Applications of Learning and Planning Methods
edited by N. G. Bourbakis

91-40

CHAPTER 5

THE APPLICATION OF SYMBOLIC INDUCTIVE LEARNING TO THE ACQUISITION AND RECOGNITION OF NOISY TEXTURE CONCEPTS

Peter W. Pachowicz

Center for Artificial Intelligence, George Mason University
4400 University Drive, Fairfax, VA 22030
ppach@gmuvax2.gmu.edu

ABSTRACT

The article presents and tests a methodology developed to acquire and recognize noisy texture concepts. The introduced methodology distinguishes the following three complementary research issues that are finally integrated within a system. The first issue deals with the acquisition of a texture concept incorporating symbolic inductive learning from provided examples. These examples consist of low-level numerical attribute values extracted from texture images incorporating two well-known methods, Laws' masks and co-occurrence matrices. The learned description of a texture class is a rule where a conditional part is a disjunctive normal form. The second issue deals with the manipulation of acquired concept descriptions in order to remove their noisy components. This manipulation optimizes the class description truncating its less significant components and preserving the most typical components of the concept. We show that the application of such optimized descriptions to the recognition of test datasets increases the system recognition effectiveness. The third issue deals with the application of concept optmization performed within a closed-loop system. This closed-loop system integrates the concept optimization, the recognition, control, and final decision making, and it works for gradually increasing optimization degree. This iterative optimization builds the characteristics of the recognition curves taken versus increasing optimization degree. The system makes decisions based on the dynamic pattern rather than static performance of such recognition curves. The proposed methodologies are tested in three separate experiments the results of which are presented in this paper.

1. Introduction and Motivation

One of the features of intelligent systems is their ability to acquire a concept of object such as a description of its characteristics or properties and to apply this concept to recognize a particular object among many other objects. Since our environment contains textured objects and the texture data contains noise and other irregularities, the acquisition and robust recognition of such noisy and imperfect concepts is crucial for many vision tasks; e.g., inspection, object recognition, object localization for navigation, remote tracking of a moving object, and intelligent surveillance. The success of these tasks depends both on the effectiveness of image understanding processes and the quality of data which a system has to deal with during both the learning phase and the recognition phase.

Once the system is trained and tuned to a given type of data, it is difficult to assume without testing that the system will be working with the same data but effected by a different noise. Moreover, a given object can be perceived differently when perceptual conditions are changing. Variability of object characteristics (particularly a texture) requires the development of capabilities that will allow the system to reconfigurate, tune, and update its model (knowledge) regarding object or sensor movement or the variation of sensing conditions (e.g., changing resolution, lighting, overlapped noise).

This paper presents our approach and methodology (along with tested results) to the following three complementary parts of the texture recognition problem:

1. the application of inductive learning methodology to the acquisition of texture concept from low-level numeric texture features,

2. the manipulation of acquired concepts in order to optimize them and reduce some of the noisy components from a concept description,

3. the recognition of texture datasets through flexible matching and newly developed multiple-matching methodology.[1]

Traditional adaptability and learning in engineering systems apply control engineering[23] and/or pattern recognition[10] techniques. These approaches work well using quantitative information. But typical processes of high-level vision are based on symbolic computation. The border line between numeric and symbolic computation has not been established. There are methods, which successfully apply the iconic approach alone to invariant recognition and data fusion.[33] However, the application of such methods to the object inspection task is difficult. For example, the inspection task uses object models or object variances that are acquired through explanation, technical diagrams, and tolerances, rather than from presentation of various real objects during a teaching phase.

Our approach to computer vision incorporates machine learning methodologies applied to object location, recognition, inspection, and scene understanding tasks, and it is based on the creation of system adaptability functions.[27] The learning capabilities of the system use context dependence to adapt to the dynamic environment. In the proposed approach, models used to represent data and system knowledge must possess a dynamic structure that can be modified by the system itself based on its own experience. Incorporating machine learning and cognitive processes such as induction, deduction, discovery, and analogy, we plan to combine elements of visual recognition and visual imagery[28] within a unified system.

Currently, we are investigating vision adaptation mechanisms that allow a system to recognize objects in new external situations. Such situations include 3-D rotation, changed resolution, quantitative and qualitative noise, changing lighting conditions, and mutual positional relations between objects such as occlusion. The system has to perform the following vision tasks: object localization, recognition, and inspection. We assume that prior to the system work, it either possesses an initial model of objects, or it acquires the optimal model from the environment based on background meta-knowledge. The adaptive

system must be able to manage its internal structure and data (dynamic memory) in increase/decrease the number of distinctive features, add new objects, and modify flow through the network. One of the most challenging issues that is currently investigation is learning object description from variable object characteristics problem attacks an issue of texture model evolution performed in time (i.e., for a se of texture images) and for moving objects. Model evolution through machine le technique incorporates an incremental mode of learning that is typical for a human be

A typical task that depends on the system adaptation capability is texture-based recognition for robot navigation in a natural terrain. The difficulties of robust recogn 3-D objects is caused by a great variability of images projected onto the sensory arra variability is caused not only by the 3-D rotation of an object, but also by chan resolution, positioning of light sources (e.g., light reflection, shadows), mutual po relations with other objects at the scene (e.g., touching, occlusion), etc. In the trad approach to object recognition, the system is associated with a set of characterist features, which are used for recognizing particular objects. Such a set of feat constant[4] in that the perceptual system is unable to verify the usefulness of the object Thus, the classification is correct as far as the accuracy of the off-line created reco model is concerned.

Recent progress in visual recognition of objects has generally been based on the ext of traditional visual recognition processes. Marr's[17] 2.5-D sketch theory intr directional information to object shapes reconstructed from contour, color, texture, motion, etc. Later, this data is integrated into a perceptual system. Currently, this c is being applied to the creation of the MIT Vision Machine.[29] The recognition of an and its position, which is insensitive to 3-D rotation, has been proposed by Ikeuc Kanade.[13] In the teaching phase, the system creates internal multi-view representations based on the CAD model. The recognition phase matches currently features of an object (where features can possess a 2.5-D description) with pre features of the object's model. In these approaches, the recognition systems have created without adaptive capabilities. One can say that these systems are closed an unable to verify and modify the object model. These typical limitations of object recog systems have been discussed by Bhanu[3] within the context of automated target recog As a consequence, a new approach has been suggested that integrates machine learnin target recognition in order to create an adaptive target recognition system.[2] Th approach integrates two machine learning techniques with a knowledge-based rea system. The first technique, explanation-based learning, provides the ability to generalized descriptions of object classes. The second technique, conceptual clust allows the system to make decisions based on simple symbolic descriptions rather numerical similarity measures. When created, the system will be capable of modifyi internal data concerning objects on the basis of its experience.

Searching the area of integrating vision and symbolic machine learning for te recognition task, we conclude that such an approach has not been explored before. It however, initiated and illustrated by a simple example by Michalski.[19] Most wor adaptive vision systems are limited to the improvement of image segmentation[12], an application of expert systems to the automatic configuration of vision systems.[15,18,24] of this effort, however, is conducted in the area of shape recognition and relatively effort is paid to the adaptability of texture recognition system dealing with great variabil texture occurrence. Researchers try generally to avoid the problem of texture variabili the application of more powerful methods for the acquisition of texture characteristics; e.g., Markov random fields[6], Gibbs random fields[8], and the spatial frequency spectrum.[16,34]

On the other hand, the traditional object recognition applies classifiers in such a way that they are adapted to the feature set to take advantage of the extracted information. Such an approach belongs to a class of feature extraction oriented methods, where an extraction of relevant features plays a very important role.[9] The main problem with traditional approach to the recognition task is that we do not have a universal feature extraction method that works effectively with noisy and imperfect data. When one considers that the training data can be noisy and imperfect, one has to agree that the derived descriptions of texture classes consequently contain noisy components. Therefore, the recognition based on the direct matching of such noisy and imperfect descriptions with testing data follows all above mentioned disadvantages; i.e., the results are affected by the noise and imperfectness of learning and testing data. This is one of the reasons why most efforts to the solution of the recognition problem focus on the improvement of the feature extraction phase to reduce the influence of noise and data imperfectness on the system performance. Contrary to the traditional approach, we modify concept descriptions to reduce hypothetical noisy componets and apply such description to the recognition phase.

This paper presents methodology of the integration of machine vision with machine learning in the area of texture recognition and it summarizes our first introductory experiments. Section 2 of this paper presents the preparation of training and testing data along with their conversion from numeric to symbolic representations. Section 3 overviews applied methodology to acquire concept descriptions, while the principles of matching such concepts with a testing dataset are presented in Section 4. The comparison of proposed machine learning approach and traditional pattern recognition approach to the problem of texture recognition is demonstrated in Section 5. Section 6 presents the idea of optimization of concept descriptions and its influence on the recognition performance. The integration of such optimization processes with the recognition processes performed iteratively within a closed-loop system architecture is presented in Section 7. Finally, Section 8 summarizes our work and presents problems for the future research.

