GMU RESEARCH ON LEARNING IN VISION: INITIAL RESULTS

by

R. S. Michalski
J. W. Bala
P. W. Pachowicz

In Proceedings of the DARPA Image Understanding Workshop, Washington D.C., April 18-21, 1993.

Proceedings of the 1991 DARPA Image Understanding Workshop, April 18-21, Washington D.C.

GMU RESEARCH ON LEARNING IN VISION: Initial Results

R. Michalski, J. Bala and P. Pachowicz
Center for Artificial Intelligence
George Mason University
Fairfax, VA 22030

Abstract

This report covers initial research on learning in vision conducted at the GMU Center for Artificial Intelligence. The research is currently conducted by two faculty members and one research assistant. The report describes our research goals, general approach, several developed methods, and results from experiments with implemented systems. The research has been concerned primarily with the development of efficient methods for inductive learning of texture descriptions from texture samples. The following methods (and implemented systems) are briefly described: *Textral* (employing multilevel symbolic image transformations and the AQ15 inductive learning program), *PRAX* (using a "principal axes" representation of texture descriptions), AQ-NT (oriented toward learning from noisy inputs), AQ-GA (combining inductive rule learning with a genetic algorithm based rule enhancement), and Chameleon (based on "model evolution" approach).

1 Introduction

The goal of this research is to explore the applicability of machine learning methods to problems of computer vision. The underlying premise is that computer vision will ultimately need to exhibit learning capabilities in order to be fully successful.

This research was supported in part by the Defense Advanced Research Projects Agency under the grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, and the grant No. N00014-91-J-1854, administrated by the Office of Naval Research, in part by the Office of Naval Research under the grant No. N00014-91-J-1351, and in part by the National Science Foundation under the grant No. IRI-9020266.

The reasons for this view are based on the following observations:

- The world changes in unpredictable ways, therefore it is impossible, in principle, to pre-program in the vision systems all the knowledge necessary for image understanding.
- Handcrafting the knowledge needed for image understanding into computer vision systems is a difficult and time-consuming process; learning provides a fundamental vehicle for simplifying this process.
- In biological vision systems, many aspects of image perception are genetically preprogrammed, but many are learned. Similarly, computer vision systems should be able to acquire some capabilities through learning.

An important result of our initial research is a demonstration that symbolic learning methods can be successfully applied to selected problems of low-level vision, in which nonsymbolic methods have been traditionally employed. Specifically, the results obtained demonstrate that these methods have been very useful for creating descriptions of textures from their samples, obtained from the original camera-generated images.

2 General Approach

The developed approach, called "Multilevel Logical Templates" (MLT) aims at automatically determining texture class descriptions ("texture signatures") from texture samples. The basic step in this process is an iterative (multilevel) application of symbolic inductive learning to generate texture rules. These rules serve as "logical templates" that are matched against window-size samples of texture classes.

The approach was originally proposed by Michalski [1973], and initially applied using the ILLIAC III image recognition computer facilities.

The research at the GMU Center for Artificial Intelligence has developed a variety of novel extensions and new directions stemming from the above general approach. The novelty is in utilizing new types of image transformations, self-improvement of the representation space (constructive induction), advanced noise-tolerant learning techniques, and new multistrategy learning techniques.

The basic idea behind the MLT approach can be explained as follows (Figure 1). Given an image with labeled samples of different textures, the learning system determines a sequence of operators that transform this image to a "symbolic" image, in which picture elements are labels of corresponding texture areas. The sequence of operators that produces such a labeling serves as a texture description ("texture signature'). The basic operator in this process is an application of a set of logic-style rules to transformed texture samples. The rules can be applied in parallel, and serve as "logical templates" that are applied to "events" (attribute vectors) representing texture samples.

To recognize an unknown texture sample, the system matches it with all candidate texture descriptions. This is done by applying decision rules to the events in the sample. For each event, the class membership (texture class) is determined.

The assignment of the sample to a given decision class (texture) is based on determining which of the candidate classes gets the majority (or) plurality of votes. Thus, even if some events in the sample are incorrectly recognized, the classification of the sample may be correct.

