90-22

COMBINING STRUCTURAL AND STATISTICAL FEATURES IN A MACHINE LEARNING TECHNIQUE FOR TEXTURE CLASSIFICATION

ABSTRACT

This paper presents a method for applying inductive learning techniques to texture description. Local features of texture described as eight attributes have been extracted for each pixel from small windows (5x5, 7x7 or 9x9) centered around the pixel and extra ninth attribute is computed from larger global area (25*25) as co-occurrence matrix parameter. All nine attributes form an *event*, which is essentially a point in a 9-dimensional attribute space. Sets of such events are computed for different texture classes, and the inductive learning AQ algorithm is used to generate a given class description. Such learned descriptions are evaluated using new different texture samples. Results of experiments performed on eight textural images are presented.

ACKNOWLEDGEMENTS

The author thanks Dr. R. S. Michalski and Dr. H. Wechsler for helpful suggestions and useful criticism.

This research was done in the Artificial Intelligence Center of George Mason University. Research activities of the Center are sponsored in part by the Defense Advanced Research Projects Agency under grant, administrated by the Office of Naval Research, No. N00014-87-K-0874, in part by the Office of Naval Research under grant No. N00014-K0226, and in part by the Office of Naval Research under grant No. N00014-88-K0397.

1 INTRODUCTION

Among the most informative properties that play a role in recognizing visual objects are their color and texture. The different textures in an image are usually very apparent to a human observer, but an automatic description of these patterns has proven to be complex. Texture provides very useful information for the automatic interpretation and recognition of the image by a computer. Textural features can be crucial for the segmentation of an image and can serve as the basis for classifying image parts. Many, if not most, objects in one's familiar environment can be recognized on the basis of just these two properties; i.e., without information about their shape, size or other characteristics. While measuring color is relatively easy, "measuring" texture is difficult. Texture may be described as the pattern of the spatial arrangement of different intensities (or colors) with two major characteristics; its coarseness and its directionality.

Traditionally, all methods of textural analysis have taken either the statistical approach [8], in which the statistical properties of the spatial distributions of the gray levels are used as the texture descriptors, or the structure [17] approach which conceives texture as an arrangement of a set of spatial subpatterns according to a certain placement rules. The statistical approach is usually motivated by a lack of strong regular patterns that are obvious in natural textures, and by a conjecture by Julesz [10,11] that second-order probability distributions suffice for human discrimination between two texture patterns.

The structural texture models are best suited to situations in which complete descriptions of individual texture primitives are derivable from the image. This usually means that the texture primitives consist of relatively large numbers of pixels, and that the boundaries of the primitives are consistently discernable. The statistical model usually describes texture by statistical rules governing the distribution and relation of gray levels. This class of models involves the use of statistical tools. The statistical texture models are suitable when the sizes of the texture primitives tend to be on the order of few pixels. The statistical approach works well for many textures which have barely discernible primitives. However, it can also be effective in cases of large texture primitives if the boundaries of the primitives are highly convoluted, or if the interior areas are not completely homogeneous in intensity. A disadvantage of this statistical method is that it is highly dependent on the chosen resolution.

The dichotomy between these two classes, however, is not clear-cut, since statistical tools and concepts are introduced into models which are basically structural, and statistical models can describe pattern-like textures and vice versa. This division is therefore sometimes artificial.

Our interest in this work is to produce symbolic descriptions of texture that are usable at the higher levels of a symbolic reasoning based vision system. These symbolic descriptions are used to isolate the texture primitives themselves in the original texture image. Once the texture primitives have been isolated, we compute "placement rules" via inductive learning techniques. We use this method for both regular and irregular (random) patterns, as we incorporate statistical features of an image. The structural features are derived for each pixel from a small a window using neighboring pixel gray level values as the primitives of an image. As the statistical texture feature, parameters derived from co-occurrence matrices were chosen as the most powerful ones.

The method described in this paper has been tested on a number of different texture images from the Brodatz Album of Textures [4]. We present results of recognition only for eight homogeneous, noisy textures. The results show that the low-level vision symbolic computation can be successfully performed even on those kind of textural images.

Since the co-occurrence matrix techniques for texture description are crucial for our method, the next chapter describes these techniques thoroughly.

2 STATISTICAL METHODS OF CO-OCCURRENCE MATRICES

In these methods texture is defined by a set of statistics extracted from a large ensemble of local picture properties [8]. Limited set of textures can be classified by some very simple statistics such as the gray level first-order statistics. It has been shown that humans are sensitive to second-order statistics. According to Julesz the texture discrimination has revealed the existence of a separate "preattentive visual system" that cannot process complex forms, yet can, almost instantaneously and without effort, recognize differences in a few local features, regardless of where they occur. These features have been called "textons" and are elongated blobs. Examples of the second-order statistics are the gray level co-occurrence matrix and the gray level difference histogram. The spatial gray level dependence method is based on the estimation of the second-order joint conditional probability density functions $f(i,j/d, \phi)$. Each $f(..)$ is the probability of going from gray level i to gray level j, given the intersample spacing d and the direction is given by the angle ϕ. The estimated values can be written in matrix form, the so-called co-occurrence matrix. Let $d=(dx, dy)$ be a vector in the (x,y) plane. For any such vector and for any image $f(x,y)$, we can compute the joint probability density of the pairs of gray levels that occur at pair of points separated by d. This joint density takes the form of an array m, where $m(i,j)$ is the probability of the pair of gray levels (i,j) occurring at separation d. This array is $M*M$, where M is the number of possible gray levels. For discrete image it is easy to compute this array/matrix, by counting the number of times each pair of gray levels (i,j) occur at separation $d=(dx,dy)$. For example, if an image is