2. Selection of Learning and Testing Data

This section presents the process of extracting texture attributes and examples (where a single example is a vector of attributes) obtained from input texture images. We begin with the presentation of experimental images of texture classes used in three experiments discussed in this article. Next, the process of extracting numeric attributes and their conversion to their symbolic representation (suitable for the applied machine learning methodology) is presented.

2.1. Preparation of Texture Images

All texture images have been obtained from Brodatz album[5] and they have been coded as black-and-white images of 512x512 pixels. Each pixel was coded onto 256 grey levels. Image data was grouped for three experiments.

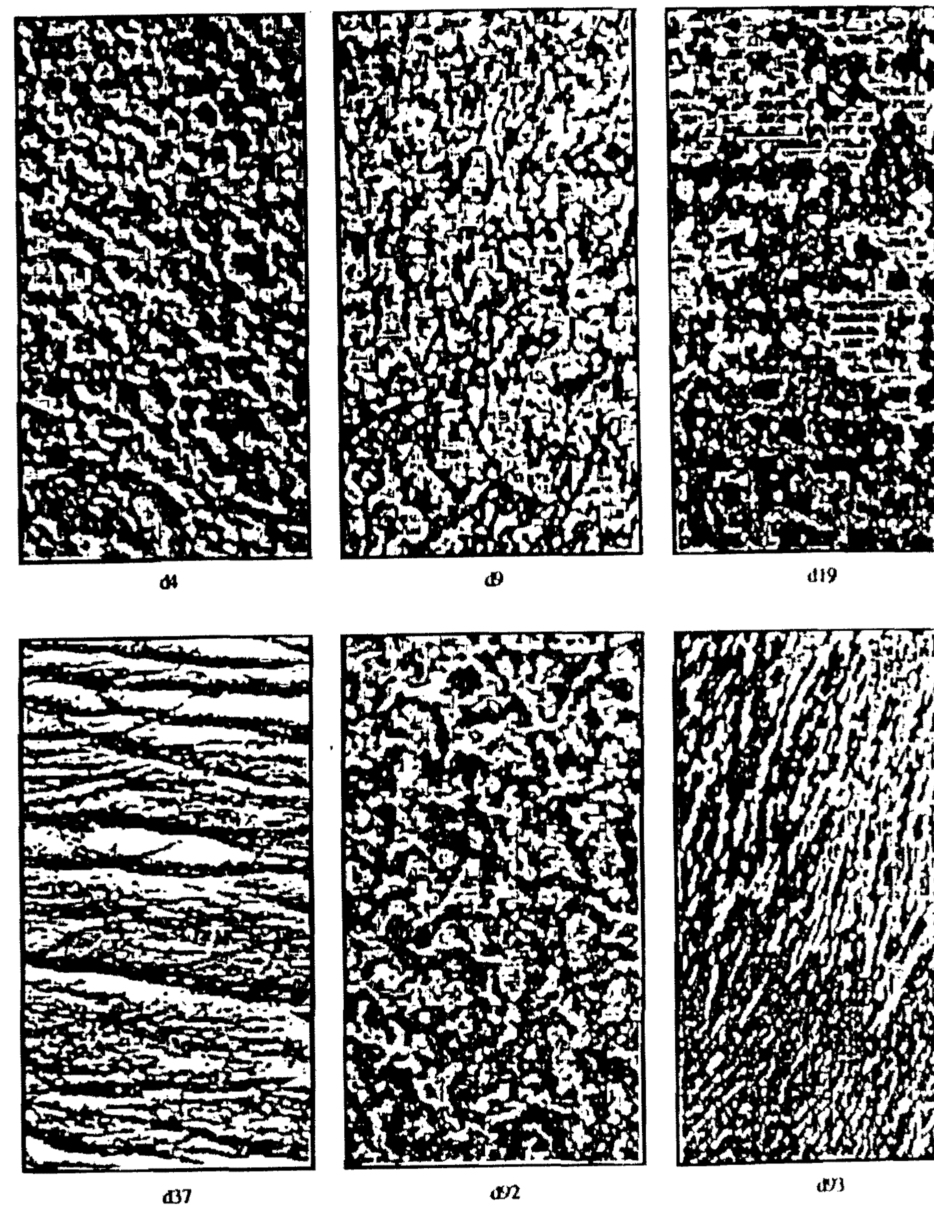

Fig. 1a Training sections of the fist set of texture images taken from the Brodatz album: pressed cork (d4) lawn (d9), woolen cloth (d19), water (d37), pigskin (d92), and fur (d93)

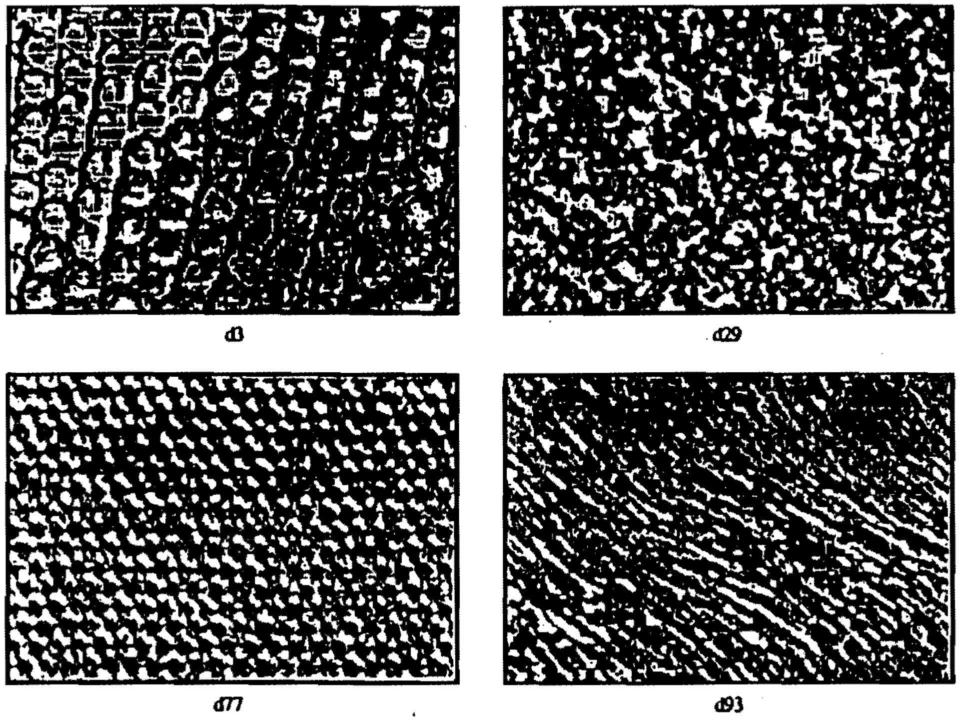

Fig. 1b Training sections of the second set of texture images taken from the Brodatz album: reptile skin (d3), beach sand (d29), cotton canvas (d77), and fur (d93)

The first set of images has been provided for the first and the second experiment. It contains the following six classes of texture (Figure 1a illustrates training sections of texture images): pressed cork (d4), lawn (d9), woolen cloth (d19), water (d37), pigskin (d92) and fur (d93). They were non-uniformly illuminated and two of them (lawn and water) have smoothly changed texture resolution caused by the angled projection of a texture surface. Next, the intensities of all images were decreased to 64 gray-levels. This limitation was imposed by the application of texture-feature detection by co-occurrence matrices (i.e., a large number of image gray-levels causes the dimension of the co-occurrence matrix to be very large). These input images were used to create two files of images. The left-hand side of the input images (sub-images of 512x256 pixels) were used in the learning phase, while the right-hand side (sub-images of 512x256 pixels) were used as data for the recognition phase.

The second separate set of images was provided for the third experiment described in this paper. This set contains the following four classes of texture (Figure 1b illustrates training sections of texture images): reptile skin (d3), beach sand (d29), cotton canvas (d77), and fur (d93). A set of relatively small sections of these images was shown to the system during

the learning phase, while the other set of image sections was shown during the recognition phase. Before the feature extraction was executed (for the learning and recognition phases) all textural images were preprocessed. The number of pixel grey-levels was changed from 256 to a number of levels below 55.

2.2. *Extraction of Texture Numeric Attributes*

The extraction of attributes is the first phase of the transformation of raw image into its more specific characteristics. The extraction of texture numeric attributes has been executed for each image of the first group of six texture images presented in Figure 1a. A single attribute determines local specific characteristics of a given number of pixels. For a given pixel, we can extract a certain number of different attributes that form an event such as a vector of these attributes.