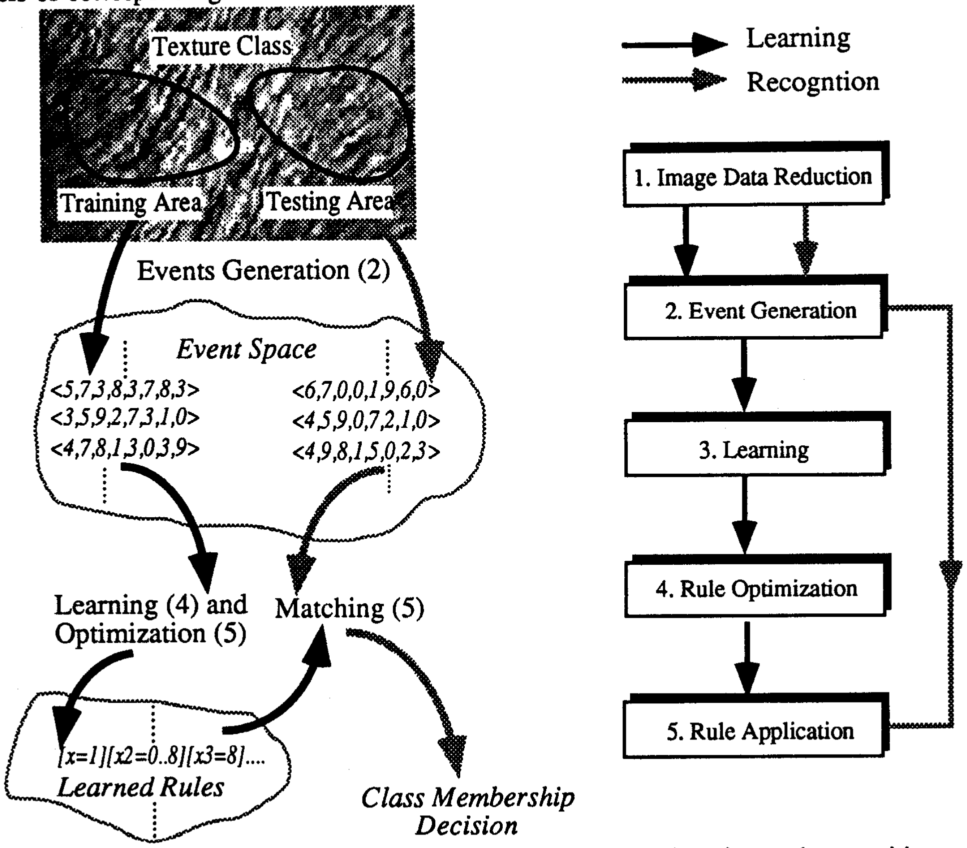


Figure 1: An illustration of the MLT approach to texture learning and recognition.

The process of learning such texture descriptions consists of the following phases (Figure 1):

1) Image preprocessing (volume reduction).
(2) Training events generation (selection of texture samples, determining attributes, and formulating training examples)
(3) Inductive learning of texture rules ("logical templates"), and
(4) Texture rule optimization.

The above process may be repeated iteratively until a desired image transformation is obtained.

The first phase of the process adapts the "image volume" to the texture classes characterized by training samples.
This is done by modifying the spatial resolution and a gray-level resolution of the image so that the similarities between samples of the same texture and dissimilarities between samples of different textures are increased. In the initial experiments, the events were extracted from the second or third level of the Gaussian pyramid. The typical resolution of camera-acquired images was 512 by 512 image elements.

The second phase extracts a set of spatial texture samples, called *events*, from classified texture regions (Module 2 in Figure 1). An event is a vector of attribute values that represent different image (texture) features. Initial attributes are predefined. Additional attributes can be determined through the process of constructive induction [Wnek and Michalski, 1991].

There are many possible attributes that could be determined to characterize textures. The most desirable are those that define a description space in which points corresponding to the same texture class constitute easily describable clusters.

The attributes generated by different systems described in this report fall into one of three categories: neighboring gray-level values, statistical measurements, and convolution filter outputs. Sets of events extracted from texture classes to be learned are used as training examples.