1	2	2	3	4
1	1	3	4	4
1	2	3	3	4
2	3	4	3	3
3	3	4	4	3

and $(dx,dy)=(1,0)$, then the corresponding co-occurrence matrix is,

1	2	1	0
0	1	3	0
0	0	3	5
0	0	2	2

where the entry in row i and column j is the number of times gray levels i occurs horizontally adjacent and to the left of gray level j. From a co-occurrence matrix a number of features can be derived. If a texture is coarse, and d is small compared to the sizes of the texture elements, the pairs of points at separation d usually should have similar gray levels. This means that the high values in the m should be concentrated on or near its main diagonal. For a fine texture, if d is comparable to the texture element size, then the gray levels of points separated by d should often be quite different, so that the high values in m should be spread out relatively uniformly. Thus a good way to analyze texture coarseness would be to compute, for various values of the magnitude d, some measure of the scatter of the m values around the main diagonal. Similarly, if a texture is directional, i.e., coarser in one direction than another, then the degree of the spread of the values around the main diagonal in m should vary with in the

direction of d. Thus texture directionality can be analyzed by comparing the spread measures of m for various directions. A set of 14 textural features can be extracted from the co-occurrence matrix. These features contain information about such image textural characteristics as homogeneity, gray-tone linear dependencies (linear structure), contrast, number and nature of boundaries present, and the complexity of the image. Usually four features are derived from the co-occurrence matrix (these four have been used in experiments to find out the one that is mot useful). These are:
Contrast , Angular Second Momentum, Entropy, and Correlation.

1. *Contrast* :

$$CON=\Sigma (i-j)^2 \mathbf{m}(i,j)$$

This is essentially the moment of inertia of the matrix around its main diagonal: it is a natural measure of the degree of spread of the matrix values.

2. *Angular Second Momentum* :

$$ASM=\Sigma \mathbf{m}(i,j)^2$$

This measure is smallest when the $\mathbf{m}(i,j)$ are all as equal as possible; it is larger when some values are high and others are low, as when the values are clustered near the main diagonal, for example.

3. *Entropy* :

$$ENT=-\Sigma \mathbf{m}(i,j)\log \mathbf{m}(i,j)$$

This measure is the largest for equal $\mathbf{m}(i,j)$ and smallest when they are very unequal.

4. *Correlation* :

$COR=\Sigma [ij\mathbf{m}(i,j)-d_x d_y]/(s_x s_y)$, where d_x and s_x are the standard deviations of the row sums of matrix $\mathbf{m}$ and d_y and s_y are analogous statistics of the column sums.
This measures the degree to which the rows (or columns) of the matrix resemble each. It is high when the values are uniformly distributed in the matrix, and low otherwise.

The first, and second-order statistics (co-occurrence matrices) are by far the most used statistical methods for texture discrimination. One problem with the co-occurrence matrices is related to the need to define the distance d and angle ϕ which will fully specify the method. An additional problem related to the n*th* order statistics in general and the second order in particular is the fact that they depend only on the relative position of the n points, but not on their absolute position. For cloud patterns or blood smears this might be a reasonable assumption, since objects can occur anywhere in the scene. For other type of textures as encountered; for example, in chest x-rays or portrait photographs it would not be reasonable to assume position independence. The other problem is that not all features derived from co-occurrence matrix are invariant under monotonic gray-tone transformation. Texture is independent of tone. Of 14 textural features from co-occurrence matrix, the Angular-Second Momentum, the Entropy, the information measure of correlation have the invariance property.

3 STRUCTURAL METHODS

The structural approach assumes that a set of primitive units ("patterns") can be easily identified. It then defines the texture as a combination of such primitives according to different placement rules. The rules of placement of texture primitives are viewed as the rules of grammar. Texture classification is then the determination of whether or not a particular texture exhibits a pattern which belongs to a given language. Structural approaches differ in their choice of primitives, such as pixels, gray level peaks, line segments, or tiles.

One of the early examples of the structural approach is Tomita; et al [19]. They define a texture element ("primitive") as the connected set of pixels. The primitives are characterized in terms of the following properties: brightness, area, size, directionality and curvature. According to these properties the elements are classified into a number of classes, and the above properties are used as the textural features. In recognition of an unknown sample, the textural features are evaluated and compared against those of each learned texture class. Lu and Fu [6] have proposed a texture model in which a texture pattern is divided into fixed-size windows. A tree grammar is used to characterize windowed patterns of the same class. This model was used for texture synthesis as well as discrimination.

Structural methods are appropriate for highly patterned textures, with regular repeated structures. These methods can be divided into the following two classes:

1. Placement rules;
   A grammar is used to describe and build the rules that govern textural structure. The grammar describes how to generate patterns by applying rewriting rules to a small number of symbols. Through a small number of rules and symbols, the grammar can generate complex textural patterns.

