P92-39

RECOGNIZING NOISY PATTERNS VIA ITERATIVE OPTIMIZATION AND MATCHING OF THEIR DESCRIPTIONS

J. W. BALA and P. W. PACHOWICZ

Center for Artificial Intelligence
George Mason University, Fairfax, VA 22030, USA

This paper presents an approach to the recognition of noisy patterns. We assume that learned concept descriptions contain noisy components. These descriptions can be optimized to gain better performance through the elimination of some less significant noisy components. Parametric control of such an optimization, however, is very difficult and a parameter value has to be found through the training process. This value can change over concept variations and time, and in consequence, it can decrease the system recognition effectiveness. We found that the classification decision can be based on the dynamic characteristics of the recognition curve acquired over increasing degree of optimization. In this approach, the optimization and matching steps are repeated iteratively. The dynamic characteristics of recognition curves are completed and a pattern of each recognition curve is labeled. For a given set of test data, the system selects recognition curves of uptrend patterns (when the recognition rate increases with the increase of the optimization degree), and the final classification is made for the class which has the highest recognition rates among selected uptrend recognition curves. Thus, the method classifies data based on a series of matches that determines the behavior of the recognition curve instead of the traditional single match. The effectiveness of the proposed methodology is illustrated on the example of specially prepared very noisy texture attributes.

Keywords: Recognition of noisy patterns, applications of machine learning.

1. INTRODUCTION AND MOTIVATION

Future intelligent systems can be characterized by their ability to acquire a concept of an object such as a description of the object's characteristics or properties, and to apply this concept to recognize a particular object among many other objects. Since engineering domains are noisy and extracted attributional characteristics can be irregular, the acquisition and robust recognition of noisy, imperfect and complex concepts is crucial for many tasks, for example, for the tasks of machine perception like inspection, object recognition, object localization and navigation, remote tracking, and intelligent surveillance. The success of these tasks depends both on the effectiveness of the data 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 affected by a different noise for instance. Moreover, a given object can be perceived differently when perceptual conditions are changing. Variability of object characteristics requires the development of capabilities that will allow the system to reconfigure, tune, and update its model (knowledge) regarding the object or sensor movement or the variation

International Journal of Pattern Recognition and Artificial Intelligence Vol. 6 No. 4 (1992) 513–537

of sensing conditions (e.g. changing resolution, lighting, overlapped noise). One of the possible solutions to this problem is development of a system adaptability function. Traditionally, control engineering[1] and/or pattern recognition[2] techniques are applied to provide adaptability in engineering systems. These approaches work well using quantitative information. However typical processes of high-level vision are based on symbolic computation. The border line between numeric and symbolic computation has not been established.

We have already proposed to develop system adaptability functions incorporating machine learning methodologies applied to such tasks as object localization, recognition, inspection, and scene understanding.[3-5] 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[6] within a unified system. For example, a typical task dependent on system adaptation capability is texture-based object recognition for robot navigation in a natural terrain. The difficulties of robust recognition of 3-D objects and annotation of perceived surface areas are caused by a great variability of images projected onto the sensory array. This variability is caused by rotation of an object, changes of resolution, lighting conditions, and camera parameters.

Tools supporting system adaptability must be developed with respect to a given task and the application domain. Artificial intelligence (AI) is well suited to handle human-oriented tasks—this is the objective of AI. For example, machine learning tools can provide natural adaptability functions when applied within an engineering system. Unfortunately, tools of artificial intelligence are noise sensitive. They are not magically robust, and their performance depends on the quality of the training data provided. While the complexity of training data and acquired concepts can be better handled by machine learning than by pattern recognition tools, we find noise in engineering data as the bottleneck in the application of machine learning. Therefore, it is very important to develop approaches that will be able to perform despite a high level of noise in the data.[7] Both concept learning and object recognition approaches that do not tolerate noise are of limited value.

Noise in engineering data affects an intelligent system in different ways. Noise comprises non-systematic errors in the values of attributes or class information.[8] Errors in the attribute value cause the correct value not to be determined. Errors in the class information cause incorrect training decisions to be built into concept descriptions. Some sources of noise can be modeled. Real-world situations, however, are too complex and the noise distribution is usually unknown. The noise affecting engineering data is a combination of different regular and irregular distributions. Thus it is very difficult to model such noise, especially errors caused by incorrect training classifications.

The traditional approach to the recognition of objects is focused on the following three phases:

(1) *feature extraction phase*—the design and selection of a set of most representative features that preserve as much of the discriminatory power of applied methodology as possible,
(2) *concept acquisition phase*—learning descriptions of object classes from teacher- or system-provided training data incorporating the separation of feature space into disjunctive sections or via the modeling and representation of feature clusters, and
(3) *concept recognition phase*—matching test data with derived class descriptions.

The overall object recognition effectiveness depends then on the quality of all the above presented phases. In the traditional approach to object recognition, the system is associated with a set of characteristic object features, which are used for recognizing particular objects. Such a set of features is constant[9] in that the perceptual system is unable to verify the usefulness of the object model. Thus, the classification is correct as far as the accuracy of the off-line created recognition model is concerned.

On the other hand, the traditional approach 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.[10]

The main problem with the 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 the above mentioned disadvantages; i.e. the results are affected by the noise and imperfection of learning and the 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.

When one considers that an object can have variable occurrences (caused by the dynamics of environment) then selected features should be sensitive in a wider range of object characteristics rather than in a very narrow range. From that viewpoint, a serious contradiction arises in selecting highly dedicated features (usually more powerful and able to filter noise out of the data) versus more flexible features (usually less powerful and noise sensitive).

Due to the mentioned disadvantages of the traditional approach to object recognition (i.e. the design of more sophisticated feature extraction methods in order to increase recognition effectiveness), we seek other methods of concept learning and their improvement according to the specifics of the engineering domain. Machine learning is an example of such an unexplored (from the engineering point of view) concept acquisition technology. It allows us to modify acquired concept descriptions, for example, to reduce noisy components before these descriptions are applied in the recognition phase.

In the recognition phase, optimized concept descriptions are applied to classify test data. The traditional approach to the recognition phase is based on the measurement of the closeness of test data to the concept. This closeness is represented by a single value. The classification decision yields a class of the shortest distance to the test data. It means that the traditional approach to the recognition phase is static: i.e. it is based on a single measure.

In this paper, we develop a new approach to the recognition of noisy patterns. Considering the situation where concepts can be optimized with different parameter degrees, the distance value can vary. We can repeat the matching process with optimized concept prototypes with different optimization degrees. The classification decision can then be made based on the dynamic behavior of the obtained recognition curve rather than on a single value. In this approach, the problem concerning the choice of the optimization degree is automatically solved; i.e. the proposed recognition method does not require knowledge of this value before the classification is run.

In Sec. 2, we present and justify the applied concept acquisition and optimization method. Section 3 presents the traditional approach to the recognition phase along with examples demonstrating dependency of recognition effectiveness on degree of optimization. The proposed approach to the recognition phase is presented in Sec. 4. Section 5 presents the experimental engineering domain, and Sec. 6 provides experimental results. Finally, Sec. 7 summarizes our work.

2. LEARNING CONCEPT DESCRIPTIONS

2.1. Concept Acquisition Techniques

In the past, many concept acquisition techniques have been developed as tools for concept learning. These techniques are grouped into the methods of pattern recognition, neural nets, and AI-oriented machine learning approaches. There is no single universal concept acquisition technique that can be applied to different domains and requirements for a learning task. Each of them has its advantages and disadvantages. For example, parametric methods of pattern recognition assume the distribution of attributes to be known and possibly to be approximated by parametric equations. Non-parametric methods of pattern recognition require storing training examples (all examples, some selected examples, or modeled "exemplars") in order to perform the recognition task. The neural net approach does not allow us to understand how the learning process was performed and how we can manipulate acquired knowledge. Finally, AI-oriented machine learning approaches, developed for simple problems, are noise sensitive and require a lot of computation.

The very powerful parametric methods of pattern recognition[2] have limited application. However, they are applied frequently without consideration of the assumptions that have to be fulfilled. The most restrictive assumption requires the distribution of attributes to be known *a priori*, and especially to be normal. The distribution of the attribute is not regular in many application domains including texture recognition. We have already proven that the distribution of attributes can be non-normal, in particular, it can be multi-modal or totally random, indicating its irrelevance to the discrimination of object classes.[5,11]

Based on the analysis of disadvantages of different concept acquisition methods, we have chosen AI-oriented learning methods to learn and represent object models. These methods have the following advantages: they can (i) describe very irregular distribution of training data, (ii) integrate numeric and symbolic attributes, (iii) use background knowledge (domain specific), (iv) apply multi-strategy learning for learning from very complex data, and (v) represent acquired knowledge in the understood (for humans) form. AI-oriented learning tools, however, have to be redesigned for engineering applications, especially to increase the speed of computation, eliminate noise sensitivity of these tools, and to improve the incremental concept acquisition mode.