Texture rules are determined using the AQ-15 method for inductive concept learning from examples ([Michalski, 1986]). The rules learned by the AQ method are represented in VL_1 (Variable-Valued Logic System 1); [Michalski, 1972]). Advantages of this representation are that it is amenable for parallel execution, and easy to interpret conceptually.

In the machine learning method described here,, a concept corresponds to a single texture class. A concept description is a logical expression in disjunctive normal form associated with a decision class (here, a texture class).

Each conjunction in this expression together with the associated decision class can be viewed as a single decision rule.

The conjunctions (serving as condition parts of the rules) are logical products of elementary conditions in the form:

[L # R]

where:

L, called the *referee*, denotes an attribute.
R, called the *referent*, is a subset of values from the domain of the attribute L.
# is one of the following relational symbols: =, <, >, >=, <=, <>.
Each rule is assigned two parameters: "t" (for "total weight")—measuring the total number of positive training examples covered by the rule, and "u" (for "unique weight")—measuring the number of positive examples covered by the given rule and not covered by any other rule for the given decision class.

Here is an example of an AQ-15 decision rule:
[Class=1]⇐[x2=1][x4>3][x6=1..7]: (t=6, u=2)
This rule covers 6 examples of Class 1, out of which 2 are covered only by this rule, and not by any other rule for this class. In the case of texture rules, x_i are attributes characterizing a texture sample (in our experiments we used primarily 8x8 windows). The above rule is satisfied, if attribute x2 takes value 1, attribute x4 has value greater than 3, and attribute x6 takes value between 1 and 7.

As mentioned earlier, a description of a texture class can be viewed a set of such rules (a "ruleset"). In such a ruleset, individual rules are ordered according to the decreasing values of the t-weight. The following is an example of a texture description:

[Texture class = sweater *surface*] ⇐
[x1=7,9,12]& [x2>]&[x3=0..4]&[x4=0..5]&[x5=0..3]
[x6=0..7] [x7=2..4] [x8=0..3] (t:28, u:21)
OR
[x1=5,7,9] [x2>2] [x3=0..2] [x4=0..4] [x5=1..4]
[x6=0..6] [x7=0..2] [x8=0..4] (t:27, u:20)
OR
[x1=2,5,7]&[x2=1..12]&[x3=1..2]
[x4=2..6]&[x5=3..4]&[x6=0..4]&[x7=1..3,5,7]
[x8=0..1,3..4] (t:16, u:11)
OR
[x1=5..14]&[x2>6]&[x3=0..2,4..5]&[x4=2..5] (t:5, u:3)

where x1 is the Laplacian edge operator, x2 is the Frequency spot, x3 is the horizontal edge operator, x4 is the vertical edge operator, x5 is the horizontal V-shape operator, x6 is the vertical V-shape operator, x7 is the vertical line operator, and x8 is the horizontal line operator.

The method uses "truncated" descriptions of texture classes. A truncated description is obtained by the removing from the initially generated rules the ones with a very low t-weight. The reason for this is that rules with a low t-weight can be viewed as insignificant, or as representing noise.

We have discovered experimentally that so truncated descriptions often give a higher texture recognition performance than non-truncated descriptions. Since truncated descriptions are also simpler, then such a truncation process is highly desirable. A detailed study of this phenomenon (in the context of non-vision applications) have been described in [Bergadano et al., 1992].

The learned texture descriptions are generalizations of the observed texture events (i.e., attribute-value vectors characterizing window-size texture samples). Therefore, they can be used to classify unobserved texture samples. There are two methods for applying the descriptions for recognizing the class membership of an event: the *strict* match and the *flexible* match.

In the strict match, the system tests whether an event strictly satisfies (the condition part of) a rule. The satisfied rule determines the classification decision. In the flexible match, the system computes a degree of match between the event and candidate rules. The degree of match can vary in the range from 0. (no match) to 1.0 (complete match). The rule with the highest degree of match determines the classification decision.

To explain the calculation of the degree of match, assume that a recognition rule contains a condition $[x = a_k]$. If the domain of the attribute x is a set of numerical values $<a_1,a_2, ...,a_n>$, and an event includes the statement $[x=a_i]$, the normalized degree of match between the rule and the condition in the event is defined:

$$1 - (| a_j - a_k | / n)$$

If the condition has several values in the referent (on its right-hand-side), the value closest to a_k is used. The degree of match between a rule containing several conditions and an event was computed as the average of the degrees of match between the conditions and the event conditions.