2. Primitive extraction;

   Different primitive features are measured (area, perimeter, direction, etc.) To discriminate between textures, features derived from the property value histograms can be used.

There are some major problems with structural methods. First, it is not easy to identify the primitives unless the texture is artificial or not complex. Secondly, the definition that the patterns are repeated according to some prespecified rules should allow for a stochastic change in the replication process, and the same should apply for the patterns themselves.

4 A MACHINE LEARNING APPROACH TO TEXTURE CLASSIFICATION

This approach of texture description employs inductive learning techniques. These techniques are used for automatically extracting the most significant spatial properties of a surface and texture. This method was originally proposed by Michalski [12], and was tested using ILLIAC III computer facilities. Early experiments produced very good results in discriminating even between very similar structural textures. Subsequently, this approach was applied to determine faults in laminates for aircraft wings using ultrasound images [2]; in the TEXPERT system. Recently, a newer version of this system (presented in this paper), that combines structural and statistical features, has been implemented. TEXPERT is designed for recognizing objects by the properties of their surfaces and textures on the basis of two-dimensional digital images (Figure 1). Given a series of images classified by a human tutor into named surface and textural regions, TEXPERT generates a procedure for classifying pixels into these regions. Such a procedures consists of a sequence of operators that transform any given texture into a uniform set of labels characterizing the individual texture type. The major step in the procedure is the formulation of rules characterizing spatial properties of a texture. This is performed by the inductive learning program AQ15 [14] (see appendix). To generate the whole procedure, the system first searches for the most discriminatory image quantization levels (a parameter that represents a inconsistency of class descriptions is used to guide changes of quantization levels so that inconsistency is below the threshold value of this parameter; the highest possible number of quantization levels for the AQ15 attribute is 55). The system then extracts a set of spatial texture samples (*events)* from different texture regions. These events are supplied to the learning program that formulates the rules. In the testing phase, the system applies the procedure to unclassified images and partitions them into different texture regions.

The concept descriptions learned by AQ15 are represented in VL_1, which is a simplified version of the Variable-Valued Logic System VL, and are used to represent attributional concept descriptions. A description of a concept is a disjunctive normal form which is called a cover. A cover is a disjunction of complexes. A complex is a conjunction of selectors. A selector is a form:

[L # R]

where,
L is called the referee, which is an attribute.
R is called the referent, which is a set of values in the domain of the attribute in L.
is one of the following relational symbols: =, <, >, >=, <=, <>.
In the AQ15 program, each generated complex is associated with a pair of weights: *total* (t-weight) and *unique* (u-weight). The following is the example of a complex:

[x1=1..3][x2=1][x4=0][x6=1..7][x8=1] (total:6, unique:2)

The t-weight of a complex is the number of positive examples covered by the complex, and the u-weight is the number of the positive examples uniquely covered by the complex. The complexes are ordered according to decreasing values of the t-weight. There are two methods for recognizing the concept membership of an instance: the strict match and the flexible match. In the strict match, one tests whether an instance strictly satisfies the condition part of a rule (a complex). In the flexible match, one determines the degree of closeness between the instance and the condition part.

In TEXPERT we use truncated descriptions of a given class. The truncated description is the one that has some of its complexes with the lowest u-weight removed. Such a truncated description will not strictly match events that uniquely satisfy the truncated complex. We have found out through different experiments that the truncated description improves the performance of texture classification by reducing the noisy data coverage. The quantitative relationship between the complex truncation and the noise reduction in textural images is not yet known and is a subject for further research.

A special tool, ATEST [16], developed to apply testing examples for assessing rules base performance, is used in this phase. ATEST provides the domain expert with two very important capabilities. First, it allows the expert to rapidly test a rule base on numerous examples under a variety of evaluation schemes. These evaluation facilities provide information about the overall performance of specific rules or examples. This feature is extensively used in the texture discrimination phase. Second, ATEST provides routines that check a rule base

for consistency and completeness. These routines can be used to point out problem areas in the rule base and to help the expert generate new examples. Rule performance is measured by the degree of agreement between the system's and expert's classifications. If the procedure applied to the testing area produces a high degree of agreement with a classification done by an expert, then it is assumed that future images can be processed without the assistance of an expert.