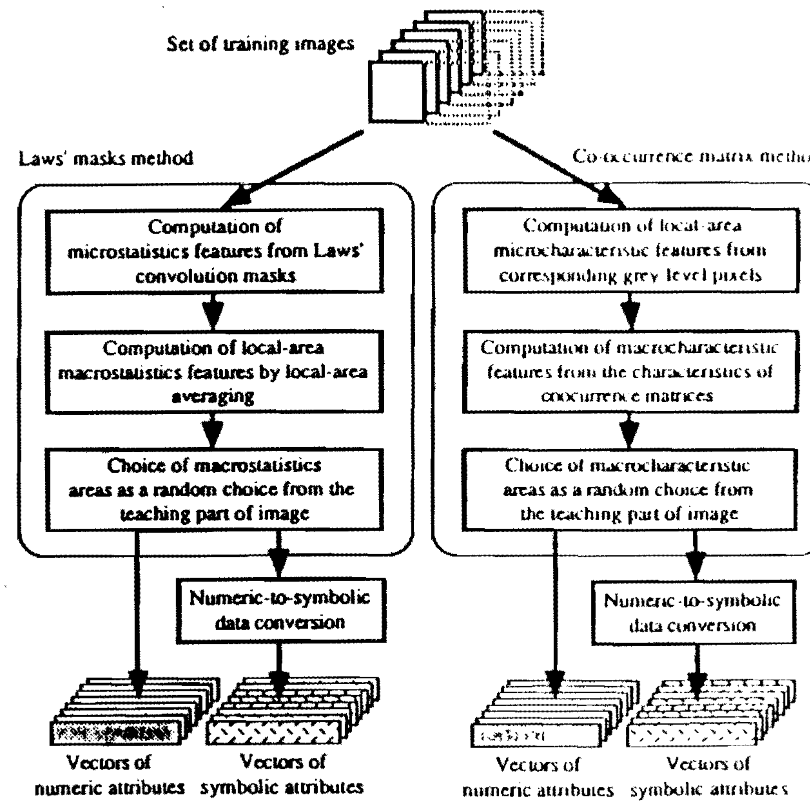

Fig. 2 Extraction of texture attributes

We applied two methods to extract local numeric attributes. The first method filters an input image incorporating Laws' energy masks[14], while the second method computes the characteristics of co-occurrence matrices extracted for a given texture area.[35] The following steps of the extraction of texture numeric attributes are presented in Figure 2.

The extraction of texture numeric attributes for the Laws' energy masks method was performed in the following three steps:

1. The computation of a single vector of microstatistical features was executed for a central pixel using one 3x3 and seven 5x5 convolution masks shown in Figure 3.

2. Based on the vectors of microstatistical features, the system computed local-area macrostatistics over a larger window of 20x20 pixels, where a vector of macrostatistical properties was computed as the average absolute value of each pixel feature over the window.

3. The third step provided a random choice of macrostatistics over the teaching image of 512x256 pixels. A set of 200 random macrostatistical vectors was computed as learning samples for each class of texture.

4	-6	4
-6	8	-6
4	-6	4

-1	-4	-6	-4	-1
-2	-8	-12	-8	-2
0	0	0	0	0
2	8	12	8	2
1	4	6	4	1

-1	0	2	0	-1
-2	0	4	0	-2
0	0	0	0	0
2	0	-4	0	2
1	0	-2	0	1

-1	0	2	0	-1
-4	0	8	0	-4
-6	0	12	0	-6
-4	0	8	0	-4
-1	0	2	0	-1

1	-4	6	-4	1
-4	16	-24	16	-4
6	-24	36	-24	6
-4	16	-24	16	-4
1	-4	6	-4	1

-1	-2	0	2	1
-4	-8	0	8	4
-6	-12	0	12	6
-4	-8	0	8	4
-1	-2	0	2	1

-1	-2	0	2	1
0	0	0	0	0
2	4	0	-4	-2
0	0	0	0	0
-1	-2	0	2	1

-1	-4	-6	-4	-1
0	0	0	0	0
2	8	12	8	2
0	0	0	0	0
-1	-4	-6	-4	-1

Fig.3 Laws' masks

The second method, the co-occurrence matrix method, was applied to build a characteristics of texture classes in the similar way with the distinction of the following three steps:

1. A vector of four microcharacteristic features was extracted, in the form of gray values of pixels located within a distance d, and direction ß, from the central pixel (see Figure 4), and associated with the gray value of the corresponding central pixel.

2. For each microcharacteristic feature and a class of texture, the system computed local-area diagonal co-occurrence matrices over a larger window (20x20 pixels). Each co-occurrence matrix was then used to calculate both the angular second momentum (also called the matrix uniformity) and the contrast as a measure of the spread of values away from the main diagonal.[31] Finally, we obtained a vector of numeric parameters, where each position within the vector represents local-area macrocharacteristics for a given distance d, and direction ß.

3. This step of the extraction of texture attributes was the same as for the Laws' masks method and it provided a random choice of macrostatistic vectors over the teaching image, where a set of 200 learning examples was selected randomly for each class of texture.

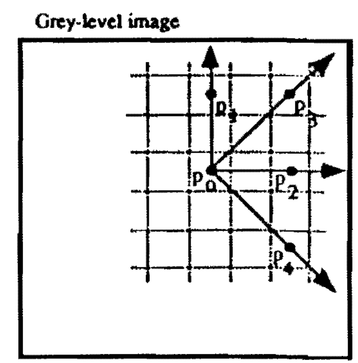

Fig.4 Local microcharacteristic features for a given direction and distance

2.3. Conversion of Numeric Attributes into Their Symbolic Representation

Symbolic inductive learning, as applied to acquire texture rule descriptions, requires to cod the data as symbolic values. To fulfill this requirement, we converted numerical attribute into their symbolic intervals incorporating a special interface. As a consequence, th conversion called as scaling process is an early generalization of examples of numeric for into their more general symbolic representation as an elementary cubic-cell in the symboli attribute space. This conversion is applied in case of a constant number of texture classe as well as the constant number of attributes. The scaling was determined by an a-prio given set of events, which cannot be changed or extended by adding data from th environment, such as that obtained during on-line system experience. On the other hand future use of dynamic scaling assumes that a system will be able to extend the number o texture classes, to add new events that are characteristic of a single class, and to change th number of attributes or modify them.

Let us assume that $V = \{ V^1, \ldots, V^j, \ldots, V^k \}$ is a set of numerical events for a singl class of texture, and $V \ni v = (v_1, \ldots, v_i, \ldots, v_m)$ is a learning event expressed as vector of m numerical attributes. Then, for each i-th attribute, we compute the followin scaling parameters: $vmin_i$, $vmax_i$, and Δ_i, which are the minimum value, maximum value, and data interval of each i-th attribute respectively. These parameters were found for all elements of the set V:

$$vmin_i = \min_{j \in [1,k]} (v_i \in V^j) \qquad vmax = \max_{j \in [1,k]} (v_i \in V^j) \qquad (1)$$

$$\Delta_i = (vmax_i - vmin_i)/(nint - 1) \qquad (2)$$

where nint is the number of intervals accepted by an inductive learning algorithm (in our case nint was equal to 50).

We applied the conversion of numeric attributes to their symbolic representation by the computation of a new attribute vector $u = (u_1 , \dots , u_i , \dots , u_m)$ with the following formula:

$$\text{if} \quad v_i \in [vmin_i + (n-1)^*\Delta_i ;\ vmin_i + n^*\Delta_i) \qquad \text{then} \quad u_i = n \qquad (3)$$

The system checked the consistency of the created vectors of attributes grouped into sets of events corresponding to different texture classes. Because these sets were consistent (i.e., each vector of attributes was unique for a given class) the system did not apply the scaling processes on the lower level (i.e., on the level of inconsistent data).

3. Acquisition of Rule Descriptions of Texture Classes

The inductive incremental learning program AQ14[11,21] was applied to learn the texture attributional descriptions from examples. The AQ program performs a heuristic search through a space of symbolic expressions, and its goal is to find the most preferred expression according to a specified criterion. The input to the AQ program consists of a string of learning events, and each event is a vector of attribute values. The set of events obtained for one class is called a set of positive examples. With respect to this particular class, all other events are negative examples. The program finds an optimal cover of all the positive examples. This cover cannot include any negative examples. The process repeated for each class of learning events produces decision rules to discriminate all classes of texture images amongst themselves.

The conditional part of a rule is defined as a cover and coded as a disjunctive normal form. A cover is a disjunction of complexes (using the OR operator), where a complex is decomposed into selectors (using the AND operator). 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: =, <, >. An example of a rule, complex and selectors can then be illustrated as below, however, the domain of texture attributes operates on symbolic values coded as numbers in range from 0 to nint:

rule:	[Transport=car] <== [Weather=bad] v [Temp<60]	
complex:	[Weather=bad] <— [Weather_type=cloudy] & [Temp>60] & [Wind_dir=South v West]	
selector:	[Weather_type=cloudy]	(4)

Each generated complex is associated with a pair of weights; i.e., t-weight and u-weight. 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 following is the example of a complex for texture attribute domain:

[x1=5..10 v 15][x2>34][x6=5..34][x8<45]   (t-weigth: 18, u-weight: 9)   (5)

where x-i is an attribute. The complexes of a rule are then ordered according to decreasing values of the t-weight.

