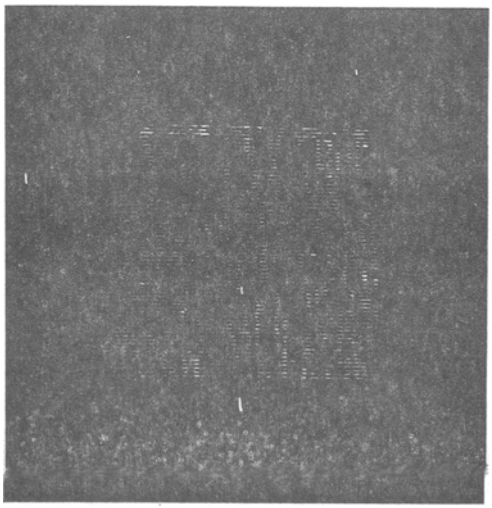

Fig. 9b. Texture of handmade paper.

where σ_1^2 is the variance of the a_{t1}'s, $=1.03$; and σ_2^2 is the variance of the a_{t2}'s, $=3.27$. Here s is found to be the length of row (i.e., 32 pixels in the present case).

We shall rewrite the above models as pure autoregressive models of the form

$$\tilde{\Phi}^i(B)\, Z_t = a_{t_i}; \qquad i = 1, 2 \tag{A.4}$$

The autoregressive operators $\tilde{\Phi}^1(B)$ and $\tilde{\Phi}^2(B)$ are in general rational functions of B. These, in turn, can be approximated by polynomials in B; their coefficients are arranged diagrammatically and shown in Fig. 10. These values indicate the influence of the neighboring pixels in determining the values of Z_t, apart from the residual a_t, whose mean value is zero. Figure 10(d) shows the template that picks up Z_t and its neighboring pixels. The set of pixels picked up by the template constitutes an "event," as defined before. The autoregressive operator operating on Z_t can be viewed as if the "event" picked up by the template is passed through the "filter" formed by the coefficients. The contour of the smallest significant coefficient value determines the upper limit on the size and shape of the template. The selection of a larger template would not yield any new information on "local properties" of the texture scene. Equation (4) indicates that the "events" picked up by this template when passed through appropriate "coefficient filters" yield an output consisting of random shocks a_t, which are normally distributed. From the engineering point of view, we can considerably reduce the size and shape of template, and yet preserve the above-mentioned property (approximately), by selecting a suitable cutoff value of the coefficients.

In the case of texture families T^1 and T^2, the template 6 (shown in Fig. 11) appears to be suitable. The output of the "coefficient filters" C^1 and C^2 (Fig. 10) appears to be fairly random when "events" extracted from T^1 and T^2 are passed through them, respectively. The randomness is tested by plotting the autocorrelation function of the output.

				0.002	-0.010	-0.040	B^{5S}
			-0.001	0.005	-0.020	-0.081	B^{4S}
			-0.002	0.010	-0.040	-0.162	B^{3S}
. . .	0.001	-0.005	0.020	-0.081	-0.325	B^{2S}
. . .	0.001	-0.005	0.021	-0.087	-0.350	B^S
. . .	-0.003	0.015	-0.062	0.250	1.000	1
	B^4	B^3	B^2	B	1	

$\tilde{C}^1$

COEFFICIENTS OF $\tilde{\Phi}^1(B)$

(a)

			-0.001	0.004	-0.018	B^{5S}
			-0.002	0.009	-0.037	B^{4S}
		0.001	-0.004	0.018	-0.075	B^{3S}
		0.002	-0.009	0.037	-0.150	B^{2S}
. . .	-0.001	0.004	-0.018	0.075	-0.300	B^S
. . .	0.003	-0.015	0.062	-0.250	1.000	1
	B^4	B^3	B^2	B	1	

$\tilde{C}^2$

COEFFICIENTS OF $\tilde{\Phi}^2(B)$

(b)