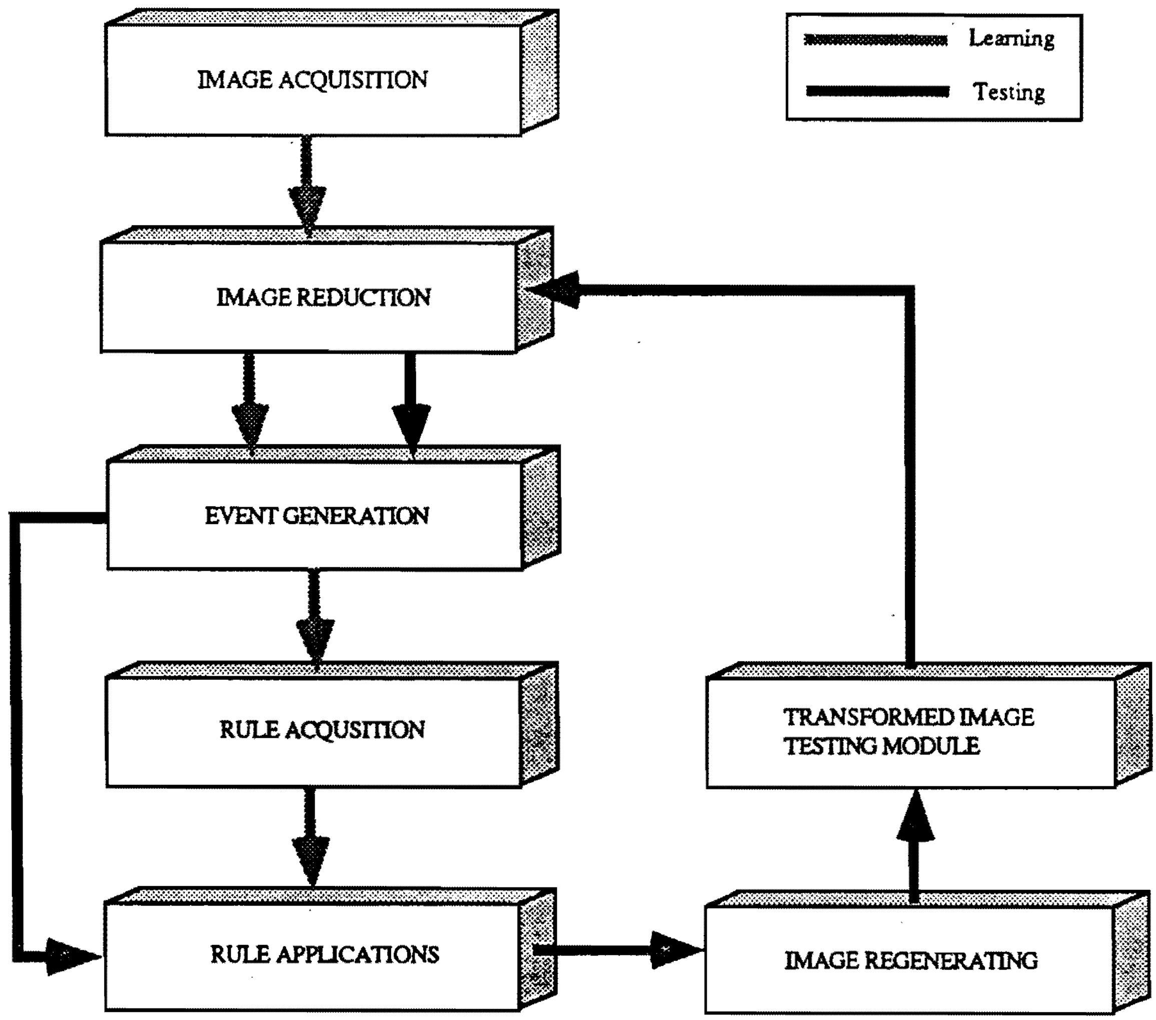


Figure 1. TEXPERT architecture.

5 METHODOLOGY FOR FEATURE EXTRACTION

Many possibilities exist for computing features to recognize texture. The best features will be those which define an attribute space which easily lends itself to partitioning performed by the learning algorithm. This section describes the methodology of computing and combining two categories of features described in previous sections: structural features of neighboring gray-level values and statistics obtained from co-occurrence matrices. This method is the result of numerous experiments performed in the AI Center at George Mason University and is a part of a large research project on learning in robotic vision.

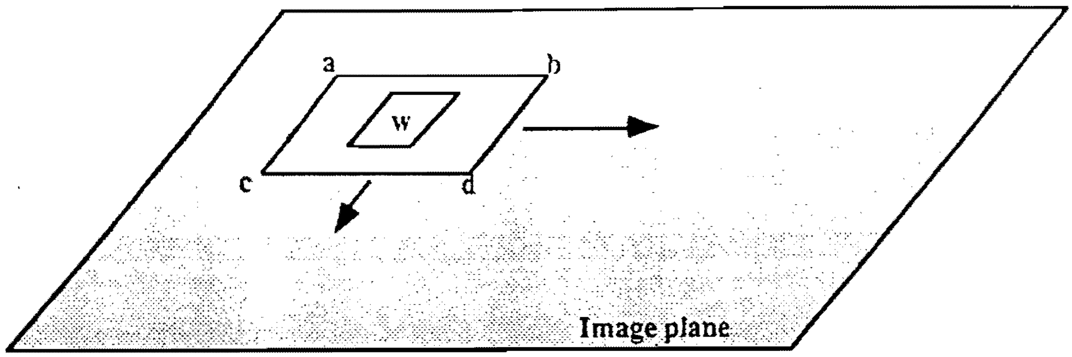
In all experiments we have used textural images acquired by the Perceptics system. Images are 512 by 512 pixels by 8 bits. The main work involved developing the methods to build the most representative texture features for the TEXPERT system. A good set of features applied during the learning phase can reduce the amount of texture information that a system must store. Textural features are very critical for proper "tuning" of TEXPERT.

A technique has been developed that uses co-occurrence matrices and structural information derived from small windows (5x5, 7x7 or 9x9) centered around the pixels. These windows are scanned through pixel position of the learning area and for each pixel, nine attributes are extracted in the following 5 steps:

Step 1: The image is input to a reduction module to optimize vertical and horizontal resolutions.