The AQ14 inductive incremental learning program can work in two modes producing intersecting or disjoint covers. Rule induction in the intersecting mode produces covers that can logically intersect with those of other classes over "don't care" areas of the attribute space. On the other hand, rule induction in the disjoint mode produces covers that do not intersect at all with covers of other classes. As a consequence, rules produced in the intersecting mode are more general than rules produced in the disjoint mode.

In our experiments, we learned rule descriptions of texture classes both in intersecting and disjoint modes, mainly to compare recognition results. The output of the AQ14 algorithm consisted of the discrimination rules which were transferred to the texture recognition phase. The program is controlled by a set of parameters and they proper use has the great influence on system recognition effectiveness. Any change of these parameters must be done very carefully.

4. Texture Recognition through Flexible Matching

Acquired rule descriptions of texture classes were then applied to recognize test texture datasets. This test data was prepared in the same way as learning data, but test events were chosen from other parts of images that were not presented to the system during the learning phase. The same methods as in the learning phase were applied to the computation of texture numeric attributes. One hundred testing samples were obtained for each class of texture, and for each separate method used in extracting texture attributes. Extracted test events were converted from their numeric representation to the symbolic form applying scaling process with parameters calculated during the learning phase. The recognition phase was then activated to check the effectiveness of this machine learning approach to texture recognition.

Generally, there are two methods for classifying the concept membership of an instance, i.e., the strict match and the flexible match. In the strict match, one tests whether an instance strictly satisfies the condition part of a rule (i.e., one of rule complex). Such matching gives the response of two possible logical values; i.e., true or false. In the flexible match, one determines the degree of closeness between the instance and the condition part. Such closeness is represented by a coefficient that can vary in range from 0 (does not match) to 1.0 (matches). In strict matching, one determines the most closely related concept description. Flexible matching must be applied when the concept does not have clear boundaries, the learning data was noisy, or the test data was obtained from other parts of the image. If image sections applied to the testing phase are different from sections applied to the learning phase then test data can include variations of concept occurrence caused by local differences in projection, illumination, and noise.

In the recognition phase, we incorporated a special tool, the ATEST program,[30] that had been developed to test the performance of a rule base. ATEST provides the domain expert with three very important capabilities. First, it allows the expert to rapidly test a rule base on numerous test examples under a variety of evaluation schemes. These evaluation facilities provide information about the overall performance of specific rules, and this feature is used in the texture recognition phase. Second, ATEST provides routines that check a rule base for consistence and completeness. Third, it applies flexible matching to determine classification decision.

ATEST views rules as expressions when applying them on a vector of attribute values. The result of this evaluation is a real number which is a degree of consonance between the conditional part of the rule and the event. In our case, the program worked on the separated sets of events, where each set was obtained for a single class of texture as described above. Each event was classified into one of the given texture concepts. There were three possible classification decisions for a single event, i.e., an event belongs only to the correct class (unique-classification), an event does not belong to the correct class (misclassification), and an event belongs to several classes where one of them is the correct class (multiple-classification). The final recognition decision is made based on counting the unique- and multiple-classified events for a single set of test data. Such classification is called *first rank*, and is our main measure of the effectiveness of recognition.

The performance of the ATEST program is controlled by a set of parameters. The user can adapt the program to a specific domain. The control parameters determine program performance in evaluating the rules on the testing events. The evaluating parameters provide definitions for logical operators (e.g., the AND operator may be evaluated as "minimum" or as "average"). They control the degree of conconance and the definition of selectors (e.g., a selector can be treated as boolean *true/false*, or as a function with a normalized value from 0 to 1). Any change of these parameters must be done very carefully because the effectiveness of system recognition can be decreased.

5. Experimental Comparison of Recognition Effectiveness

In this section, we present the performance of applied symbolic machine learning methodology to the problem of texture recognition. Obtained results are compared with the k-NN non-parametric pattern recognition method.

5.1. System Architecture

The traditional open-loop architecture, presented in Figure 5, was used incorporating learning and recognition systems and two recognition approaches (i.e., machine learning and pattern recognition). The system included the extraction of texture numeric attributes, conversion of these attributes to symbolic representation, a learning-from-examples module (AQ14), an inductive assertion module (ATEST program), and a pattern recognition classifier. While most of these modules/tools have been explained in preceding sections, the application of the pattern recognition classifier is discussed below.

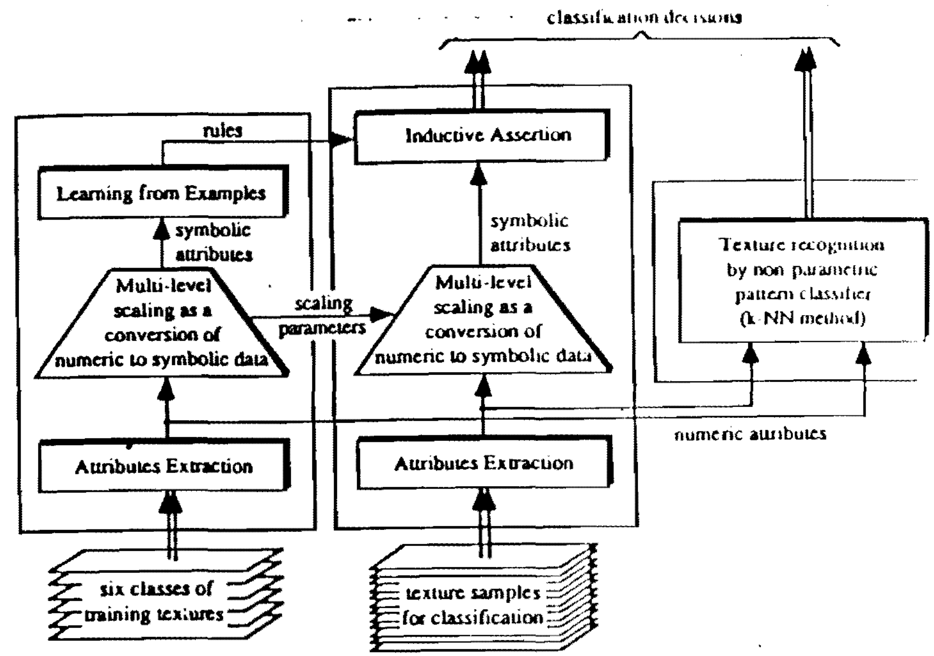

Fig.5 Open-loop architecture for texture concept acquisition and recognition incorporating symbolic machine learning and pattern recognition approaches

5.2. Recognition by Pattern Classifier

Several pattern recognition techniques have been considered for the texture recogniti problem. The parametric methods, e.g., risk minimization using Bayes decision metho were excluded after testing the attribute space. The creation of parametric models of featu distribution was not satisfactory because the distribution of training data was irregular an difficult to estimate from parametric curves. Therefore, we chose the well-known k N non-parametric statistical pattern recognition method.[10]

In this method, learning examples are cumulated into sets corresponding to texture classe During the recognition phase, a subset of k-nearest learning examples is selected from t set of all learning examples presented to the system. Thus, the classification decision created indicating this class, for which most of the k-nearest learning examples we selected. The main advantage of this method is its handling of irregular and compl training data. However, the requirement of storing all learning data (or selected data fro the most representative samples of the attribute domain distribution) for their use during t recognition phase is time and memory consuming. This disadvantage limits the use of su methods. The second problem is related to the choice of the most appropriate k value i our comparison, we found k=10 and k=30 as most representative.

5.3. Results Comparison

The results obtained from the recognition experiments are presented separately for the machine learning approach (Table 1) and for the pattern recognition approach (Table 2).[26] These tables compare results both for the Laws' method and the co-occurrence matrices method applied to extract texture attributes. The comparison includes recognition effectiveness for each class and the complexity of rule description learned by the AQ14 program.

The comparison of results presented in Table 1 and 2 distinguishes the following four most important conclusions:

1. The average recognition effectiveness was the same for machine learning and pattern recognition approaches when texture features were obtained from the co-occurrence matrix method. For the Laws' masks method, the machine learning approach was better than the k-NN pattern recognition method.

2. The worst recognition result was obtained for the pigskin image (d92), however, the machine learning approach recognizes this class with sufficiently high confidence, especially for the Laws' masks approach.

3. Neither Laws' masks method nor the co-occurrence matrix method applied to the extraction of texture attributes was consistent when compared amongst themselves for each class of texture. The Laws' masks method combined with the machine learning approach to the acquisition of texture description was generally worse considering both the number of generated complexes of the rules and the recognition results. But in the case of the fifth class, the number of complexes in the rule was significantly lower and the recognition rate was higher.

4. The average deviation of the recognition, defined as $d = 1/N \sum |x - x_i|$, was lower for the machine learning approach than for the pattern recognition approach. Such lower deviation with high average recognition rate makes the system more stable, i.e., each texture class can be recognized with the same level of confidence despite the similarity to another class.