2.2. Learning Concept Descriptions from Examples

The attributional description of texture classes is acquired by concept learning from examples methodology. Concept learning is a subdomain of inductive inference, where the inductive assertion is expressed as a classification rule for assigning any object, seen or unseen, to a class. Concept learning systems are intended for discovering general classification rules from the examination of a given set of examples described in terms of a collection of properties. The examples are provided by a source of information, which can be a teacher who knows the concept or the environment on which the system performs experiments and from which it receives feedback.

We applied the AQ14 inductive learning program. However, other programs can be applied as well (e.g. see Refs. 8 and 12). 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.[13,14] The input consists of separated sets of learning events labeled by a class name and a characteristics for each class of texture, where each event contains a given number of attributes. The set of events characteristic for one class is a set of positive examples. The program finds an optimal cover over only positive examples, where examples belonging to the other classes are negative examples. The learning process is repeated for each class (a set of positive examples) to create rule descriptions of all texture classes.

A description of a class in the AQ program is a disjunctive normal form (DNF) which is called a cover. A cover is a disjunction of complexes. A complex is a conjunction of selectors, and a selector is a form:

$$[L \# R] \tag{1}$$

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 AQ program, each generated complex is associated with a pair of weights; i.e. total (t-weight) and unique (u-weight). The following is the example of a complex:

$$[x1 = 1 \ldots 3][x2 = 1][x4 = 0][x6 = 1 \ldots 7][x8 = 1] \quad \text{(total: 6, unique: 2)} \tag{2}$$

where x_j is an attribute. 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 t-weight (see the example in the Appendix).

2.3. Optimizing Concept Descriptions

Most methods of machine learning research assume that concepts are precise entities, represented by a single symbolic description. In such a representation, the boundaries of a concept are well-defined. All instances of a concept are assumed to be equally representative. If an instance satisfies a given concept description, then it belongs to the concept, otherwise it does not. Moreover, most learning systems make the "noise-free domain" assumption; i.e. the examples presented to a system do not contain errors, the description language is complete, and consequently a system constrains its searches only for rules that are both consistent and complete. The requirement of "noise-free domain" cannot be satisfied in the real world, especially in such areas like signal processing and understanding, autonomous robotics, and intelligent systems. The flexibility of a concept, the variability of concept occurrences (caused by external conditions like resolution, illumination, and sensor positioning), and the noise in learning datasets prevent most learning systems from being applied to many real world learning tasks.

Different methods of noise handling have been proposed. Quinlan's work[8,15] is concerned with pruning decision trees. Two tree-pruning techniques have been developed to reduce the effect of noise: pre-pruning[8] and post-pruning.[15] Pre-pruning constrains the stopping criterion to prevent splitting of nodes when no feature provides a significant increase in information resulting from a split. Post-pruning operates in two stages. In the first stage, a complete and consistent decision tree is constructed. In the second stage, the subtrees which only cover a small number of examples are removed from the original decision tree. Rule truncation[14,16] is similar to tree pruning in that both approaches remove some components that cover only a small number of examples from complete and consistent descriptions in order to achieve better performance in the presence of noise. Like tree pruning, there have also been two rule truncation techniques: pre-truncation[17] and post-truncation.[14] Pre-truncation in CN2 uses a heuristic function to terminate search during rule construction, based on an estimate of the noise present in the data. This results in rules that may not classify all the training examples correctly, but that perform well on new data. In post-truncation, a complete and consistent description is generated first, and the disjuncts in the description are ordered according to the number of examples covered by them. Then the disjuncts that cover fewer examples than a predefined threshold are removed. In post-truncation, the disjuncts removed are small disjuncts.

In most inductive learning systems, a concept description is represented as a set of disjuncts. Each disjunct covers some positive examples, and jointly they cover all positive examples. Of these disjuncts, some cover a large number of examples, the others only cover a small number of examples. The disjuncts covering only a small number of examples are called small disjuncts.[18] Holte *et al.*[18] empirically proved that small disjuncts are much more error prone than large disjuncts.

Our approach to dealing with the negative influences of noise and the imperfection of models (i.e. acquired rule descriptions) applies optimization processes in order to remove less significant concept components (small disjunct or small complexes). In this way, we assume that the lightest complexes can represent noisy components of concept description and at least they represent the less typical characteristics of a concept. The application and comparison of rule optimization to the recognition problem has already been demonstrated on different kinds of real data.[3,4,19]

To optimize a rule description, we use the truncation method that was first introduced by Michalski *et al.*[14] In the AQ14 program, each generated complex is associated with a pair of weights; i.e. the t-weight (as the total number of positive examples covered by a complex) and the u-weight (as the number of positive examples uniquely covered by the complex). The complexes of a concept description are ordered according to the decreasing values of the t-weight. The t-weight may be interpreted as the measure of typicality or the representativeness of a complex. The complexes with the highest t-weight may be viewed as describing the most typical concept examples, and thus serve as prototypical descriptions.

Suppose we have a t-weight ordered disjunction of complexes, and we remove from it the "lightest" complex: i.e. the complex with the lowest t-weight. Such a truncated description will not strictly match events that do not uniquely satisfy the removed complex. However, by applying a flexible match (see next section), these events may still be closely related to the correct concept, and thus be correctly recognized. A truncated description is, of course, simpler but carries a potentially higher risk of recognition error. It also requires a more sophisticated evaluation.

We can proceed further and remove the second "lightest" complex from the cover, and observe the performance. Each such step produces a different trade-off between the complexity of the description on the one hand, and the risk factor along with the evaluation complexity on the other. At some step the best overall result may be achieved for a given application domain.

3. RECOGNITION OF NOISY CONCEPTS

3.1. Traditional System Architecture

Most approaches to the problem of object recognition are based on the traditional architecture illustrated in Fig. 1(a). This architecture shows the mutual connection between the training and recognition systems. Both systems are separated in such a way that the training system has to be run first in order to provide concept descriptions to the recognition system. There is no cooperation between both systems during the recognition phase.

Such traditional open-loop architecture was used in our first experiments with the application of machine learning to texture recognition problems.[3,4] The architecture of the system used in those experiments is presented in Fig. 1(b). The architecture of the