			0.003	0.014	0.022	B^{5S}
		0.001	0.007	0.029	0.044	B^{4S}
		0.003	0.014	0.058	0.087	B^{3S}
. . .	0.001	0.007	0.029	0.156	0.175	B^{2S}
. . .	0.002	0.009	0.039	0.162	0.050	B^S
. . .	0.006	0.030	0.124	0.300	0.000	1
	B^4	B^3	B^2	B	1	

$|\tilde{C}^1 - \tilde{C}^2|$

(c)

. . .	•	•	•	•	•
. . .	•	•	•	•	•
. . .	•	•	•	•	Z_{t-3S}
. . .	•	•	•	Z_{t-2S-1}	Z_{t-2S}
. . .	•	•	Z_{t-S-2}	Z_{t-S-1}	Z_{t-S}
. . .	•	Z_{t-3}	Z_{t-2}	Z_{t-1}	Z_t

TEMPLATE

(d)

Fig. 10. Relationship between the coefficient values of pure autoregressive operators and the size of the template.

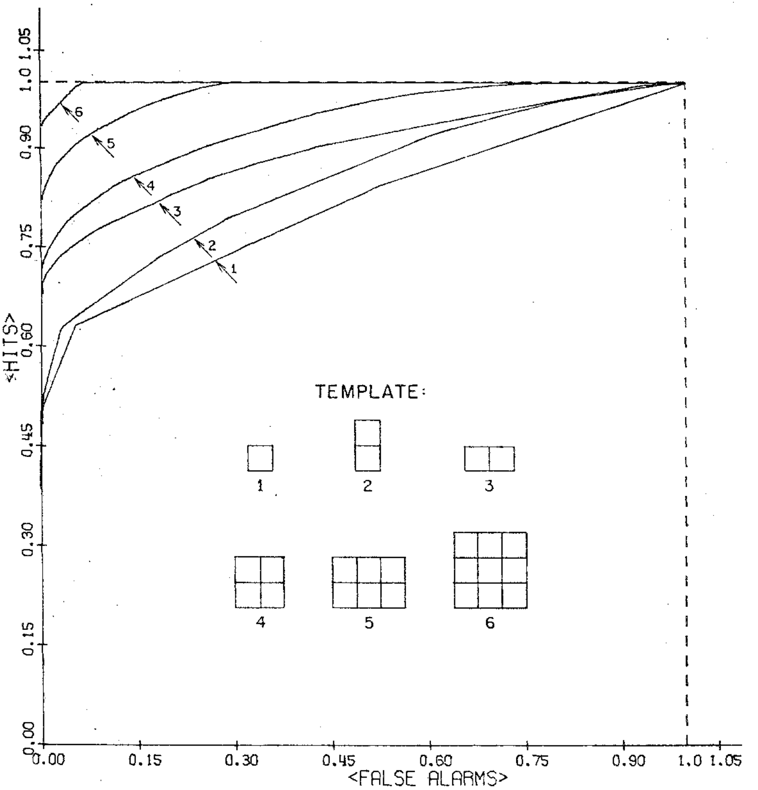

Fig. 11. Operating characteristic curves for textures shown in Fig. 10 (obtained using the set of templates shown).

In the decision theory method, we need to consider one more criterion, viz. the template size and shape should be judicially selected in order to effect good discrimination between the pair of texture families considered. We show how the coefficient values of the autoregressive operators help make this choice, too.

The OC curves for T^1 and T^2 families are obtained (Fig. 11) using the six templates shown. Here we notice that template 3 is more discriminating than template 2 although they are of the same size. The explanation is as follows: Consider the diagram of values $| \tilde{C}^1 - \tilde{C}^2 |$ (Fig. 10c). It can be seen that coefficients of B are more distinctly different than those of B^s. This means that the influence of Z_{t-1} on Z_t is more *discriminatingly* different in T^1 and T^2 than is that of Z_{t-s} on Z_t, and so is the joint distribution of $[Z_{t-1}, Z_t]$ as opposed to $[Z_{t-s}, Z_t]$.