The recognition results shown in Table 1 were obtained for the intersection cover mode of the inductive learning algorithm. The generation of rules for the intersecting mode was much faster than for the disjoint cover mode. The average recognition effectiveness was also better. The results show that for approximately the same number of complexes generated in the intersecting and disjoint modes, the recognition results were better for the disjoint mode. This tendency is presented in Table 3 for class d4 of texture. On the other hand, the recognition rate for class d37 of texture is also included, to show the tremendous decrease in recognition rate for the disjoint mode, where a large number of complexes was generated.

	Texture description method					
	Laws' masks			Cooccurrence matrices		
	Number of generated complexes	Recognition First rank	Recognition Unique	Number of generated complexes	Recognition First rank	Recognition Unique
d4	38	72 %	37 %	26	88 %	78 %
d9	35	74 %	45 %	31	78 %	57 %
d19	23	81 %	57 %	24	90 %	78 %
d37	8	93 %	91 %	6	99 %	96 %
d92	31	73 %	45 %	46	57 %	35 %
d93	23	87 %	64 %	13	84 %	78 %
Average recognition		80 %	56 %		83 %	70 %
Averaged deviation		7.	14.2		10.	16.2

Table 1 Recognition results for machine learning approach (AQ14) - intersection mode

	Texture description method			
	Laws' masks		Cooccurrence matrices	
	Recognition for k = 10	Recognition for k = 30	Recognition for k = 10	Recognition for k = 30
d4	53 %	49 %	88 %	90 %
d9	73 %	68 %	75 %	77 %
d19	88 %	91 %	95 %	97 %
d37	99 %	97 %	98 %	96 %
d92	40 %	38 %	49 %	46 %
d93	69 %	57 %	97 %	93 %
Average recognition	70 %	66 %	83 %	83 %
Averaged deviation	16.3	18.7	14.4	14.4

Table 2 Recognition results for k-NN pattern recognition approach

	Texture description method							
	Laws' masks method				Co-occurrence matrices method			
	DC mode		IC mode		DC mode		IC mode	
	Number of complexes	Recog. result	Number of complexes	Recog. result	Number of complexes	Recog. result	Number of complexes	Recog. result
d4	38	85 %	38	72 %	26	89 %	26	88 %
d92	118	54 %	31	73 %	111	34 %	46	57 %

Table 3 Results comparison for two modes of rule generation - disjoint cover mode (DC), intersection cover mode (IC)

6. Increasing Recognition Effectiveness through the Optimization of Class Descriptions

6.1. Motivation and Methodology

In the experiment described above, the texture description was directly applied in the recognition phase to generate classification hypothesis for the test data. However, when one considers that the learning texture data can be noisy and imperfect, one has to agree that the derived descriptions of texture classes consequently will contain noisy components. The application of such descriptions in the recognition phase causes the missclassification of some testing data. If these texture descriptions are filtered (e.g., optimized in order to remove some noisy components) before the recognition phase is executed then applied matching of testing data with reduced concept prototypes should give better recognition results.

The above conclusion motivated us to perform the second experiment. This experiment investigated rule optimization in order to compress the description of a concept and to reduce some of its noisy/imperfect components. Obviously, this optimization must be executed after the learning and before the recognition phases. We used a method of rule optimization that is based on the two-tiered description of imprecise concepts introduced by Michalski et al.[22] A simple two-tiered concept description generates both the Base Concept Representation (BCR) of typical properties of a concept, as well as the Inferential Concept Interpretation (ICI) of allowed concept modifications. The SG-TRUNC method was applied to obtain a BCR through a sequence of generalization and specialization operations.[35] Initially, the SG-TRUNC method performs generalization to remove selectors from the complexes. After such removal, a complex is more general, i.e., it covers more examples. Then, a specialization operation removes the number of complexes. In this way, the description covers less examples.

The rule optimization processes performed by the SG-TRUNC method are based on rule characteristics described in Section 3. This rule characteristics is composed of two coefficients, the t-weight and the u-weight. The t-weight is the total number of examples covered by a complex, while the u-weight is the number of examples covered by the same complex and no other. Applyed optimization method preserves those complexes that have high t- and high u-weights and modifies these complexes with low t- and u-weights. The degree of rule optimization is controlled by two parameters. The first parameter controls the removal of selectors and the second one controls the reduction of complexes.[35,36] increase in parameter values causes greater rule modification.

6.2. Recognition results

We applied the SG-TRUNC method as contained in the AQ16 algorithm. Relativel parameter values were applied, both equal to 5 (where a single parameter has a value fr to 100), to control the removal of selectors and complexes. This means that the optimiz of rules was low. Obtained recognition rates are presented in Table 4 and can be comp with the results presented in Table 1, where both tables illustrate results for the ma learning approach to texture recognition.[26] The analysis of these results gave u following conclusions:

1. The description of texture classes has been compressed significantly without decreasing recognition effectiveness, and a secondary effect of the optimization was the increase of recognition speed.

2. The recognition effectiveness, both for the Laws' masks method and the co occurrence matrix method, has been increased significantly.

3. The average deviation of the recognition rates has been reduced while the recognition effectiveness was increased.

	Texture description method					
	Laws' masks			Cooccurrence matrices		
	Number of generated complexes	Recognition First rank	Recognition Unique	Number of generated complexes	Recognition First rank	Recognition Unique
d4	6	95 %	1 %	5	90 %	42 %
d9	6	88 %	6 %	3	77 %	34 %
d19	3	90 %	25 %	3	86 %	61 %
d37	1	92 %	80 %	1	94 %	92 %
d92	7	83 %	2 %	12	75 %	32 %
d93	5	98 %	39 %	4	96 %	61 %
Average recognition		91 %	25 %		86 %	54 %
Averaged deviation		4.1	22.7		7.	[illegible]

Table 4 Recognition results for combined learning from examples and rule truncation method (AQ16 and ATEST programs) - intersection cover mode

The final summary of recognition effectiveness for three methods (machine learning with optimized rules, machine learning without applied optimization of rules, and k-NN pattern recognition method) is presented in Table 5.

Through the application of rules optimization, we were able to improve overall recognition results. The average recognition rate increased to 91% (from 80%) in the case of the Laws' masks method and to 86% (from 83%) for the co-occurrence matrix method applied to extract texture attributes. The recognition rate was significantly increased (up to 83% and 75% from 73% and 57%, respectively) for the recognition of the pigskin texture. In this way, the minimum recognition rate for both methods for extraction of texture attributes was improved. Moreover, the variation of recognition rates over texture classes has been reduced; i.e, the recognition rates have been smoothed. The smoothing effect has been observed for both methods of extraction of texture attributes.

	Texture description method					
	Laws' masks			Cooccurrence matrices		
	ML approach optimal rules	ML approach rules	PR approach k-NN method	ML approach optimal rules	ML approach rules	PR approach k-NN method
Average recognition rate	91%	80%	70%	86%	83%	83%
Highest recognition rate	98%	93%	99%	96%	99%	78%
Lowest recognition rate	83%	72%	40%	75%	57%	49%
Averaged deviation	4.	7.	16.3	7.	10.	14.4

Table 5 Summary of results for machine learning (ML) and pattern recognition (PR) approaches

6.3. *Characteristics of Recognition Effectiveness*

The optimization processes applied to modify rule descriptions are controlled by parameters. These parameters influence on the degree of performed removal of selectors and reduction of complexes. If they are increased then the description of texture classes are more compressed. However, it is difficult to set up these parameters a-priori. If the compressed descriptions perform better during the recognition phase then the question arises of what is the maximum value of each parameter that allows such optimized rules to perform better. To answer to this question, we decided to obtain characteristics of the recognition effectiveness versus the values of control parameters.

These characteristics were obtained for the average recognition rate calculated over the set of six classes of texture presented in Figure 1a and the Laws' method of texture attributes extraction. The individual characteristics of selected classes of texture were obtained, as well. We applied the AQ16 program[36] to perform the acquisition, optimization and recognition processes within its integrated software environment. Unfortunately, integrated software package does not give us necessary flexibility in tuning the recogn part of the software system. The recognition processes of the AQ16 program cann controlled by externally defined parameters. In this way, the flexible matching implem within the AQ16 program does not perform as well as implemented within the A' program that can be tuned to the specific domain of texture attributes (as applied in the experiments.)

Bacause of mentioned limitations of the AQ16 program, results presented below cann compared to the first experiment with machine learning and pattern recognition appr to the texture recognition problem. They are presented to illustrate the behavior recognition effectiveness versus increasing optimization parameters. Both optimi parameters responsible for removal of selectors and reduction of complexes were incr gradually by 1 from 0 to 16. Received recognition rates for six classes of images (1 1a) were then averaged and are presented graphically in Figure 6.