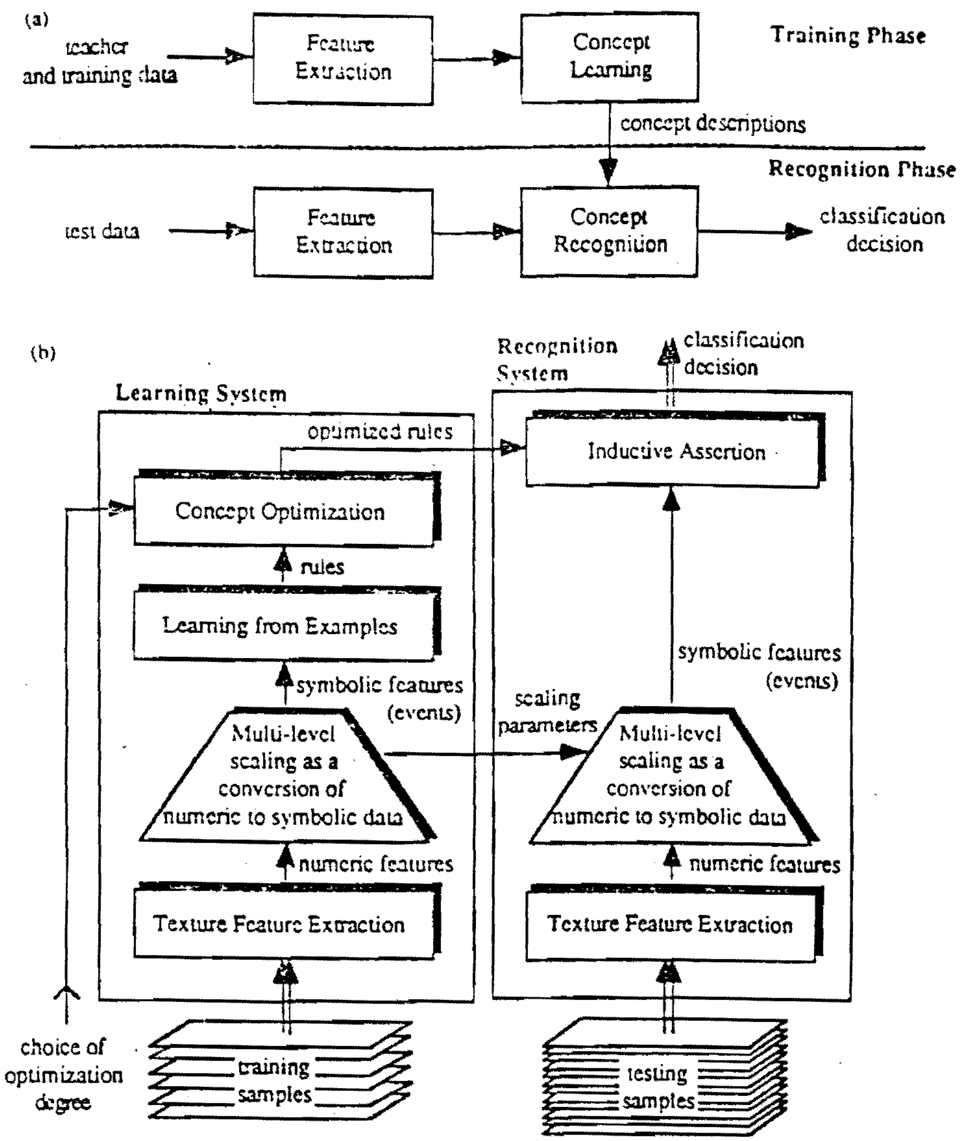


Fig. 1. Open-loop architecture for concept acquisition and recognition: (a) general architecture, (b) an architecture used in experiments with the machine learning approach to texture recognition.

created system is consistent with the traditional schema. The learning part is composed of the following modules: feature extraction, data conversion, learning from examples, and concept optimization. The recognition part of the system is composed of the same feature extraction and data conversion modules (i.e. the same as the learning part) and an additional inductive assertion module. The connection between the learning part and the recognition part of the system is one-directional, where concept descriptions along with scaling parameters are transferred to the recognition part of the system.

In such architecture, the choice of the optimization degree has to be determined by a teacher. This optimal degree can be found through the acquisition of recognition characteristics and the search for the most optimal optimization value. Such training, however, assumes that the teacher is well prepared and is able to interpret the data properly. Even if the teacher is able to find the optimization degree correctly, this degree can differ, for example, with changes in perceptual conditions. Then, the peak of recognition characteristics can be shifted out of the range of the initially selected optimization degree. An example for the recognition characteristic (prepared from our previous experiments, presented in Fig. 2(a), illustrates the sharp peak in recognition rate. This peak, however, is followed by sharp decline in the recognition rate. This decline is much faster than the preceding gain in the recognition rate when measured along with the increase in the optimization degree. This decrease is caused by the elimination of significant concept components from the concept description. In the second example, illustrated in Fig. 2(b), the variation of the recognition rate is too large to get a stable recognition level at a certain optimization degree. The choice of the optimization degree can be impossible because of the lack of recognition stability when this optimization degree varies slightly. The positive effect, however, is that the trend of the recognition rate is up with a local increase in the optimization degree.

The above explained problems inspired our investigations in the area of object recognition, where the static selection of the optimization degree can be eliminated or replaced by a dynamic selection. This work has given rise to the development of a new methodology for the recognition of noisy patterns presented in Sec. 4.

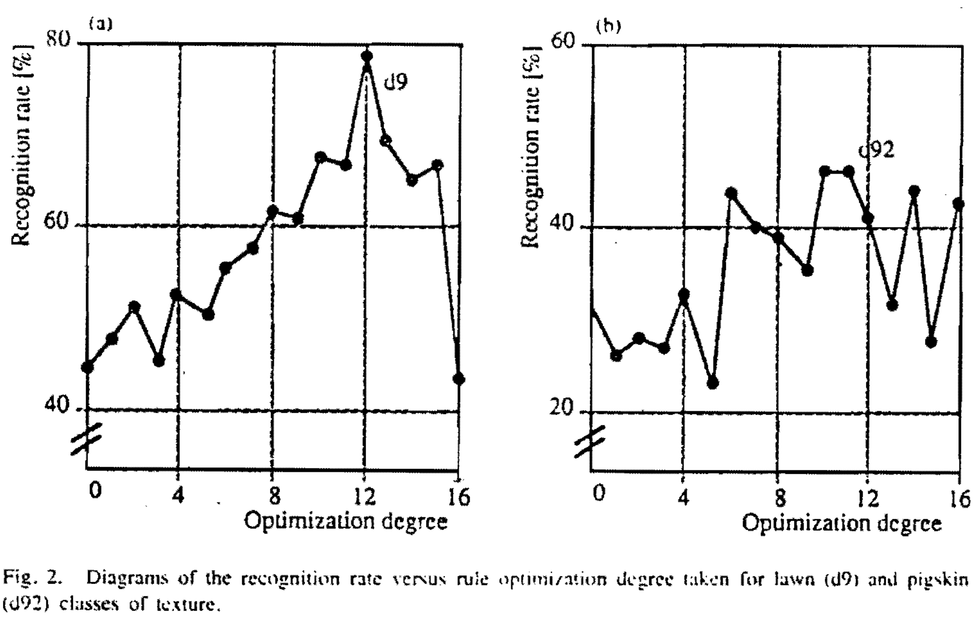


Fig. 2. Diagrams of the recognition rate versus rule optimization degree taken for lawn (d9) and pigskin (d92) classes of texture.

3.2. Flexible matching technique

Traditionally, the classification decision is based on the measure of the closeness of test data to concept descriptions. Then, the classification yields a class membership for which this closeness is the shortest. This process is frequently called the matching of test data with concept descriptions.

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 complexes). Such matching gives the response of two possible logical values, i.e. true or false. In the flexible match, the degree of closeness between the instance and the condition part is determined. Such closeness is represented by a coefficient that can vary in range from 0 (does not match) to 1.0 (matches). In the strict match, a concept is recognized if it overlaps with the concept description. In the flexible match, the most closely related concept description is determined.

The truncated description is the one that has some of its complexes with the lowest t-weight (i.e. the number of training examples covered by a complex) removed.[16] Such a truncated description will not strictly match events that satisfy the truncated complex. So, to evaluate the membership of an event in a texture class, one has to apply a flexible match. Such flexible matching can improve the classification decision when less significant components (i.e. possibly noisy components) are truncated from the class description.

In the recognition phase illustrated in Fig. 1(b), we incorporated a special tool, the ATEST program,[20] that had been developed to test the performance of a rule base. This program 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 or examples and this feature is extensively used in the texture discrimination phase. Second, ATEST provides routines that check a rule base for consistency and completeness. Third, it applies flexible matching to determine classification decision. Rule performance is measured by the degree of agreement between the system's and expert's classifications.

ATEST views rules as the expressions when applying them on a vector of attribute values. The result of this evaluation is a real number which is the degree of consonance between the conditional part of the rule and the event. ATEST takes as input (i) a set of attribute definitions, (ii) a set of rules, (iii) a set of testing events, and (iv) a set of parameters. One of the parameters, the evaluation parameter, controls the definition of elementary conditions, called selectors. A selector may be treated as a boolean condition (i.e. it may evaluate to 0 or 1), or as a function (i.e. a normalized real number between 0 and 1).

Given a selector in an attribute x whose domain is an ordered list $\langle a_1, a_2, \ldots, a_n \rangle$, and an event where $x = a_k$, the normalized value for the selector $[x = a_j]$ is

$$1 - (|a_j - a_k|/n)\,. \tag{3}$$

If the selector has several values on its right-hand side, the value closest to a_k is used. The complex is evaluated by calculating an average evaluation value for all selectors in the complex. For example, the evaluation of the following complex of a rule

$$[x1 = 0][x7 = 10][x8 = 10 \ldots 20] \tag{4}$$

with a given instance $x = \langle 4, 5, 24, 34, 0, 12, 6, 25\rangle$ and the number of attribute values # levels = 55 is computed as follows:

$$c_{x1} = 1 - (|0 - 4|/55) = 0.928 \tag{5}$$

$$c_{x7} = 1 - (|10 - 6|/55) = 0.928 \tag{6}$$

$$c_{x8} = 1 - (|20 - 5|/55) = 0.91 \tag{7}$$

$$c_{x2}, c_{x3}, c_{x4}, c_{x5}, c_{x6} = 1 \tag{8}$$

$$c = c_{x1}{*}c_{x2}{*}c_{x3}{*}c_{x4}{*}c_{x5}{*}c_{x6}{*}c_{x7}{*}c_{x8} = 0.78 \tag{9}$$

Thus, the total evaluation of a class description for a given testing example is equal to the evaluation value of the best matching complex (the complex with the highest evaluation value). The program is controlled by several parameters that determine the evaluation process. Most frequently, we apply the approximation of distance from the boundary of concept prototype to testing data.