The degree of match between a class description (which may have several rules) and a given testing event (an example) is determined as the maximum of the degrees of match between individual rules in the description and the event. The description with the highest match among classes determines the recognition decision. The measure of recognition accuracy of a rule when applied to a set of testing events is the percentage of the number of correctly classified test events to the total number of testing events in the set.

3 Implemented Systems

3.1 Learning Texture Signatures: TEXTRAL

The TEXTRAL system implements a version of the MLT approach ("Multiple Logical Templates"). The system generates multiple level of descriptions (rulesets) by applying the same learning process to images generated at each level. The first level ruleset relates to the original camera-acquired image. The next level ruleset relates to a "symbolic image" that consists of numerical labels associated with the texture classes. These labels are generated by the application of the first level ruleset, and represent texture classes assigned to texture events in the original image [Bala and Michalski, 1991]. Subsequent levels of rulesets are generated by reapplying this same process to the symbolic images generated at the previous step..

Here is a more detailed description of the algorithm:

• Step 1 extracts a random set of training events from the training areas in the original images by applying various local operators (such as Law masks, statistical measures, convolution operators, etc.), and learns the "first-level" texture of rules;

• Step 2 determines rulesets generalizing the training events. These rulesets are applied to the training areas of the original image, and a new image (a "symbolic image") is created. The pixels of the new image (the next level image) are numerical labels of texture classes assigned by the ruleset to corresponding events in the original (previous level) image.

• Step 3 determines the match between the texture training areas labeled by the teacher and the corresponding areas in the symbolic image. If the match is sufficiently high (or the system reaches a designated number of levels) then the process stops. Otherwise, the control is passed to the step 1. The events are extracted from the symbolic image (the last level image) and assigned classes corresponding to the training assignment of pixels in the original image (i.e., representing the "correct" partitioning of the image into texture classes done by the teacher).

We have performed a number of experiments with the system for various numbers of texture classes (between 4 and 16), representing fined-grained textures, such as sand, paper, pebbles, etc. Training events were determined from texture samplings using 8x8 windows, and selected from texture training areas. The texture training and testing areas for each texture class was determined by a teacher.

Table 1 and 2 show the confusion matrices characterizing the system's recognition rates (in %) for individual texture events (using 8x8 windows) selected from testing areas of four texture classes, C1, C2, C3 and C4. Table 1 shows the recognition rate for first level rules, and Table 2—for the second level rules. Recall that the conditions of the second level rules apply not to properties of the original image, but to the distribution of texture labels generated by the first level rules.

	Recognized texture class			
Correct Class	C 1	C 2	C3	C4
C 1	***84***	*15*	*16*	*23*
C 2	*10*	***78***	*20*	*10*
C 3	*7*	*14*	***79***	*27*
C 4	*26*	*17*	*27*	***67***

Recognition rates using the first level rules.
Table 1.

	Recognized texture class			
Correct Class	C 1	C 2	C 3	C 4
C 1	***94***	*3*	*2*	*6*
C 2	*4*	***96***	*4*	*4*
C 3	*4*	*9*	***88***	*12*
C 4	*13*	*7*	*12*	***80***

Recognition rates using the second level rules.
Table 2.

The average correct recognition rate of individual events for the 4 class experiment was 77% when using the first level rules, and 89.5% when using the second level rules. At the same time, the average misclassification rate decreased from 17.6% to 6.6%, respectively. Thus, the experiment has demonstrated that multilevel learning (using higher level rules) can increase the system's recognition of individual events.

It should be clearly noted, however, that to recognize a given *sample* of a texture, one would extract from the unknown texture not just single event (representing an 8x8 window), but also several neighboring events. In such a case, the texture identification decision will be based on the majority of class assignments of individual events in the neighborhood.

Therefore, *even when there is a relatively low recognition rate of individual events, one can achieve 100% recognition rate of the sample* (it is sufficient that the plurality of events in the sample are recognized correctly). A problem may occur mainly when a sample is taken from a border area between different textures, or includes events characterizing rare local texture distortions.