Step 2: A learning area is chosen, and two square windows are scanned through all pixel positions in this area. (Figure 2).

Step 3: A large window (usually 20x20 pixels) is used to compute four co-occurrence matrices. Each matrix is computed for one of four vector directions (Figure 3).For each matrix, Correlation, Entropy, Contrast, and Angular Second Momentum are calculated.



abcd - area used to compute 4 co-occurance matrices
w - small window used to derive local attributes

Figure 2. Two windows for attribute derivation.

Step 4: A decision is made as to which of four sets of parameters (for which of four directions) derived from co-occurrence matrices are used to enter a set of attributes. The correlation parameter COR is used (a measure of gray-tone linear-dependencies in the image) to make this decision, and the direction with the highest COR is chosen. The Angular Second Momentum (a measure of homogeneity of the image) enters the set of eight attributes as the additional ninth attribute. (A large number of experiments performed on Sun 3 workstations using different texture images proved that the Angular Second Moment was the most useful feature). If no direction is found in the image the ASM with the highest value is chosen. This attribute is scaled into 50 different levels.

Step 5: A small window is chosen according to the direction of texture found in the step 4. The small window cut-out 8 gray level pixel values from pixels around the pixel for which this window is used. The local attributes derived from this window correlate to the attribute derived from the larger (20 x20) area obtained from co-occurrence matrix. This is accomplished by enhancing the pixel gray levels in the "best" vector direction found from co-occurrence matrices.

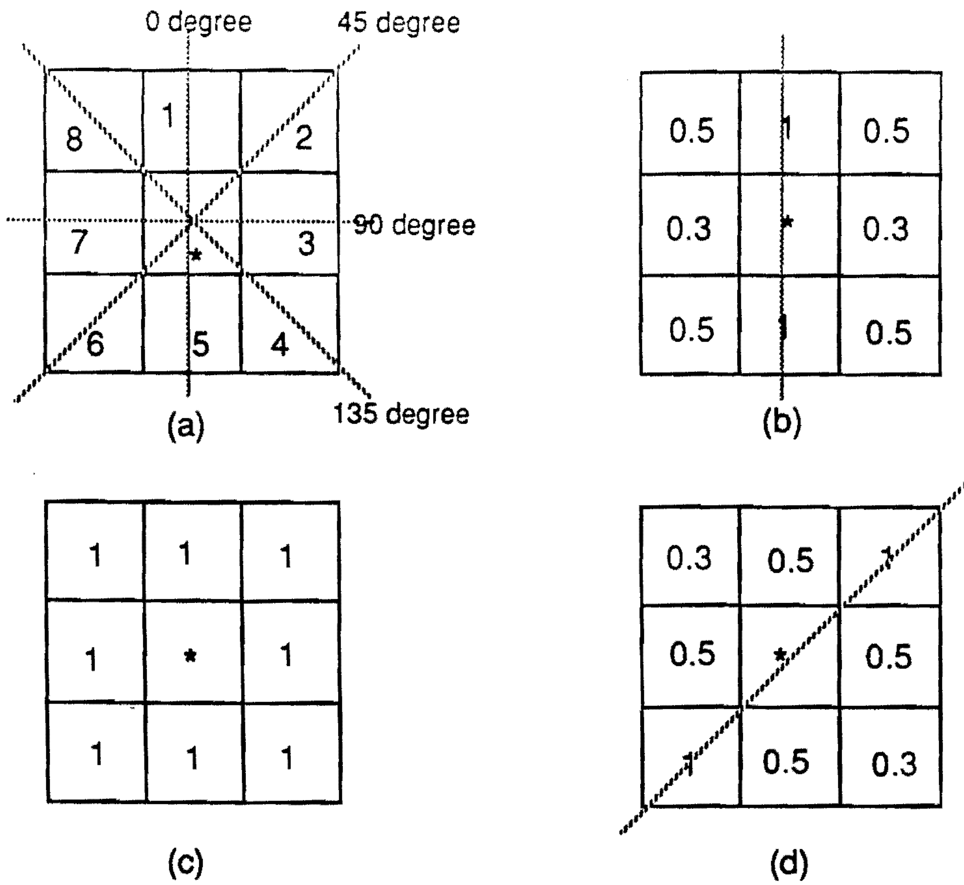


Figure 3. Small windows for local attribute (neighboring gray-level values) extraction.

(a) pixels 1 and 5 are 0° (vertical) nearest neighbors at the distance *d* to the pixel * ; pixels 2 and 6 are 45° nearest neighbors; pixels 4 and 8 are 135° nearest neighbors. (Note the distance *d* depends on the window size, for the 5 by 5, 7 by 7, and 9 by 9 windows the *d* is 2,3, and 4, respectively. (b) the 0° window is used to extract eight attributes for the pixel *, if the directionality of texture is calculated (using co-occurrence matrix parameters derived from the larger window) to be the nearest to 0° Each pixel's gray level value is derived with different weights (1, 0.5, 0.3). (c) If there is no direction found, this window is applied. (d) the 45° window.