4. CONCEPT RECOGNITION VIA ITERATIVE OPTIMIZATION AND MATCHING

4.1. Modified System Architecture

Based on the conclusions from Secs. 2 and 3, we decided to modify the system architecture. The modified architecture, presented in Fig. 3, involves the iterative optimization of concept descriptions and the iterative matching of test data with optimized descriptions. Modifications include (i) the redesigning of the recognition system, (ii) the implementation of a new decision making method, and (iii) the creation of a control loop of the iterative optimization of rules.

The learning system is generally the same as in Fig. 1(b), however, the concept optimization has been eliminated from the learning phase. So, the goal of the learning system is reduced to the acquisition of concept descriptions without the consideration of their optimality. Attributional descriptions of object classes are acquired by the same learning program, i.e. the AQ14 program. Acquired rules are then applied to the recognition phase, where testing samples are shown to the system.

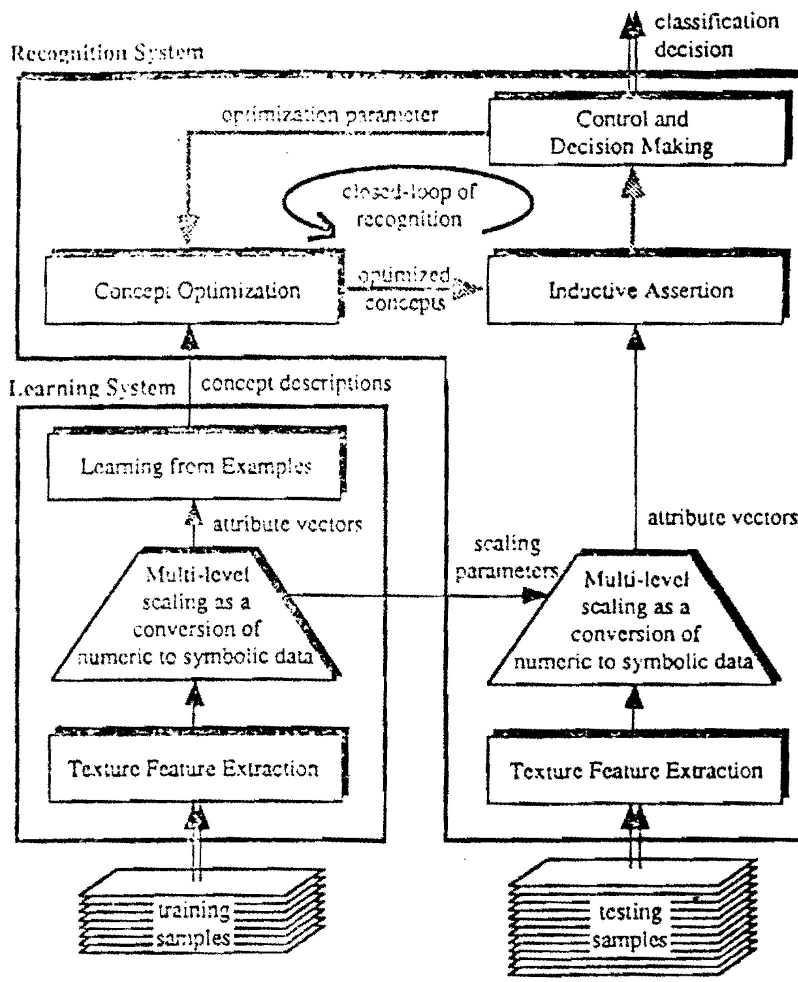


Fig. 3. Modified system architecture.

The recognition process has been redesigned and arranged into the iterative optimization loop. In this way, a recognition curve is created versus the optimization inductive assertion, and a module of control and decision making. The loop is controlled by an optimization parameter and it activates inductive assertion. Inductive assertion processes are performed each time for optimized concept descriptions. The system increases the optimization degree for each iteration loop. The decision making module completes belief values of partial recognition results computed for each optimization loop. In this way, a recognition curve is created versus the optimization degree. Finally, the classification decision is made based on the characteristics of the recognition curve.

4.2. Reversing Classification Decision

Existing approaches to the recognition problem assume that to recognize a concept one needs to match the observed data with a stored description of concepts. When the properties of the observed object match the stored description of a concept, then the name of the concept is returned. An approximate match is computed and used as a measure of confidence in recognizing a particular concept among others. A major problem with such a method is that in order to recognize an object one always needs to measure its same properties. Yet the same concept can be recognized using many different subsets of properties, with truncated concept descriptions.

In our method, we recognize textures by matching unknown texture samples with learned descriptions on the different truncation levels. The description on a given level is obtained by truncating a given number of less significant complexes from the lower part (i.e. less significant part) of the description. At each level, we match testing samples with given descriptions of object classes and complete recognition rates. For example, by matching a given sample with four class descriptions and on six different truncation levels, we have 24 recognition rates grouped into four recognition curves (each representing recognition effectiveness for a single class).

Let us demonstrate how such a truncation method (i.e. the optimization of concept description) can reverse the classification decision. Figure 4 presents a graphical representation of two class descriptions, class A and class B, where the domain of events consists of two attributes. Class A is described by four complexes, and class B by six complexes. The black dots represent sample vectors to be recognized. For the initial description of these two classes (Fig. 4(a)), four events (P6, P7, P8, and P10) are assigned to class A and six events (P1, P2, P3, P4, P5, and P9) to class B. By

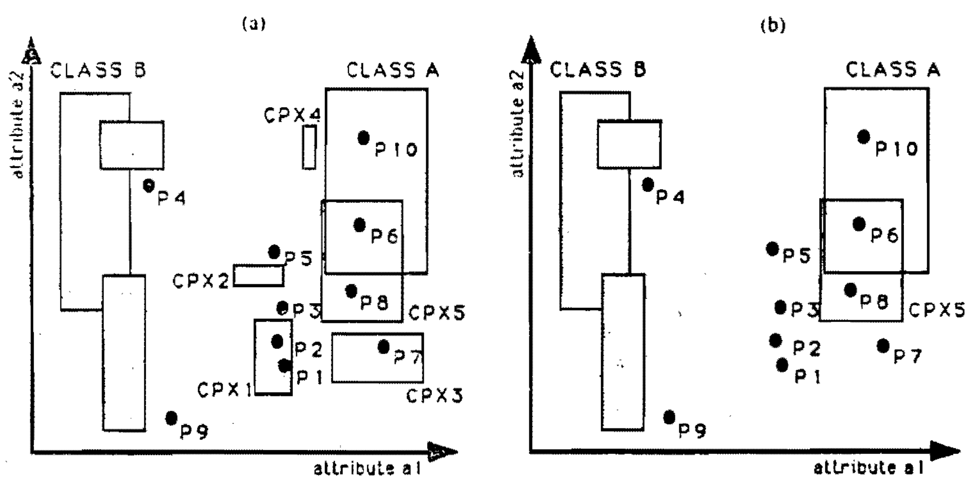


Fig. 4. An example for reversing the recognition decision for a set of 10 testing events: (a) before truncation, 4 events are classified to the class A and 6 events to the class B, (b) after truncation of CPX1, CPX2, CPX3, and CPX4 complexes, 8 events are classified to the class A and only 2 events to the class B.

truncating the CPX2 complex from class B, the P5 event is assigned to class A by closeness with the CPX5 complex, and the recognition membership (the number of samples classified to class A versus the number of samples classified to class B) is changed from 4–6 to 5–5. Further truncation of complex CPX1 of class B (see Fig. 4(b)) changes the recognition membership to 8–2 and class A yields the recognition. Notice that complexes are also simultaneously truncated from class A for the consecutive truncation degrees, i.e. complexes CPX3 and CPX4.