Figure 2 presents the recognition rate of *individual events* on learning of 12 textures, using rules of level 1, 2, 3 and 4.

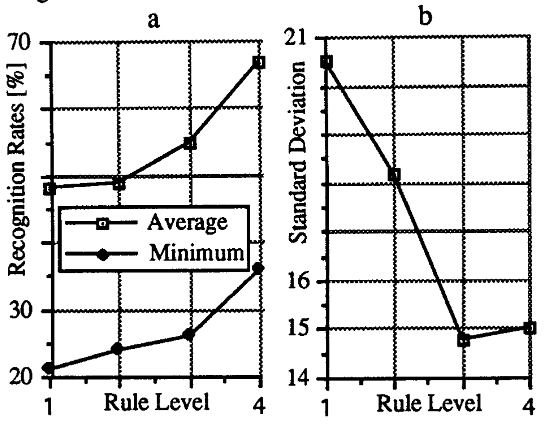


Figure 2: An increase of the system performance with the rule level.

Figure 2A shows the increase of the recognition rate of individual events from 12 texture classes with the rule level. Figure 2B shows the corresponding decrease of the standard deviation (in %) of the recognition rates with the rule level. The average recognition rate of individual events increased from 48% with level one rules to 58% with level four rules. At the same time, standard deviation (of correct recognition values) decreased from above 20 to 15, respectively. The minimum recognition rate increased from 21% to 36%.

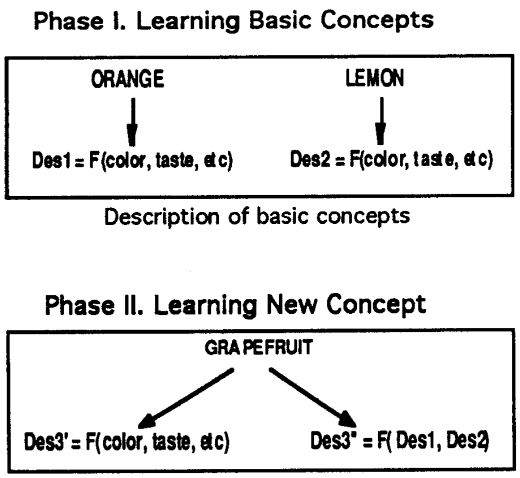
The TEXTRAL method represents an extension and improvement of our earlier method implemented in the TEXPERT system [Channic, 1989]. TEXPERT used our earlier inductive learning system GEM ("Generalization of Examples by Machine;" also called AQ14) [Reinke, 1984]. TEXPERT was applied to the problem of recognizing faults in laminated aircraft materials using ultra-sound images

3.2 Learning Large Number of Classes: PRAX

In the TEXTRAL system, each texture class is represented by a ruleset [Bala et al., 1992]. If there are very many texture classes, there will be correspondingly many rulesets, and the learning and recognition process may become complex. The PRAX system represents an alternative approach to the problem of learning a large number of concept descriptions (in our application, texture classes).

The basic idea is to designate some concepts to be basic, and describe the remaining concepts in terms of the relations to the basic concepts. This idea can be simply illustrated by the example in Figure 3.

If the system already knows the concept of "orange" (Des1) and "lemon" (Des2), then it can learn the concept of "grapefruit" by relating properties of the grapefruit to those of the lemon and the orange (Des3"), rather than in terms of original properties (Des3').



Phase I. Learning Basic Concepts
ORANGE
LEMON
Des1 = F(color, taste, etc)
Des2 = F(color, taste, etc)
Description of basic concepts
Phase II. Learning New Concept
GRAPEFRUIT
Des3' = F(color, taste, etc)
Des3" = F(Des1, Des2)

Figure 3: A simple illustration of the PRAX method.

In the PRAX method, descriptions of the basic concepts are called "principal axes." They are learned in the similar way as in the TEXTRAL system.

To learn a new, non-basic concept, the system determines a *similarity matrix* (SM) for that concept. The SM specifies the average degrees of similarity between the training examples of the new concept and all the principal axes.