Thus in selecting the size and shape of the template for T^1 and T^2

discrimination we require that the corresponding cells from time series analysis have significant values of $\tilde{C}^1$, $\tilde{C}^2$, and $|\tilde{C}^1 - \tilde{C}^2|$. In addition, we impose optional requirements like connectivity, symmetry, etc. on the resulting template. After making this choice, we check whether or not in its truncated form the corresponding autoregressive operator still satisfies Eq. (4).

REFERENCES

1. D. W. Bacon, "Seasonal time series," Ph.D. Thesis, University of Wisconsin, Madison, Wisconsin (1965).
2. R. Bajcsy, "Computer identification of textured visual scenes," Report CS-321, Computer Science Department, Stanford University, Stanford (1972).
3. Peter Bartels and G. L. Wied, "Tumor cell diagnosis based on stochastic properties of digitized images," *Proc. Two-Dimensional Digital Signal Processing Conf., Columbia, Missouri, 1971.*
4. J. E. P. Box and G. M. Jenkins, *Time Series Analysis* (Holden-Day, San Francisco, California, 1970).
5. P. Brodatz, *Textures* (Dover, New York, 1966).
6. P. Conroy, *Simulation of Texture by Computer Graphics*, TR-14 (Department of Computer Science, University of Toronto, Toronto, 1969).
7. H. R. Flock and A. Moscatelli, "Variables of surface texture and accuracy of space perceptions," *Percept. Mot. Skills* **19**:327–34 (1964).
8. H. R. Flock, "Optical texture and linear perspective as stimuli for slant perception," *Psychol. Rev.* *72*:505–16 (1965).
9. J. J. Gibson, "The reproduction of visually perceived forms," *J. Exp. Psychol.* **12**:1–39 (1929).
10. J. J. Gibson, *The Perception of the Visual World* (Houghton Mifflin, Boston, Massachusetts, 1950).
11. J. J. Gibson, "The perception of visual surfaces," *Am. J. Psychol.* **63**:367–84 (1950).
12. D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics* (Wiley, New York, 1966).
13. H. E. Gruber and W. C. Clark, "Perception of slanted surfaces," *Percept. Mot. Skills Monogr. Suppl. 2* **6**:97–106 (1956).
14. J. C. Hancock and P. A. Wintz, *Signal Detection Theory* (McGraw-Hill, New York, 1966).
15. J. K. Hawkins and G. T. Elerding, "Image feature extraction for automatic terrain classification." *Computerized Imaging Techniques* **VI**:1–8 (1967).
16. J. K. Hawkins, "Textural properties for pattern recognition," B. S. Lipkin and A. Rosenfeld, eds., in *Picture Processing and Psychopictorics* (Academic, New York, 1970).
17. *IEEE Transactions on Computers* (July 1972) (Special issue on two-dimensional digital signal processing).
18. S. N. Jayaramamurthy, "Computer methods for analysis and synthesis of visual texture," Ph.D. Thesis, DCS Report 601, Department of Computer Science. University of Illinois, Urbana, Illinois (October 1973).
19. B. Julesz, "Visual pattern discrimination," *IRE Trans.* **IT-8**:86–92 (1962).
20. B. Julesz, "Texture and visual perception," *Sci. Am.* **1965** (February).