Such a recognition method assumes, however, that the removal of complexes is carefully done and can be supported by the t-weight and u-weight characteristics of these complexes. Most of these complexes can be created by only several positive examples. Such descriptions of concepts can possess built-in imperfect knowledge propagated by noisy teaching examples. Removing the "lightest" complexes from all descriptions, we optimize these descriptions through the elimination of less significant components. Proceeding further with such optimization, we can reduce concept description to the most significant clusters of learning data.

Theoretically, such reduction of rule description should lead to the extraction of main clusters of learning data. Applied flexible matching, should then classify test data to the nearest cluster. So, the approach seems to be very similar to the traditional pattern recognition methods and it would have all disadvantages characteristic of these methods.[2] On the other hand, there is the question of what the best optimization degree is. It is rather impossible to answer this question because the best optimization degree depends on the characteristics of feature distribution in the attribute space. If this distribution is complex, for example, then there will be many local clusters in the description of one class. In such a case optimization cannot be continually performed for higher degrees of optimization.

From the above discussion, we assume that such optimization could be applicable without significantly damaging the class description, i.e. it can be performed for the lowest truncation degrees. For the low truncation degree, the system would be able to remove a small number of complexes only, complexes that are less significant (with low t-weight). It has already been found through different experiments that the truncated description improves the performance of the recognition phase by reducing the noisy data coverage.[3,4,19,21–23] However, these experiments have been executed for a single truncation degree and such an approach has the disadvantage of finding the best optimization value.

4.3. Recognition Algorithm

If we accept that the truncation of less significant components from the concept description eliminates some noise, then the dynamic increase of the truncation degree (but performed in the lowest range of optimization values) should slightly increase the recognition effectiveness in this range. Such an effect was observed in diagrams presented in Fig. 2. We concluded then that a recognition curve created for each class description and for different optimization degrees could be a useful classification feature for the final decision making process.

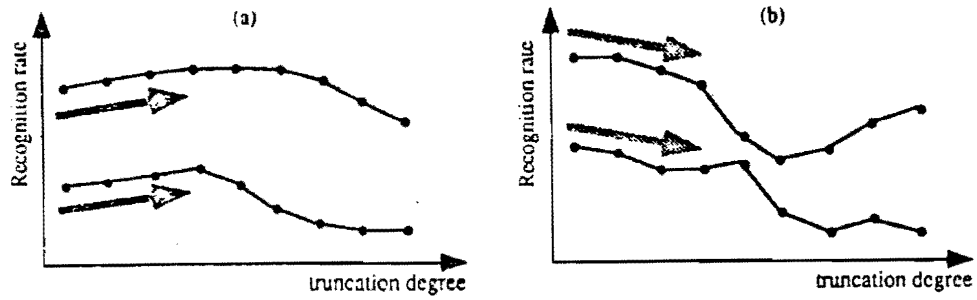


Fig. 5. Examples for (a) the uptrend pattern and (b) the downtrend pattern of the recognition curve.

Generally, we distinguished the following two patterns of the recognition curve with regard to the first phase of the increase in the truncation degree: (1) curves of the recognition uptrend, and (2) curves of the recognition downtrend. Figure 5 illustrates these two patterns of hypothetical recognition curves, acquired by smoothing the original recognition results versus increasing optimization (truncation) degree. The pattern of the uptrend recognition curve is divided into uptrend section, plateau, and downtrend section. The plateau section can be very short or significantly long before changing the trend direction. Such a division of the curve pattern is typical for a class that should be recognized correctly. The pattern of the downtrend recognition curve is more complex and irregular. This curve can be divided into three or more sections. The first section, a downtrend section, is very characteristic for the downtrend recognition curve. The next sections (plateau, uptrend section, random walk section) can vary this recognition curve significantly.

Based on the behavior of these curves, we define the following recognition algorithm that incorporates iterative optimization of concept description and flexible matching with test data.

Step 1: Label the first section of each recognition curve as an uptrend or downtrend recognition pattern.

Step 2: Select the recognition curves that have the uptrend recognition pattern only.

Step 3: Make the final classification decision indicating the class for which the uptrend pattern runs through the highest recognition rates.

5. TESTING DOMAIN

5.1. Image Data

We illustrate the above methodology and results through two separate experiments. The first experiment was performed to show recognition effectiveness, while the second experiment was performed to show hard cases that can decrease the recognition power of the introduced methodology. Each experiment was carried out with four texture classes selected from six texture images (see Fig. 6) taken from the Brodatz album.[24] In the first experiment, we presented the system training sections of the

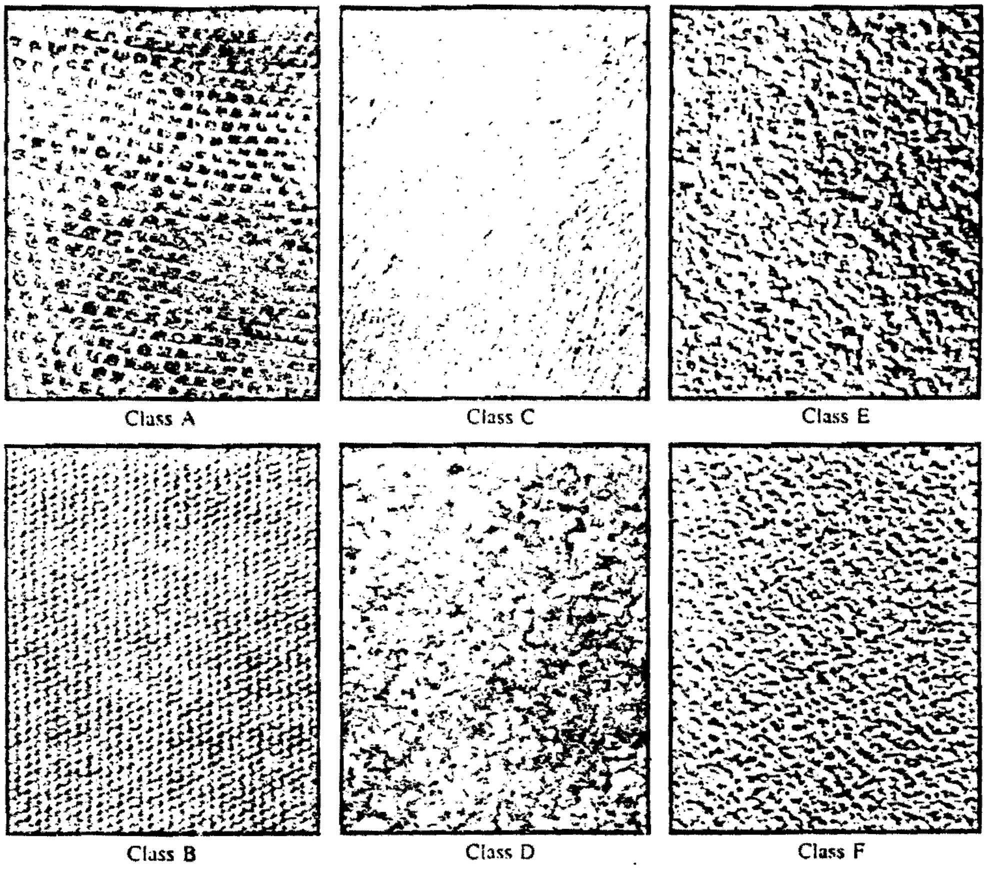


Fig. 6. Samples of texture classes used for two experiments (class A: reptile skin, class B: cotton canvas, class C: fur, class D: beach sand, class E: pressed cork, class F: handmade paper). The first experiment was done for classes A, B, C, and D, while the second experiment was for classes C, D, E, and F.

following images: reptile skin (class A), cotton canvas (class B), fur (class C), and beach sand (class D). After an analysis of the first experiment, we decided to prepare another set of four texture classes for the second experiment. Two of them, fur (class C) and beach sand (class D), are common for both experiments, while pressed cork (class E) and handmade paper (class F) replaced classes A and B.

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

5.2. Feature Extraction

As mentioned in the introduction, our goal is focused on the development and presentation of a new recognition methodology that can be effective for recognizing

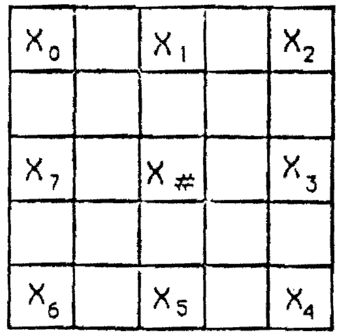


Fig. 7. Feature extraction window.

very noisy concepts. Therefore, for the purpose of this paper, we did not look for a more powerful feature extraction methodology that could be applied to code local texture characteristics. Contrary to the development of more powerful noise-reducing methods, we have applied the simplest extraction of texture features. Such an extraction preserves the negative influence of noise and the imperfection of features acquisition in the learning phase where texture class descriptions are acquired from examples. Then we manipulate the concept descriptions in order to remove noisy components; which increases system recognition effectiveness.