The degree of similarity between an event and each principal axis is determined according to a procedure called ATEST [Michalski, et al, 1986]. The procedure determines the accumulated difference between the attribute values in the event and the conditions in each rule in the principal axis. To obtain a uniform representation of all class descriptions, the similarity matrix is also computed for all basic concepts.

These degrees of similarity can be viewed as values of the new constructed attributes. Thus, this method represents a special case of constructive induction. (The general concept of constructive induction includes any method that self-modifies the concept representation space during the induction process. Generating additional, problem oriented attributes is an important form of such self-modification of the representation space [Michalski, 1978; Wnek & Michalski, 1991]).

To recognize an unclassified event, the method creates an SM for it, that is, determines a matrix of similarities between the event and the principal axes. Subsequently, the system determines the best match between the SM of that event and SMs of all candidate concepts. The best match indicates the class membership.

The method was empirically evaluated by applying it to the problem of learning 24 texture classes from examples (Table 3). Each example was described in terms of eight multivalued attributes (representing detectors of various basic geometrical concepts, such as the presence of lines, edges, V-shapes, etc.). The performance of the PRAX-derived descriptions was compared with the performance of the k-NN classifier. Different level of misclassification noise were added to test the robustness of the method.

The main strength of the method lies in a problem-relevant transformation of the descriptor space. The new descriptors form generalized sub-spaces of the initial, training space. In addition, the method uses a non-linear distance metric to calculate values of constructed attributes. The distance metric based on the idea of flexible matching is less sensitive to noise, then traditional Euclidean distance metric often used by pattern recognition methods.

METHOD	The Recognition Rate (in %) of Examples from Unknown Texture
PRAX	
No Noise	100%
5% Noise	100%
10% Noise	100%
K-NN	
No Noise	96%
5% Noise	92%
10% Noise	87%

Table 3: The results from comparing PRAX with the K-NN method.

The current problem with the method is that it does not have a mechanism for deciding how to choose basic concepts. Choosing the minimal subset of concepts to be used for principal axes generation is important for method to be efficient. This problem will be a subject of future research. Another weakness is that the similarity matrix is a relatively complex representation.

3.3 Learning From Noisy Data: AQ-NT

The AQ-NT method represents a novel way of handling problems of learning from noisy real-world data [Pachowicz and Bala, 1991]. It is based on the idea that events covered by rules with a low t-weight may be representing noise in the data. The assumption is that the system learning from a dataset that does not contain such events has a greater chance to produce correct concept descriptions than when learning from the original events.

The process of learning concept descriptions (in the form of a ruleset) is done in the following two phases:

Phase 1: Performs a rule-based "filtration" of the noise from the training data. This is done in the following way:

1. Induce decision rules from a given dataset using the AQ learning program.
2. Truncate concept descriptions by removing "the least significant" rules, defined as rules that cover only a small portion of the training data (have small t-weight relative to the t-weight of other rules).
3. Create a new training dataset that includes only training examples covered by the modified concept descriptions.
4. If the size of the dataset falls below an assumed percentage of the training data (which reflects an assumed error rate in the data), then go to Phase 2. Otherwise, return to step 1.

Phase 2: Acquire concept descriptions from the reduced training dataset using the AQ learning program.

Figure 4 shows results from one run of the AQ-NT system for six texture classes.