21. B. Julesz, "Cluster formation at various perceptural levels," W. W. Dunn, ed., in *Methodologies of Pattern Recognition* (Academic, New York, 1968).
22. R. S. Ledley, "Texture problems in biomedical pattern recognition," in *Proc. 1972 IEEE Conf. on Decision and Control and 11th Symp. in Adaptive Process, 1972.*
23. B. S. Lipkin and A. Rosenfeld, eds., *Picture Processing and Psychopictorics* (Academic, New York, 1970).
24. B. H. McCormick and S. N. Jayaramamurthy, "Time series model for texture synthesis," *Int. J. Comp. Inf. Sci.* 3(4) (1974).
25. B. H. McCormick and S. N. Jayaramamurthy, "Analysis of texture," DCS Report 531, Department of Computer Science, University of Illinois, Urbana, Illinois (July 1972).
26. B. H. McCormick and S. N. Jayaramamurthy, "Synthesis of texture," DCS Report 566, Department of Computer Science, University of Illinois, Urbana, Illinois (July 1973).
27. R. S. Michalski, "A geometric model for the synthesis of interval covers," Report 461, Department of Computer Science, University of Illinois, Urbana, Illinois (1971).
28. R. S. Michalski and B. H. McCormick, "Interval generalization of switching theory," in *Proc. Third Annual Houston Conf. of Comput. Syst. Sci., Houston, Texas, April 1971* (Extended version in Report No. 442, Department of Computer Science, University of Illinois, Urbana, Illinois, 1971).
29. J. L. Muerle, "Some thoughts on texture discrimination by computer," in B. S. Lipkin and A. Rosenfeld, eds., *Picture Processing and Psychopictorics* (Academic, New York, 1970).
30. R. M. Pickett, "The perception of a visual texture," *J. Exp. Psychol.* **68**:13–20 (1964).
31. R. M. Pickett, "The perception of random visual texture," in W. W. Dunn, ed., *Models for the Perception of Speech and Visual Form* (MIT Press, Cambridge, Massachusetts, 1967).
32. R. M. Pickett, *Perceiving Visual Texture: A Literature Survey* (Aerospace Medical Research Lab., TR-68-12, 1968).
33. R. M. Pickett, "Visual analyses of texture in the detection and recognition of objects," in B. S. Lipkin and A. Rosenfeld, eds., *Picture Processing and Psychopictorics*, London: (Academic, 1970).
34. J. S. Read and S. N. Jayaramamurthy, "Automatic generation of texture feature detectors," *IEEE Trans. on Computers* (July 1972).
35. A. Rosenfeld, "On models for the perception of visual texture," in W. W. Dunn, ed., *Models for the Perception of Speech and Logical Form* (MIT Press, Cambridge, Massachusetts, 1967.
36. A. Rosenfeld, *Picture Processing by Computer* (Academic, New York, 1969).
37. A. Rosenfeld, Y. H. Lee, and R. B. Thomas, "Edge and curve detection for texture discrimination," in B. S. Lipkin and A. Rosenfeld, eds., *Picture Processing and Psychopictorics* (Academic, New York, 1970).
38. A. Rosenfeld and B. S. Lipkin, "Texture synthesis," in B. S. Lipkin and A. Rosenfeld, eds., *Picture Processing and Psychopictorics* (Academic, New York, 1970).
39. A. Rosenfeld and E. B. Troy, *Visual Texture Analysis* (Computer Abstracts, TR 70-116) (Computer Science Center, University of Maryland, Maryland, 1970).
40. A. Rosenfeld, M. Thurston, and Y. H. Lee, "Texture edge, spot and streak detection," in *Proc. Two-Dimensional Digital Signal Processing Conf., Columbia, Missouri, 1971.*
41. P. Smith and O. W. Smith, "Veridical perceptions of cylindricality: a problem of depth discrimination and object identification," *J. Exp. Psychol. 1962*:145–156.
42. R. N. Sutton and E. L. Hall, "Texture measures for automatic machine recognition

and classification of pulmonary disease," in *Proc. Two-Dimensional Digital Signal Processing Conf., Columbia, Missouri, 1971.*

43. G. D. Swanlund, "Design Requirements for Texture Measurements," in *Proc. Two-Dimensional Digital Signal Processing Conf., Columbia, Missouri, 1971.*
44. J. A. Swets, "Effectiveness of information retrieval methods," Scientific Report No. 8, Air Force Cambridge Research Laboratories, USAF, Bedford, Massachusetts (June 1967).
45. H. R. G. Trout, "The description of texture," File No. 832, Department of Computer Science, University of Illinois, Urbana, Illinois (1970).
46. I. T. Young, "Automated leukocyte recognition," in G. L. Wild and G. F. Bahr, eds., *Automated Cell Identification and Cell Sorting* (Academic, New York, 1970).