Texture attributes were extracted from input images in the following way. A 5 × 5 feature extraction window (Fig. 7) was applied to extract eight local texture attributes, defined as the absolute difference of a neighboring pixel and the central pixel gray values:

$$a_i = |x_{\#} - x_{i-1}| \qquad \text{for } i = 1, \ldots, 8 \tag{11}$$

where $x_{\#}$ is a gray level value of the central pixel, and x_i is a gray level value of a neighboring pixel. The event has a vector of eight attribute values $(a_1, a_2, \ldots, a_8)$ which are derived for each randomly selected pixel from the learning or testing image data.

6. EXPERIMENTAL RESULTS

6.1. Experiment 1: Recognition Effectiveness

The first experiment illustrates the behavior of recognition curves, and discussed the recognition principle and the effectiveness of the presented recognition methodology on the example of texture domain. Training image samples of four texture classes (classes A, B, C, and D from Fig. 6) were shown to the system and processed to extract texture attributes. During the learning phase, the system completed four sets of training events, each set consisting of 200 examples selected randomly from training sections of texture images. For these four sets of examples, the system acquired

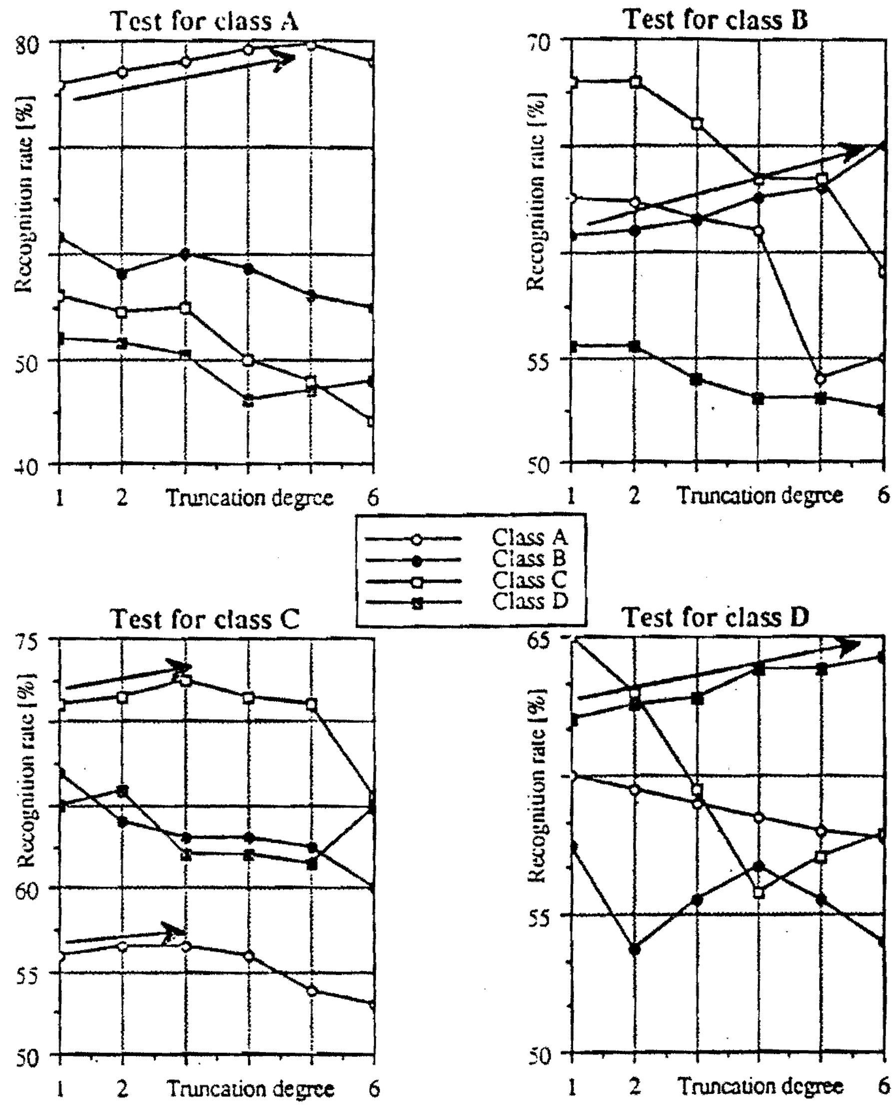


Fig. 8. Recognition curves for the first experiment, where rule description of each class was learned from 200 examples, showing uptrend sections and correct recognition of all classes (an arrow → indicates recognition curves with uptrend pattern).

texture class descriptions that incorporates the learning from examples methodology as explained in Sec. 2.2.

In the recognition phase, the system randomly selected 100 testing events for each texture class but from different sections of the texture image. The recognition process (flexible matching) was performed on four such sets of test data. Reconstructed non-smoothed recognition curves are presented for each testing dataset in Fig. 8. All uptrend sections of the recognition curves are indicated by an arrow. The truncation degree corresponds to the consecutive steps of rules optimization—all rules were optimized in each step by the truncation of a given number of less significant complexes. The recognition rate was computed as the number of test samples correctly recognized for a given class divided by the total number of test samples of a given dataset. It should be pointed out that a single testing event can be recognized for more than one class when it is covered by complexes of different classes or the distance to such complexes is the same.

Figure 8 shows that if one applies our methodology, all four classes are recognized correctly. But if one applies the traditional single match method, two texture classes (i.e. classes B and D) are not recognized.

Based on these results, let us discuss the major advantages of the new methodology following the recognition results for classes A and C versus the recognition results for classes B and D. Classes A and C are recognized correctly by the traditional single match approach without consideration of the dynamic behavior of the recognition curves. Both of these classes have a typical uptrend pattern for the recognition curve, while other curves have the downtrend pattern. Classes A and C are recognized correctly by the new method as well. The decision making process, however, performed for the test for class C, considers two uptrend recognition curves, i.e. classes A and C. The final decision points to class C because its recognition rates for the uptrend section of the recognition curve are higher than for class A.

Classes B and D are not recognized if one applies the traditional single match method. However, they are successfully recognized when applying the new methodology based on the analysis of the dynamic behavior of the recognition curve. This metholodogy can give us more effective recognition results in cases of very noisy and imperfect data/features used to learn concept descriptions.

6.2. Experiment 2: Hard Cases

The second experiment deals with the hard recognition cases that can be caused, for example, by parameters of the learning processes. These parameters, when changed, can influence the recognition effectiveness of any recognition methodology. We studied one such effect, i.e. the decrease in the number of learning examples presented to the system.

We decreased the number of learning examples from 200 to 100 for each training class. This decrease influenced the representation of texture classes in the feature space and the characteristics of acquired rule descriptions. The descriptions are less powerful in that (i) they are less complete and (ii) rule characteristics are less informative. In decreasing the number of provided examples, we observe two problems that can decrease the effectiveness of the new methodology. However, when compared with the traditional single match method the results produced are still better. These effects are illustrated in the second set of four texture classes D, C, E, and F which were chosen for the experiment. The change in the learning set of texture classes is due to the intention to illustrate these problems better.