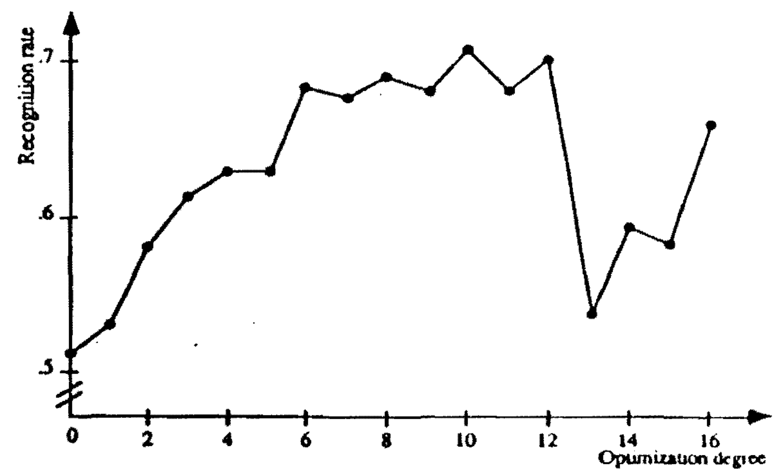

Fig.6 Diagram of the average recognition rate versus rule optimization degree (for the AQ16 integrated system - recognition rates are lower due to the non-tuned recognition system)

It is seen that average recognition rate was increased in the first range of optimiz parameters from 51% to 71%, i.e., for the parameters value from 0 to 10. Any fu increase of optimization degree caused a major drop of the average recognition rate. sharp drop was caused by the optimization system reducing the class description significantly that it decreased their classification power. This reduction removed correc significant components of class descriptions. This tendency is presented in Figure selected characteristics of two classes: lawn (d9) and the worst performer pigskin (d92). These characteristics confirm our observation that the optimization of noisy con can increase recognition effectiveness, but the applicable degree of such optimizat limited to the lower values and it is followed by a sharper decrease of the recognition p of applied methodology. Therefore, the increase in optimization parameters must be watched carefully.

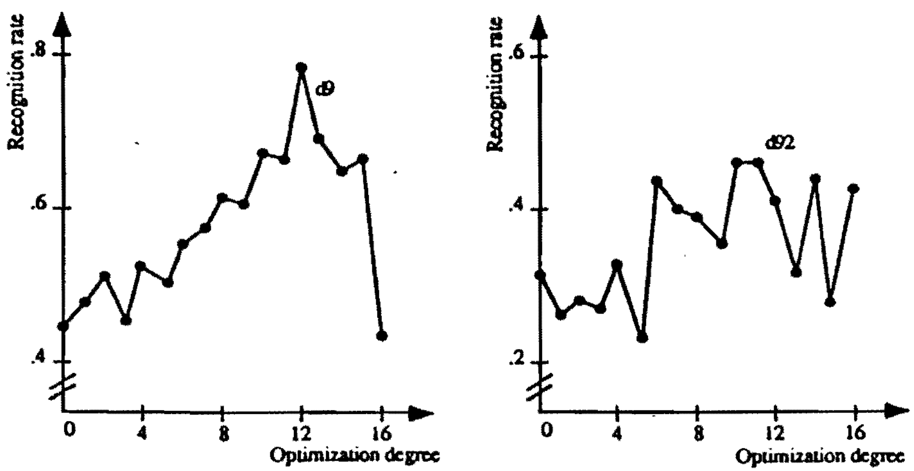

Fig. 7 Diagram of the recognition rate versus rule optimization degree taken for lawn (d9) and pigskin (d92) classes of texture (for the AQ16 integrated system - recognition rates are lower due to the non-tuned recognition system)

7. Recognition of Noisy Concepts through Multiple-Matching of Their Prototypes

In the experiments carried out, the optimization degree of texture class descriptions is controlled by parametric coefficients. If we request the higher optimization degree, we increase these coefficients. However, the increase of the optimization coefficients is combined with the risk that the optimized description will remove important components of a class description. Since the increase or decrease of the optimization degree is based on the noise evaluation, the noise prediction is necessary to control the optimization process properly. This is, however, a very difficult task to perform in the systems that have to work autonomously, especially of the close-loop architecture integrating recognition and learning processes for model evolution.

The experiments with the optimization of concept descriptions, presented in section 6, gave us some very interesting observations. These observations lead to the development of a novel recognition methodology. This methodology is based on the consecutive optimization of concept prototypes and matching these prototypes with test data. The novelty of this method is that the recognition decision is based on the observation of the behavior of recognition curves over a sequence of optimization degree rather than on a single matching of concept prototypes.

The idea of optimization of a concept prototype in order to remove noisy/imperfe components has great advantage in such situations when the quality of learning example presented to the system is low. This low quality can be caused not only by the low qualit of image data or inefficient methods for the extraction of attributes, but it can be mostl caused by the system inefficiency in providing learning examples. It means that the syster is working without teacher assistance, and it has to activate learning processes when th concept prototype must be updated according to a new information. Thus, the quality o updated concept descriptions depends on the system capability of selecting learnin examples corectly.

7.1. Multiple-Matching System Architecture

A series of new experiments[1] has been executed incorporating modified architectu presented in Figure 8. The modifications (when one compares with the former architectu in Figure 5) include the extraction of texture attributes, re-designing of the recogniti system, implementation of a new decision making methodology, and the creation of a clos control loop of the iterative optimization of rules and the recognition of test data. Th experiments were carried out with the second set of texture images (see Figure 1b) and wi the simplest method of attributes extraction.

The learning system is composed of the AQ15 inductive learning program. Acquired rul were applied to the recognition phase, where testing texture samples have been shown to t system. The recognition process has been redesigned and arranged into the closed loo This loop consists of three modules: rules optimization, inductive assertion (ATE program), and a module of control and decision making. The loop is controlled by optimization parameter. For a given parameter, the system optimizes rule descriptions a forwards them to the inductive assertion module. Inductive assertion processes a performed each time for newly optimized descriptions and on the same set of test data.

The system increases the optimization degree for each iteration loop. The decision maki module completes partial classification rates computed for each optimization loop. In t way, a recognition curve is created for each set of partial descriptions of learned clas versus the optimization degree. Finally, the classification decision is made based on t characteristics of the recognition curves.

Since our goal is focused on the development of a novel approach to the recogniti problem that deals with concept descriptions acquired from the noisy training data, we we not looking for a more powerful method applied for the extraction of texture attribute Contrary to the development of more powerful, i.e., noise reducing methods, we hav applied the simplest extraction of texture attributes that preserves the negative influence noise and imperfectness of attribute acquisition on the learning phase of texture cla descriptions from provided examples. This extraction of texture attributes was executed the following way. A 5x5 window was applied to compute eight local attributes. Ea attribute was calculated as a difference between the central pixel p(i,j) and one of t pixels belonging to the window's edge (i.e., p(i-2,j), p(i-2,j+2), p(i,j+2), p(i+2,j+ p(i+2,j), p(i+2,j-2), p(i,j-2) and p(i-2,j-2).) Two hundred randomly selected traini events were presented during the learning phase, while the recognition phase w performed with one hundred testing events.

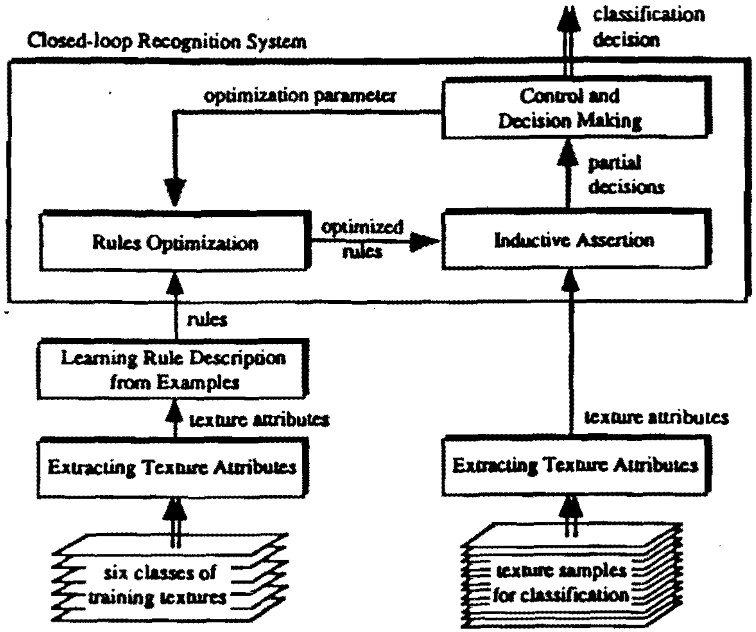

Fig.8 Closed-loop architecture for texture recognition incorporating iterative optimization of concepts prototypes and their matching with testing data

7.2. Recognition Methodology

Traditional approach to the concept recognition assumes that to recognize a concept one needs to match the observed data with a stored concept prototype. We have already discused the issue of matching a concept prototype with testing data. Flexible matching was suggested as a preferred method rather then the strict matching. In flexible matching, an approximate match is computed and used as a measure of confidence in recognizing particular concept among others. A concept prototype can have different less or more optimal forms depending on the applied optimization degree. We have already shown that the recognition result depends on the applied optimization degree and that the a-priori set up of the optimization parameters is very difficult to achieve. The solution of this problem can be seen in the application of a multiple-matching when testing data is matched several times with the gradually reduced (optimized) concept prototypes. This idea, when applied on experimental data, gave us very interesting observations.