The following is the beginning of the input file for the learning module (x9 is the co-occurrence matrix attribute).

```
parameters
run  echo  maxstar  wtc  trim  mode  ambig
 1    pv     10          cpx   gen   dc      neg

 variables
```

#	type	levels	cost	name
1	lin	10	1.000	x1
2	lin	10	1.000	x2
3	lin	10	1.000	x3
4	lin	10	1.000	x4
5	lin	10	1.000	x5
6	lin	10	1.000	x6
7	lin	10	1.000	x7
8	lin	10	1.000	x8
9	lin	50	1.000	x9

l0-events

x1	x2	x3	x4	x5	x6	x7	x8	x9
1	4	8	5	8	4	4	8	12
1	8	6	7	4	6	3	8	12
4	7	4	5	6	6	3	2	11
3	6	5	5	4	3	2	8	12
6	6	5	7	4	9	6	5	13
5	4	3	6	8	4	3	5	14
9	7	5	0	5	9	4	4	15

The following is an example of the output from the learning module. This is already the truncated description.

parameters

test	misclass	tau	andtype	ortype	threshold	norm	cc	dweight	dropa2
yes	false	0.02	average	maximum	0.50	no	no	0.50	1.00

domaintypes

name	type	levels	cost

variables

#	type	levels	cost	name
1	lin	10	1.000	x1
2	lin	10	1.000	x2
3	lin	10	1.000	x3
4	lin	10	1.000	x4
5	lin	10	1.000	x5
6	lin	10	1.000	x6
7	lin	10	1.000	x7
8	lin	10	1.000	x8
9	lin	50	1.000	x9

l0-outhypo

```
1    [x1=5..9]  [x3=6..9]  [x9=0..15]
2    [x4=6..9]  [x7=6..9]  [x9=0..16]
3    [x2=4..9]  [x4=4..9]  [x6=0..5]  [x9=0..15]
4    [x3=5..9]  [x5=0..6]  [x6=0..6]  [x9=0..16]
5    [x3=5..9]  [x7=7..9]  [x9=0..15]
6    [x1=3..9]  [x4=0..6]  [x6=0..5]  [x7=0..4]  [x9=0..16]
7    [x4=0..4]  [x9=0..14]
8    [x1=7..9]  [x2=5..9]  [x5=0..7]  [x6=0..7]  [x9=0..16]
9    [x6=0..6]  [x8=6..7]  [x9=0..15]
```

```
10    [x1=6..9] [x6=7..9] [x7=7..9] [x9=0..15]
11    [x1=8..9] [x7=0..6] [x8=5..9] [x9=0..17]
12    [x3=0..3] [x4=0..3] [x9=0..15]
13    [x2=0..5] [x5=7..9] [x9=0..15]
14    [x2=0..6] [x3=0..4] [x5=7..9] [x9=0..16]
15    [x1=0..6] [x2=0..4] [x6=7..9] [x9=0..16]
16    [x3=0..3] [x5=6..9] [x9=0..16]
17    [x3=0..3] [x8=7..9]
```

```
l1-outhypo
1     [x4=0..8] [x9=17..49]
2     [x1=5..9] [x3=0..8] [x6=5..9] [x8=0..7] [x9=16..49]
3     [x1=0..5] [x3=3..5] [x5=0..6] [x6=4..6] [x9=16..49]
4     [x1=0..4] [x3=0..8] [x4=4..9] [x6=5..6] [x7=4..6] [x8=4..5]     [x9=15..49]
5     [x1=4] [x3=4..8] [x5=0..7] [x6=5..7] [x7=0..7] [x8=0..5] [x9=15..49]
6     [x1=5..9] [x3=0..4] [x4=4..9] [x6=7..9] [x8=0..7] [x9=15..49]
7     [x1=0..6] [x2=5..9] [x3=0..5] [x4=0..6] [x6=7] [x7=5..6]
8     [x1=6] [x3=4] [x6=6]
9     [x1=4..9] [x3=5] [x6=8] [x7=6]
```

The attribute x9 (from the co-occurrence matrix) is present in all complexes but in 17 of the class 0 description (l0-outhypo), it is the last selector of a given complex. All but 8 and 9 complexes include a selector with x9 in the class 1 description (l1-outhypo).

6 RESULTS OF EXPERIMENTS

The results obtained for eight different images using a small 5x5 window (with eight attributes derived from neighboring gray-level values) and 25 by 25 larger window (with one attribute derived from a co-occurrence matrix by scaling the Second Angular Momentum parameter into 50 levels) are presented in Table 1. Textures that have been used in experiments are depicted in Figure 4.

The following textures from the Brodatz Album of Textures have been used:

1. Woolen cloth
2. Fur
3. Pigskin
4. Water
5. Pressed cork
6. Grass lawn
7. European marble
8. Japanese rice paper

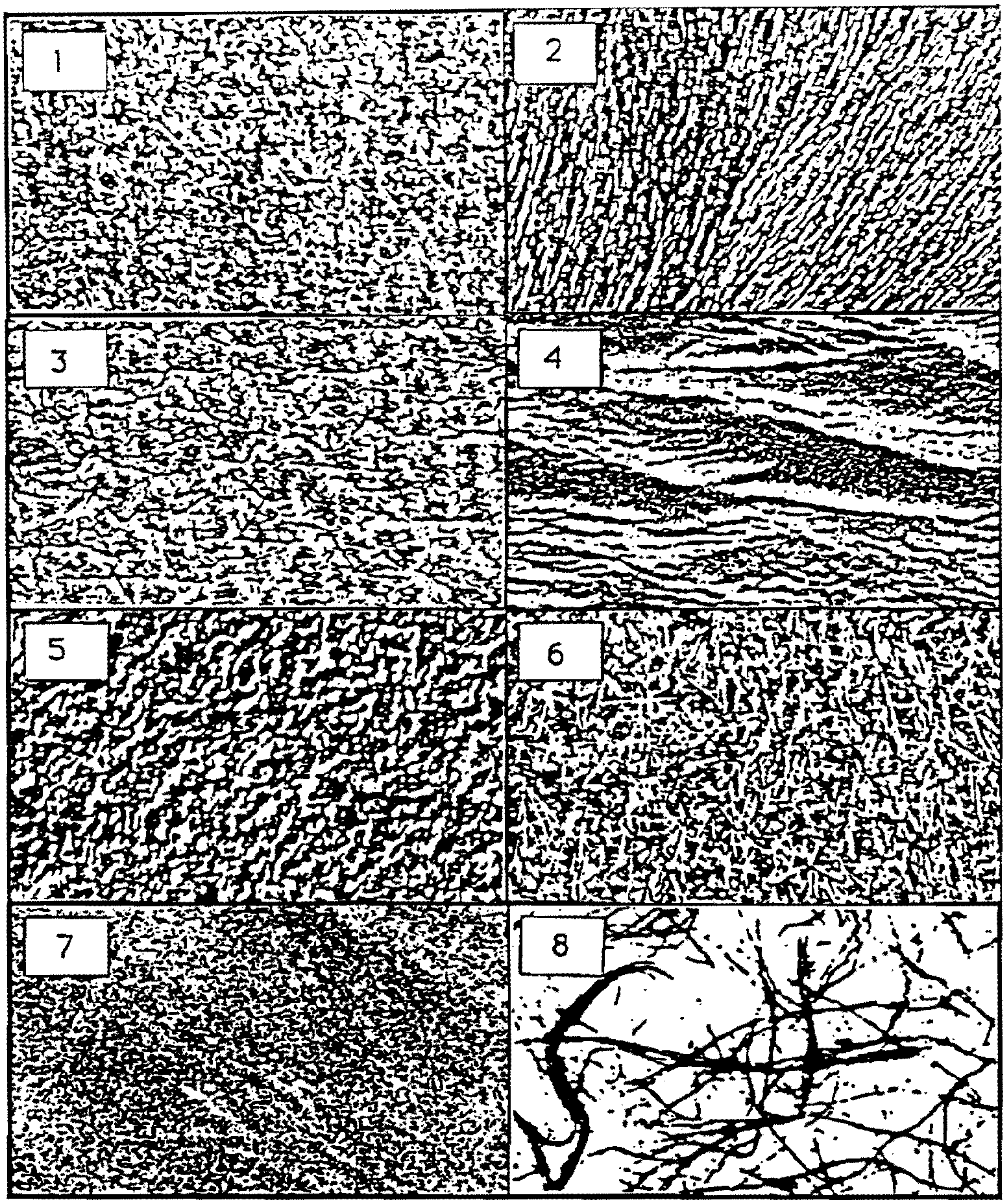


Figure 4. Textures used in experiments.

All images have been acquired as the 512 by 512 by 8 bits image files on the Perceptic system. The learning area occupied 120 by 120 pixels of the 512 by 512 image plane.

		ASSIGNED CATEGORY							
		1	2	3	4	5	6	7	8
TRUE CATEGORY	1	83	0	5	0	6	3	3	0
	2	0	77	0	20	0	0	2	1
	3	3	0	69	0	5	23	0	0
	4	0	0	0	98	0	0	2	0
	5	33	0	10	0	54	3	0	0
	6	1	0	7	0	15	77	0	0
	7	0	25	6	5	0	0	64	0
	8	1	22	0	6	1	2	0	68

Table 1. Classification results-confusion matrix.

7 CONCLUSION

We have presented a system for generating descriptions of natural textures which can be used to test the effectiveness of machine learning in a textural vision domain. A combination of structural and statistic features was used to tune the image and learning algorithm to produce acceptable rules. The experiments described in the previous chapter show the capabilities and effectiveness of inductive learning techniques in a low-level vision domain. Reasonable accurate rules were able to be learned. Effective rules were generated using only a small percentage of the actual pixels in the image.

Although test results are promising, they are by no means conclusive (because of the small number of textures tested). There is also a question of efficiency for real-time applications. Currently, testing is done with acceptable efficiency, but learning rules used in

APPENDIX

AQ15 ALGORITHM

In generating the description of a texture class we have been using Michalski's AQ inductive learning module [13] to incrementally generate map descriptions. The AQ algorithm is essential to our method of texture classification. This appendix gives an overview of the AQ algorithm.

The AQ15 program is based on the AQ algorithm, which generates decision rules from a set of examples. When building a decision rule, AQ performs a heuristic search through a space of logical expressions to determine those that account for all positive examples and no negative ones. Because there are usually many such complete and consistent expressions, the goal of AQ is to find the most preferred one, according to flexible extra-logical criteria.