The first negative effect that the system has to be aware of during the recognition phase is the very short uptrend section of the non-smoothed recognition curve. This

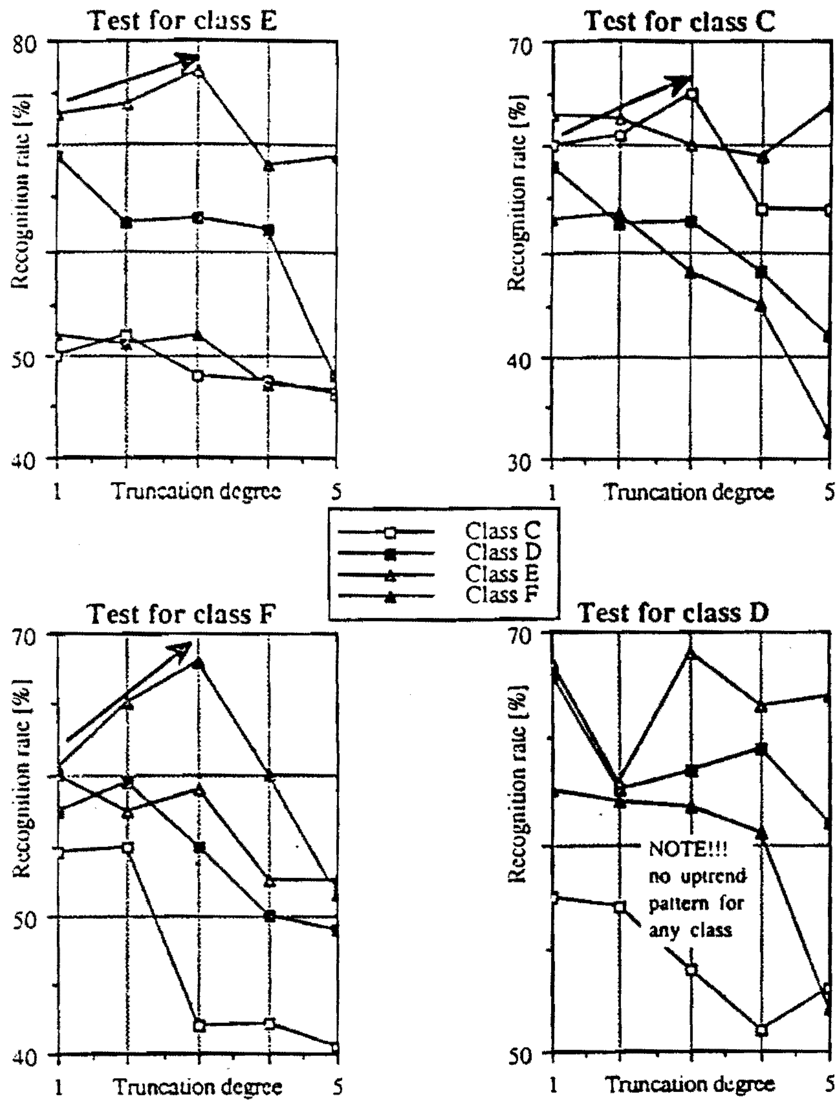


Fig. 9. Recognition curves for the second experiment, where each class was learned from 100 examples only, showing very short uptrend sections and correct recognition of classes C, E, and F. The test for class D failed due to the missed uptrend section (an arrow → indicates an uptrend pattern for the recognition curve).

uptrend cannot be detected if the system smooths recognition curves. Let us compare the uptrend sections of the recognition curves presented in Figs. 8 and 9. The uptrend sections were long (or even very long) for different experiments when we presented to the system a large number of training examples. Generally, if the number of examples decreases, the length of the uptrend section of the recognition curves decreases and is followed immediately by a very sharp downtrend section (i.e. there is no plateau). This observation indicates that future design of the truncation procedure must be improved with regard to the case of learning from very small sets of noisy/imperfect training data.

The second negative effect that can occur during the recognition phase is the possibility that there is no uptrend recognition curve. This effect is shown in Fig. 9 for the test for class D. To deal with such cases, we have redesigned the recognition procedure. A new recognition procedure decides that if there is no uptrend recognition curve for a given test then the test data cannot be classified into any class based on matched class descriptions (i.e. this is a case of "no decision"). Such modification gives still better classification decisions when one compares it to the traditional single match method which classifies the test dataset incorrectly into the class E.

7. CONCLUSIONS AND FUTURE WORK

We have presented a novel approach to the recognition of concepts, which incorporates machine learning methodology and deals with very noisy data. Our approach is based on the following elements:

(1) Acquiring concept descriptions by symbolic machine learning methodology (learning from examples) and representing them as the disjunctive normal forms.
(2) Manipulating (optimizing) concept descriptions in order to remove their less significant (noisy/imperfect) components.
(3) Flexible matching of unknown data to optimized concept descriptions.
(4) Applying closed-loop iterative repetition of optimization and matching steps, performed for different optimization degrees, in order to acquire the dynamics of the recognition curves over reduced concept descriptions.
(5) Final decision making based on the uptrend pattern of recognition curves.

The approach presented assumes that learned concept descriptions contain noisy components, and that these concept descriptions have to be optimized to gain better performance. The performance improvement of concept descriptions is reached via the elimination of some less significant noisy components. Parametric control of such optimization, however, is very difficult. The stabilization of the recognition rate is not guaranteed and a parameter value has to be found through the training process. This value can change over concept variations and time, and in consequence, it can decrease (rather than increase) the system recognition effectiveness. While the stability of the recognition decision is questionable for a single match, this stability can be improved for a sequence of matches.

We found that the classification decision can be based on the dynamic characteristics of the recognition curve acquired over increasing degree of optimization. The trend of the recognition can then be found. In this approach, the optimization and matching steps are repeated iteratively. The dynamic characteristics of recognition curves are completed and a pattern of each recognition curve is labeled. For a given set of test data, the system selects recognition curves of uptrend patterns (when the recognition rate increases with the increase in the optimization degree), and the final classification is made for the class having recognition rates which are the highest among selected uptrend recognition curves. Thus, the method classifies data based on a series of matches (rather than on a traditional single match) that determines the behavior of the recognition curve.

This approach was tested on texture recognition problems, where features were extracted by a very simple method. The results were presented to illustrate the introduced methodology, to show its effectiveness in cases where traditional approaches failed to recognize concepts correctly, and to identify hard cases for the recognition step. This methodology requires, however, further effort in its development and additional experimentation with different types of data. Our future work also includes the formalization of the methodology, the modification of the truncation algorithm, and the investigation of its effectiveness in various object recognition problems.

Developed object recognition methodology is being applied to create an intelligent vision system for outdoor navigation of autonomous robots and for remote surveillance systems.[4] Such a system has to acquire visual concepts both in the supervised and the unsupervised mode. While the supervised mode assumes a good representation of training data, the unsupervised acquisition must be based on the collection of training examples provided by the system itself. In such a case, the influence of imperfectly collected data on the final decision has a crucial role in the later recognition phase and overall system performance.

ACKNOWLEDGEMENTS

This paper is an extension and modification of a short paper presented at the IEEE Conference on AI Applications '91. The authors wish to thank Ryszard Michalski and Harry Wechsler for valuable comments, Janet Holmes, Eric Bloedorn, and Janping Zhang for technical help, and anonymous reviewers for their comments leading to the improvement of the final presentation. This research was done in the GMU Center for Artificial Intelligence. Research of the Center is supported in part by the Defence Advanced Research Projects Agency under the grant administered by the Office of Naval Research No. N00014-87-K-0874 and No. N00014-91-J-1854, in part by the Office of Naval Research under grants No. N00014-88-K-0397, No. N00014-88-K-0226 and No. N00014-91-J-1351, and in part by the National Science Foundation Grant No. IRI-9020266.

REFERENCES

1. K. S. Narendra, Ed., *Adaptive and Learning Systems*, Plenum Press, 1986.
2. O. R. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.
3. P. W. Pachowicz, "Integrating low-level features computation with inductive learning techniques for texture recognition", *Int. J. Pattern Recogn. Artif. Intell.* 4, 2 (1990) 146–165.