First of all, we simplified the optimization method of rule descriptions. A new method takes into consideration the t-weight characteristics of complexes within a rule. This t-weight characteristics of a complex may be interpreted as a measure of typicality or the representativeness of this complex within a concept description. A complex with high t-weight is viewed as describing the most typical concept properties, while a complex with

very low t-weight is viewed as describing concept exceptions from the typical propertie Such an exception from the typical properties can be viewed as a noisy component. So, tl removal of complexes characterized by very low t-weight can be interpreted as the remov of noisy components from a concept description.

The rule complexes are ordered according to the decreasing values of the t-weight. O approach to the optimization processes applies then the removal of a given number of tl less significant (the lightest) complexes from the descriptions of all candidate concepts. Tl optimization of rule description by removing its lightest components is called compl truncation. Thus, the simplest definition of a truncation degree (as an optimizatio parameter) points out the number of the lightest complexes to be removed from the origin description. How such complex truncation can have the influence on the improvement recognition effectiveness? The answer is simple - because we remove noisy compone from the concept description, i.e., we increase the probability that the testing data will l classified through flexible matching to a more typical complex.

Let us proceed further and ask whether it is possible to reverse the classification hypothes It means that the system classifies a set of testing data to one class for the non optimiz concept descriptions, and after applying optimization processes, it classifies the same testi data to the other class. It means that, the improvement of the class description that the te data was classified incorectly removes noisy components from the class description and that way decreases the recognition rate for this class. In the same time, the optimizati processes remove noisy components from the class description that should be recogniz correctly. If these optimization processes are balanced properly, they can reverse tl classification hypothesis.[1]

Such a possibility of reversing classification decision is difficult to achieve and it high depends on the distribution of learning examples in the attribute space, the form a characteristics of generated complexes, and noise overlaped with learning and testi events. Decision making based on the recognition rates obtained for a single matchi approach is very difficult, however, the observation of recognition curves obtained throu the increasing truncation degree can help to support final classification decision. Based the behavior of the recognition curve, we develop a novel recognition methodology as complement to the traditional single match approach.

A new recognition algorithm incorporates the characteristics of the recognition curv obtained for matching test data with the gradually optimized descriptions of differe classes. The recognition curve has "uptrend pattern" if the recognition rate increases wi the increase of the truncation parameter, or it has "downtrend pattern" if the recognition ra decreases with the increase of the truncation parameter. Such characteristics is being bu only for the low truncation degree because the higher optimization can cause the removal significant complexes as well. Incorporating such characteristics of recognition curves, w define the following three step recognition algorithm:

Step 1. Label the first section of each recognition curve as the "uptrend" section or "downtrend" section,

Step 2. Select these recognition curves that have the "uptrend" pattern only, and

Step 3. Make the final decision indicating this class for which the "uptrend" pattern runs through the highest recognition rates. In case of no "uptrend" recognition curves, return "NO-DECISION".

7.3. Experimental Results

The recognition curves obtained incorporating introduced system architecture and recognition methodology are illustrated in Figure 9 (for the image data from Figure 1b).

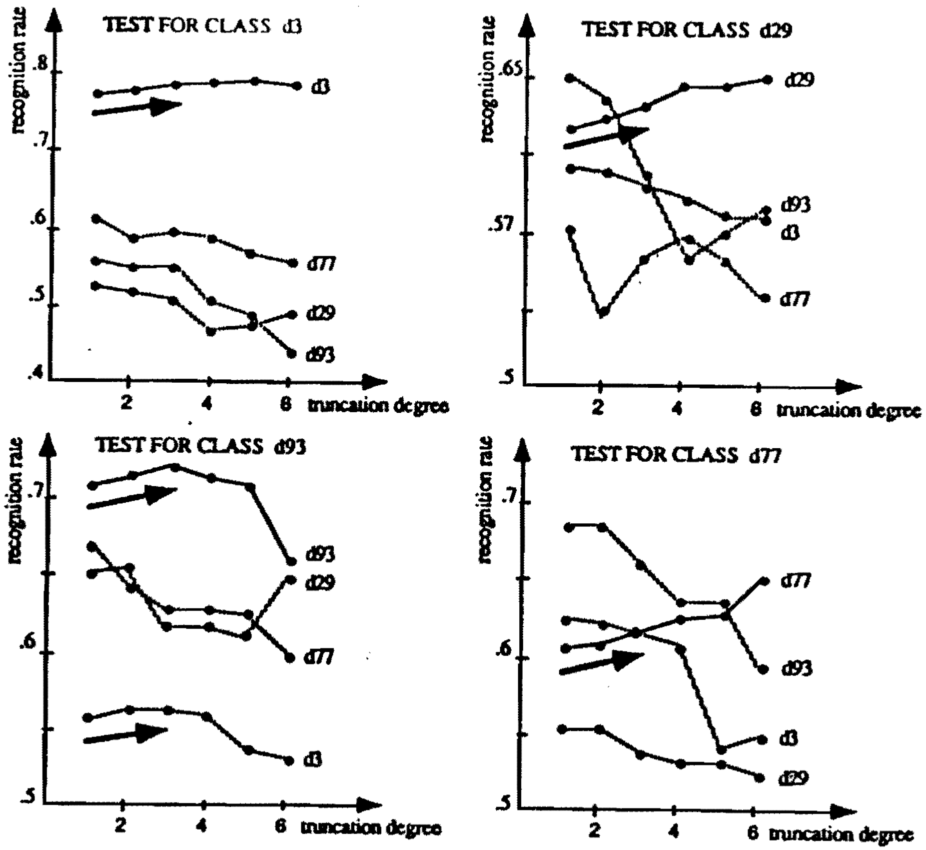

Fig.9 Recognition curves for the experiment with the closed-loop recognition incorpotating an idea of multiple-matching (an arrow indicates recognition curves with uptrend pattern, solid line - a class that should be recognized and was recognized correctly by the system, dotted line - other classes)

During the learning phase, the system completed four sets of training example (corresponding to texture classes d3, d29, d77 and d93), each set consisting of 20 examples selected randomly from training sections of texture images. Acquired rul descriptions of these four classes were applied to recognize test sets of data, each se consisted of 100 testing events but selected from different sections of the texture image.

In Figure 9, obtained recognition curves (for each set of testing events) are marked by a arrow if they have "uptrend" pattern. The solid curve represents the class that should b recognized correctly and it was recognized by our method. Other classes are characteriz by dotted lines. The truncation degree corresponds to the number of lightest complexes of rule that were removed before the matching step. The recognition rate was computed as th number of test samples recognized to a given class divided by the total number of te samples of a given dataset. It should be pointed out that a single testing event can b recognized to more than one class when it is covered by complexes of different classes o the distance to such complexes is the same.

Applied recognition methodology recognized correctly all four classes of texture, while th traditional single-match approach failed to recognize two of them; i.e., cotton canvas (d77 and beach sand (d29). Both of these classes have a typical uptrend pattern of th recognition curve, while other curves have the "downtrend" pattern. The decision makin process, however, performed for the fur class (d93) considers two "uptrend" recognitio curves; i.e., classes d3 and d93. The final decision points out the class d93 because it recognition rates for the uptrend section of the recognition curve are higher than for the clas d93.

Textures of cotton canvas (d77) and beach sand (d29) are not recognized if one applie traditional single match method. However, they are successfully recognized applyin presented in this paper methodology based on the analysis of dynamic behavior of th recognition curve. It is seen that introduced new methodology can give us more effectiv recognition results in the case of noisy and imperfect data applied to learn class description what is in fact typical for engineering domain and for autonomous systems where a syste (rather than a teacher) has to choose training data.

8. Conclusions and Future Work

This paper has presented the application of symbolic machine learning approach to the texture recognition problem. Learning from examples methodology has been incorporated to acquire texture concept descriptions. Then, we applied an idea of optimization of concep prototypes in order to remove noisy components from acquired class descriptions. The recognition of texture testing data has been performed through flexible matching rather tha through strict matching. We have presented recently developed multiple matchin methodology[1] that allows to recognize concepts applying noisy/imperfect clas descriptions.

Three separate experiments have been presented to illustrate principles of introduced methods. The first experiment has compared the effectiveness of applied machine learnin methodology with the k-NN pattern recognition method. The second experiment ha illustrated characteristics of recognition effectiveness depending on the applied optimizatio degree. Finally, the third experiment has demonstrated how the characteristics of recognition rate was incorporated to develop a new recognition methodology that is based on the dynamic characteristics rather than static characteristics of the recognition rates computed for each texture class.

The results presented in the paper cannot be compared effectively to other works. They were received for different experiments. In some cases, they even differ between themselves because of applied different learning and/or matching tools, and different parameters that control these tools. However, these results confirm introduced approaches, what was the main goal to achieve.