Learning examples are given in the form of events, which are vectors of attribute values. Attributes may be of three types: nominal, linear or structured (hierarchical). Events represent different decision classes or, more generally, concepts. Events from a given class are considered its positive examples, and all other events are considered its negative examples. For each class a decision rule is produced that covers all positive examples and no negative ones. Rules are represented in VL1 (Variable-valued Logic system 1). VL1 is a multiple-valued logical attributional calculus with typed variables. These multi-valued variables are expressed by using selectors which are two-valued functions. Examples of selectors are:

[x7=2,5,6]
[weather_type=cloudy or rain]

Conjunctions of selectors form complexes. An example of a complex is:

[x3=2,3,5][x1=3,7]

Complexes are assembled into covers. A cover is a disjunction of complexes describing all positive examples and none of the negative examples of the concept. A cover is formed for each decision class separately. It defines the condition part of a corresponding decision rule. The following are two examples of decision rules:

```
[transport=car]    <=    [weather_type=cloudy or rain] or [Temp=40..60]
[transport=bike]   <=    [weather_type=sun][Temp>60]
```

The major idea behind the covering algorithm is to generate a cover in steps, each step producing one conjunctive term (complex) of the cover.

A condition is represented by a VL1 [Michalski, 1975] complex. A VL1 complex is a conjunction of relational statements called selectors. A selector, representing a logical statement concerning the value range of an attribute, is of the following form:

[attribute=lower-range..upper-range]

where both lower-range and upper-range are integer numbers. For example, the VL1 complex [x=10..13][y=1..5][d=2] indicates that the value of the attribute x falls inclusively in the range between 10 and 13, the value of y falls inclusively in the range between 1 and 5, and the d attribute represents direction value 2.

REFERENCES

[1] Brodatz, P., "*A Photographic Album for Arts and Design*", Toronto, Dover Publishing Co., 1966.

[2] Channic, T., "*Texpert : An Application of Machine Learning to Texture Recognition*", A publication of the Machine Learning and Inference Laboratory; MLI 89-17, George Mason University, Fairfax, Virginia.

[3] Chen, P. and Pavlidis, T, "*Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm*", Tech. Rep. 237, Princeton University, 1987.

[4] Conners, R. W. and Harlow, C.A., "*Towards a structural texture analyst based on statistical methods*", IEEE Trans. Patt. Analy. Mach. Int., PAMI-2, 3, 1980, pp 204-222.

[5] Davis, L., Johns, S., and Aggarwal, J., "*Texture analysis using generalized co-occurrence matrices*", IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1, 1979,pp 251-259.

[6] Lu, S.Y. and Fu, K.S., "*A syntactic approach to texture analysis*", Comput. Graphics Image Process. 7, 1978, pp 303-330.

[7] Gagalowicz, A., "*Stochastic textures fields synthesis from a priori given second order statistics*", Proc. Pattern Recognition and Image Processing Conference, Chicago, Il., Aug. 6-8, 1978, pp 376-381.

[8] Haralick, R. M., K. Shanumugam, and I. Dinstein, "*Textural features for image classification*", IEEE Trans. Syst., Man, Cybern., vol. SMC-3, po 610-621,, Nov. 1973.

[9] Hawkins, J. K., "*Textural properties for pattern recognition*", in Picture processing and psychopictorics, 128-136, Academic Press, New York, 1970.

[10] Julesz, B., "*Visual pattern discrimination*", IRE Tran. Information Theory, Vol. IT-8, No. 1, Feb. 1962, pp.84-92.

[11] Julesz, B., "*Experiments in a visual perception of texture*", Sci. Amer., vol. 232, Apr. 1975, pp 34-43.

[12] Michalski R. S. "*AQVAL/1--Computer Implementation of a Variable-Valued Logic System VL1 and Examples of Its Application to Pattern Recognition*", First International Joint Conference on Pattern Recognition, October 30, 1973, Washington D.C.

[13] Michalski, R. S., "*A Theory and Methodology of Inductive Learning*", in Machine Learning: An Artificial Intelligence Approach, TIOGA Publishing, Palo Alto, CA, pp 83-134.

[14] Michalski, R.S., Mozetic I., Hong J.R., Lavrac N., "*The AQ15 Inductive Learning System*", Report No. UIUCDCS-R-86-1260, Department of Computer Science, University of Illinois at Urbane-Champaign, July, 1986.

[15] Pachowicz, W.P., "*Low-level Numerical and Inductive Learning Methodology in Texture Recognition*", IEEE International Workshop on Tools for AI, Washington, D.C. October 1989.

[16] Reinke R. E., "*Knowledge Acquisition and Refinement Tools for the Advice Meta-Expert System*", ISG 84-4, Department of Computer Science, University of Illinois at Urbana-Champaign, July 1984.

[17] Rosenfeld, A., and Lipkin, B.S. "*Texture analysis*", in Picture processing and psychopictorics, Academic Press, 1970, pp 300-322.

[18] Tamura H., S. Mori and T. Yamawaki, "*Textural features corresponding to visual perception*", IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-8, No.6, June 1987, pp 46-473.

[19] Tomita F., Y. Shirai and S. Tsuji, "*Classification of textures by a structural analysis*", Proc. 4th Int. Joint Conference on Pattern Recognition, Kyoto, Japan, Nov. 1978, pp 556-558.