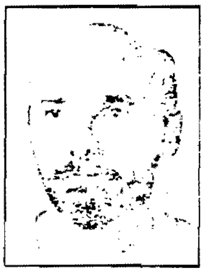
4. P. W. Pachowicz, "Learning-based architecture for robust recognition of variable texture to navigate in natural terrain", *Proc. IEEE Int. Workshop on Intelligent Robots and Systems '90*, Tsuchiura, Ibaraki, Japan, Jul. 1990.
5. P. W. Pachowicz, "Recognizing and incrementally evolving texture concepts in dynamic environments: An incremental model generalization approach", submitted to *IEEE Trans. Syst. Man Cybern.* 1991 (also Report of the Center for AI, George Mason University).
6. S. Pinker, *Visual Cognition*, MIT Press, 1985.
7. S. Chien, B. Whitehall, T. Dietterich, R. Doyle, B. Falkenhainer, J. Garrett and S. Lu, "Machine learning in engineering automation", *Proc. 8th Int. Workshop on Machine Learning*, Evanston, 1991.
8. J. Quinlan, "Induction of decision trees", *Machine Learning* **1** (1986) 81–106.
9. R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects: The local-feature-focus method", in *Robot Vision*, Ed. A. Pough, Springer-Verlag, 1983, pp. 44–81.
10. J. M. H. DuBuf, M. Kardan and M. Spann, "Texture feature performance for image segmentation", *Pattern Recogn.* **23**, 3–4 (1990) 291–309.
11. P. W. Pachowicz and J. Bala, "Advancing texture recognition through machine learning and concept optimization", submitted to *IEEE Trans. Pattern Anal. Mach. Intell.* 1991. (also Report of the Center for AI. George Mason University).
12. I. Kononenko, I. Bratko and E. Roskar, "ASSISTANT: A system for inductive learning", *Informatica* **10** (1986).
13. R. S. Michalski, "A theory and methodology of inductive learning", in *Machine Learning: An Artificial Intelligence Approach*, Tioga Publishing, 1983, pp. 83–134.
14. R. S. Michalski, I. Mozetic, J. R. Hong and N. Lavrac, "The AQ15 inductive learning system", Report No. UIUCDCS-R-1260, Department of Computer Science, University of Illinois at Urbana-Champaign, Jul. 1986.
15. J. R. Quinlan, "Simplifying decision trees", *Int. J. Man-Machine Studies* 27 (1987).
16. J. Zhang and R. S. Michalski, "Rule optimization via SG-TRUNC method", *Proc. 4th European Working Session on Learning*, Montpellier, Dec. 1989, pp. 251–262.
17. P. Clark and T. Niblett, "The CN2 induction algorithm", *Machine Learning* **1**, 3 (1989).
18. R. C. Holte, L. E. Acker and B. P. Porter, "Concept learning and the problem of small disjuncts", *Proc. 11th Joint Conf. on Artificial Intelligence*, 1989.
19. F. Bergadano, S. Matwin, R. S. Michalski and J. Zhang, "Learning two-tiered descriptions of flexible concepts: The POSEIDON system", accepted for publication in *Machine Learning* (1992).
20. R. E. Reinke, "Knowledge acquisition and refinement tools for the advice meta-expert system", ISG 84-4, Department of Computer Science, University of Illinois at Urbana-Champaign, Jul. 1984.
21. R. S. Michalski, "Two-tiered concept meaning, inferential matching and conceptual cohesiveness", in *Similarity and Analogy*, Eds. S. Vosniadou and A. Orton, Cambridge University Press, 1987.
22. 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.
23. P. W. Pachowicz, "Low-level numerical characteristics and inductive learning methodology in texture recognition", *Proc. IEEE Int. Workshop on Tools for AI*, Washington, D. C., Oct. 1989, pp. 91–98.
24. P. Brodatz, *A Photographic Album for Arts and Design*, Dover Publishing Co., 1966.

APPENDIX

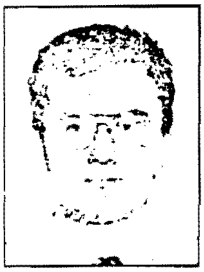
The following is an example of a class description with 27 complexes. Six truncation steps reduce this class description to 21 complexes.

Rule Description of a Class		*Characteristics*
complex 1:	$[x1 = 1 \ldots 2][x2 = 0 \ldots 2][x4 = 3]$	(total: 9, unique: 8)
complex 2:	$[x1 = 1 \ldots 5][x4 = 3 \ldots 5][x8 = 3]$	(total: 8, unique: 5)
complex 3:	$[x1 = 1 \ldots 3][x4 = 1 \ldots 5][x5 = 0 \ldots 3][x7 = 2 \ldots 3][x8 = 0 \ldots 2]$	(total: 8, unique: 3)
complex 4:	$[x1 = 3 \ldots 5][x2 = 2 \ldots 5][x3 = 0 \ldots 2][x5 = 0 \ldots 3][x7 = 1 \ldots 3]$	(total: 8, unique: 2)
complex 5:	$[x3 = 0][x4 = 1 \ldots 5][x6 = 0 \ldots 1][x7 = 2 \ldots 5][x8 = 0 \ldots 2]$	(total: 8, unique: 2)
complex 6:	$[x3 = 0 \ldots 2][x4 = 1 \ldots 5][x5 = 2 \ldots 5][x7 = 0][x8 = 1 \ldots 5]$	(total: 7, unique: 1)
complex 7:	$[x1 = 1 \ldots 5][x2 = 1][x3 = 1 \ldots 5][x6 = 0 \ldots 1]$	(total: 6, unique: 6)
complex 8:	$[x4 = 2 \ldots 5][x5 = 1 \ldots 5][x6 = 0][x7 = 0][x8 = 1 \ldots 3]$	(total: 6, unique: 3)
complex 9:	$[x2 = 0][x6 = 1 \ldots 3][x7 = 0][x8 = 2 \ldots 5]$	(total: 6, unique: 3)
complex 10:	$[x1 = 0][x2 = 1 \ldots 2][x4 = 1 \ldots 5][x6 = 0][x8 = 0]$	(total: 5, unique: 4)
complex 11:	$[x1 = 2 \ldots 3][x2 = 3][x3 = 0 \ldots 2][x6 = 0]$	(total: 5, unique: 1)
complex 12:	$[x2 = 0 \ldots 1][x3 = 1 \ldots 5][x6 = 0 \ldots 1][x7 = 1][x8 = 0 \ldots 1]$	(total: 4, unique: 2)
complex 14:	$[x3 = 0 \ldots 3][x5 = 0 \ldots 3][x6 = 1 \ldots 5][x8 = 4 \ldots 5]$	(total: 3, unique: 3)
complex 15:	$[x1 = 1 \ldots 3][x4 = 0 \ldots 1][x5 = 0 \ldots 2][x7 = 4 \ldots 5][x8 = 0 \ldots 2]$	(total: 3, unique: 3)
complex 16:	$[x1 = 2 \ldots 5][x3 = 0][x6 = 0 \ldots 1][x7 = 1]$	(total: 3, unique: 3)
complex 17:	$[x4 = 2][x6 = 1][x8 = 0]$	(total: 3, unique: 2)
complex 18:	$[x1 = 1][x4 = 1][x7 = 0][x8 = 1 \ldots 5]$	(total: 3, unique: 1)
complex 19:	$[x1 = 0 \ldots 3][x2 = 0][x3 = 1 \ldots 5][x5 = 0 \ldots 3][x6 = 1 \ldots 3]$	(total: 3, unique: 1)
complex 20:	$[x2 = 0][x4 = 0][x5 = 1 \ldots 5][x = 1]$	(total: 3, unique: 1)
complex 21:	$[x1 = 1][x2 = 1][x8 = 2 \ldots 3]$	(total: 2, unique: 2)
complex 22:	$[x2 = 0][x3 = 0 \ldots 2][x6 = 0][x7 = 1 \ldots 2][x8 = 0]$	(total: 2, unique: 2)
complex 23:	$[x2 = 0][x3 = 0][x5 = 1 \ldots 2][x7 = 1 \ldots 2]$	(total: 2, unique: 1)
complex 24:	$[x6 = 2 \ldots 5][x7 = 2][x8 = 1]$	(total: 2, unique: 1)
complex 25:	$[x1 = 1][x2 = 2][x3 = 0][x5 = 0]$	(total: 1, unique: 1)
complex 26:	$[x1 = 2 \ldots 3][x6 = 1 \ldots 3][x8 = 1]$	(total: 1, unique: 1)
complex 27:	$[x2 = 3][x3 = 1]$	(total: 1, unique: 1)

Received 30 April 1991; revised 25 November 1991



Jerzy W. Bala is a doctoral candidate in the Department of Computer Science and a research assistant in the Center for Artificial Intelligence at George Mason University. He received his M.S. in electrical engineering in 1985 from the University of Mining and Metallurgy, Cracow, Poland. His interests include machine learning, computer vision, and combining symbolical and genetic learning. His dissertation deals with the development of computer vision systems with learning capabilities and their applications to texture recognition.



Peter W. Pachowicz received the M.S. in computer and electrical engineering and the Ph.D. in computer science and engineering from the University of Mining and Metallurgy, Cracow, Poland, in 1981 and 1984, respectively. In 1984 he became a faculty member at the Institute of Control Engineering, University of Mining and Metallurgy, where he worked on fast and cheap processing of images in industrial applications. In 1986, he received the Alexander von Humboldt Research Fellowship to study self-adaptation capabilities of robot vision systems. From 1986 to 1988 he worked with the Cognition Systems Group of the Computer Science Department, University of Hamburg in West Germany. In 1989, he joined both the AI Center and the Computer Science Department of George Mason University. His research approaches are usually practically oriented. His areas of interest include intelligent autonomous systems, robot vision, adaptive systems and the application of AI. His present effort at the AI Center is related to the integration and application of high-level AI (i.e. machine learning) within an engineering domain.