The paper has presented our introductory experiments with the application of symbolic machine learning to typical numeric domain such as texture recognition problem. These experiments and developed methods gave us the following conclusions as future research topics:

- the development of a new machine learning tool (i.e., learning from examples) to support the input of numerical attributes that is typical for engineering domain --- or to develop a model and a method for learning hierarchical concept description incorporating multi-level scaling method (i.e., quantization of numeric data into symbolic intervals on different levels of resolution),
- further development of the iterative matching recognition methodology including the development of optimization methods of concept descriptions and a method for characterizing dynamics of recognition curves,
- the integration of traditional single match and proposed multiple match methods for generation classification hypotheses and local unification of these hypotheses in order to segment texture image,
- the application of the incremental learning mode for the evolution of texture concept over time that is caused by variable texture occurrences perceived for different external conditions (e.g., noise, illumination) and projection parameters (e.g., resolution).

Presented in this paper methods and experiments demonstrates our introductory effort in the area of the integration of computer vision and machine learning. We believe that the future progress in both areas will be increased through their theoretical and experimental integration. We expect to contribute in the computer vision area through the implementation of more flexible and powerful tools on higher levels of vision architecture to build smart vision systems capable to adapt to a new task and variable environment. On the other hand, acquired experience in such integration will be used to develop new machine learning methodologies and to test them on specific applications.

Acknowledgement

The author wishes to thank Ryszard Michalski, Azriel Rosenfeld and Yiannis Aloimonos for constructive discussion under the issues of the integration of learning methodologies with machine vision area, Janet Holms and Michael Hieb for their help in the preparation of this

article. Special thanks are to Jerzy Bala for his cooperation, especially for caring out experiments presented in Section 7, and for technical help.

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 N00014-88-K0226, and in part by the Office of Naval Research under grant No. N00014-88-K-0397.

References

1. J.W. Bala. and P.W. Pachowicz, "Recognition of Noisy and Imperfect Texture Concepts via Iterative Optimization of Their Rule Descriptions", Report No 90-25, MLI-90-12, Center for Artificial Intelligence, George Mason University, 1990
2. B. Bhanu and J.C. Ming, "TRIPLE: A Multi-Strategy Machine Learning Approach to Target Recognition", Proc. DARPA Image Understanding Workshop, Cambridge, pp.537-547, 1988.
3. B. Bhanu, "Automatic Target Recognition: State of the Art Survey", *IEEE Transaction on Aerospace & Electronic Systems*, AES-22 (4), pp.364-379, 1986
4. R.C. Bolles and R.A. Cain, "Recognizing and Locating Partially Visible Objects: the Local-Feature-Focus Method", in Robot Vision, A. Pough (Ed.), Springer Verlag pp.44-81, 1983.
5. P. Brodatz, "A Photographic Album for Arts and Design", Toronto, Dover Publishing Co., 1966.
6. G.R. Cross and A.K. Jain, "Markov Random Field Texture Models", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.PAMI-5, No 1, pp.149-163, 1983
7. L Davis, S. Johns and J. Aggarwal,"Texture analysis using generalized co-occurrence matrices", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-1, pp. 251-259, 1979
8. H. Derin and H. Elliot, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.PAMI-9, No.1, pp.39-55, 1987.
9. J.M.H. DuBuf, M. Kardan and M. Spann, "Texture Feature Performance for Image Segmentation", *Pattern Recognition*, Vol.23, No.3-4, pp.291-309, 1990.
10. O.R. Duda and P.E. Hart, Pattern Classification and Scene Analysis, John Wiley&Sons, 1973.
11. J. Hong, I. Mozetic and R.S. Michalski, "AQ15: Incremental Learning of Attribute-Based Descriptions from Examples, the Method and User's Guide", ISG 86-5, UIUCDCS-F-86-949, Department of Computer Science, University of Illinois, Urbana, 1986.
12. J.Y. Hsiao and A.A. Sawchuk, "Unsupervised Textured Image Segmentation Using Feature Smoothing and Probabilistic Relaxation Techniques", *Computer Vision Graphics and Image Processing*, Vol.48, pp 1-21, 1989.
13. K. Ikeuchi and T. Kanade, "Modeling Sensors and Applying Sensor Model to Automatic Generation of Object Recognition Program", Proc. DARPA Image Understanding Workshop, Cambridge, pp.697-710, 1988.
14. K.I. Laws, "Textured Image Segmentation", PhD Thesis, Dept. of Electrical Engineering, University of Southern California, Los Angeles, 1980.

15. C.-E. Liedtke and M. Ender, "A Knowledge Based Vision System for the Automated Adaptation to New Scene Contents", Proc. 8th Int. Conf. on Pattern Recognition, Paris, pp.795-797, 1986.
16. S.-S. Liu and M.E. Jernigan, "Texture Analysis and Discrimination in Additive Noise", *Computer Vision, Graphics and Image Processing*, Vol.49, pp.52-67, 1990.
17. D. Marr, "Vision", Freeman, San Francisco, 1982.
18. T. Matsuyama, "Expert Systems for Image Processing: Knowledge-Based Composition of Image Processes", *Computer Vision, Graphics and Image Processing*, Vol.48, pp.22-49, 1989.
19. R.S. Michalski, "AQVAL/1 - Computer Implementation of a Variable-Valued Logic System VL1 and Examples of its Application to Pattern Recognition", Proc. of the First Int. Joint Conf. on Pattern Recognition, pp.3-17, Washington, DC, 1973.
20. R.S. Michalski, "A Theory and Methodology of Inductive Learning", in Machine Learning: An Artificial Intelligence Approach, TIOGA Publishing, Palo Alto, CA, pp 83-134, 1983.
21. R.S. Michalski, I. Mozetic, J. Hong and N. Lavrac, "The AQ15 Inductive Learning System: An Overview and Experiments", ISG 86-23, UIUCDCS-R-86-1260, Department of Computer Science, University of Illinois, Urbana, 1986.
22. R.S. Michalski, "Two-Tiered Concept Meaning, Inferential Matching and Conceptual Cohesiveness", in "Similarity and Analogy", S. Vosniadou and A. Orton (Eds.), Cambridge University Press, 1987.
23. K.S. Narendra, (Ed.), "Adaptive and Learning Systems", Plenum Press, 1986.
24. B. Neumann, "Towards Computer Aided Vision System Configuration", in Artificial Intelligence II: Methodology, Systems, Applications, Ph. Jorrand and V. Sgurev (eds), pp.385-393, Elsevier Pub., 1987.
25. P.W. Pachowicz, "Low-level Numerical Characteristics and Inductive Learning Methodology in Texture Recognition", Proc. of the IEEE International Workshop on Tools for AI, Washington, D.C., pp.91-98, October 1989.
26. P.W. Pachowicz, "Integrating Low-Level Features Computation with Inductive Learning Techniques for Texture Recognition", *International Journal of Pattern Rec. and Artificial Intell.*, Vol.4, No.2, pp.147-165, 1990a.
27. P.W. Pachowicz, "Learning-Based Architecture for Robust Recognition of Variable Texture to Navigate in Natural Terrain", Proc. IEEE International Workshop on Intelligent Robots and Systems '90, Tsuchiura, Ibaraki, Japan, pp.135-142, 1990b.
28. S. Pinker, "Visual Cognition", MIT Press, 1985.
29. T. Poggio, J. Little, E. Gamble, W. Gillett, D. Geiger, D. Weinshall, M. Villalba, N. Larson, T. Cass, H. Buelhoff, M. Drumheller, P. Oppenheimer, W. Yang, A. Hurlbert, "The MIT Vision Machine", Proc. DARPA Image Understanding Workshop, Cambridge, pp.177-198, 1988.
30. R.E. Reinke, "Knowledge Acquisition and Refinement Tools for the ADVICE META-EXPERT System", ISG 84-4, UIUCDCS-F-84-921, Department of Computer Science, University of Illinois, Urbana, 1984.
31. S.J. Roan, J.K. Aggarwal and W.N. Martin, "Multiple resolution imagery and texture analysis", *Pattern Recognition*, Vol.20, No.1, pp.17-31, 1987.
32. L. Van Gool, P. Dewaele and A. Oosterlinck, "Texture Analysis Anno 1983", *Computer Vision, Graphics and Image Processing*, Vol.29, pp.336-357, 1985.
33. H. Wechsler and L. Zimmerman, "2-D Invariant Object Recognition Using Distributed Associative Memory", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.10, No.6, pp.811-821, 1988.

34. J.S. Weszka, C.R. Dyer and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification", *IEEE Trans. Systems Man Cybern.*, Vol.SMC-6, No.4, pp.269-285, 1976.
35. J. Zhang and R.S. Michalski, "Rule Optimization via SG-TRUNC Method", Proc. of the Forth European Working Session on Learning, Pitman, Morgan Kaufamnn, pp.251-262, Montpellier, Dec. 1989.
36. J. Zhang, "Learning Flexible Concepts from Examples: employing the idea of Two-Tiered Concept Representation", Ph.D. Thesis, Computer Science Department, University of Illinois, 1990.