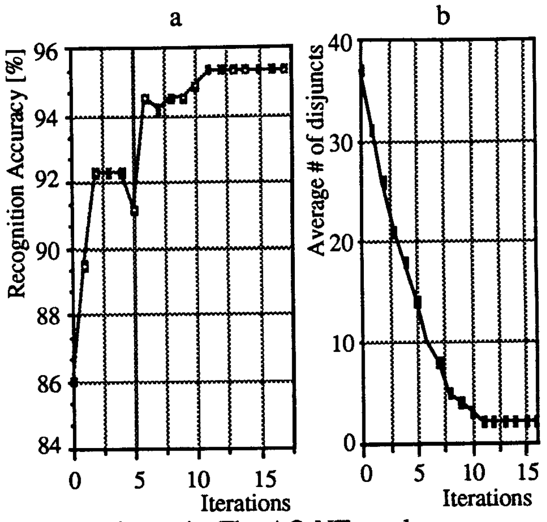


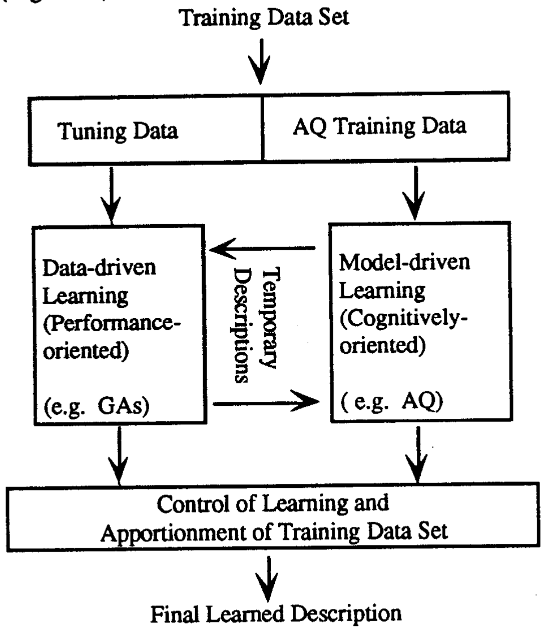
Figure 4: The AQ-NT results.

Figure 4A shows the increase of the recognition accuracy (in %) of individual events with the number of iterations. After 12 iterations, the recognition accuracy reached 95.3%. Figure 4B shows the average number of rules for each iteration. An average number of rules can be viewed as a measure of description complexity. Figure 4B shows a significant decrease in the average number of rules (from 37 to 3). This result is a significant indication of the advantages of the proposed approach.

3.4 Rule Improvement by Genetic Algorithm: AQ-GA

The size, complexity, variability and an inherent noise in the vision data pose significant difficulties in developing a reliable concept learning system. The AQ-GA multistrategy system was developed to address some of these issues [Bala et. al., 1993]. This system integrates two forms of learning, symbolic inductive generalization and genetic algorithm based learning. The integration is done in a closed-loop fashion in order to achieve robust concept learning capabilities.

The learning process cycles through two phases (Figure 5).



Training Data Set
Tuning Data
AQ Training Data
Data-driven Learning (Performance-oriented)
(e.g. GAs)
Temporary Descriptions
Model-driven Learning (Cognitively-oriented)
(e.g. AQ)
Control of Learning and Apportionment of Training Data Set
Final Learned Description

Figure 5: The AQ-GA architecture.

In the first phase, initial concept descriptions are acquired by running a noise-tolerant extension of the AQ15 rule induction system. The resulting concept descriptions may not be, optimal from the performance viewpoint, due to the AQ bias to generate simple, cognitively-oriented descriptions. Therefore, in the second phase, the system attempts to improve the performance of the descriptions by employing a genetic algorithm (GA).

The descriptions obtained from AQ15 are semi-randomly modified, using basic genetic operators: mutation and crossover. The resulting descriptions are evaluated according to a performance criterion. The criterion was the recognition accuracy of the descriptions on the "tuning" data (a subset of the training set of events). The best performing descriptions are selected from the population, and a new generation is repeated. The process stops when a desirable performance level is achieved, or the number of generations exceeds some limit.

The effectiveness of this multistrategy approach was tested on several texture recognition problems.

Genetic algorithms typically represent individuals in a population (here, concept descriptions), using fixed-length binary strings. However, if the effective cooperative learning A novelty of this method is that it uses, instead of binary strings, concept descriptions (formally, VL_1 expressions) produced by AQ15. To this end, a special mutation operator was designed to introduce small changes to selected condition parts of the rules in each concept description. The condition parts are selected by randomly generating two pointers: the first selects a rule, and the second one selects a condition in this rule.

The most-left or the most-right values of the referent in this condition are slightly modified.. For example, the condition [x1= 10..23] might be mutated to any of the following: [x1 = 10..20], [x1 = 10..24], [x1 = 12..23] or [x1 = 8..23], as well as others. Such a mutation process samples the space of possible concept description boundaries to improve the performance criteria. The mutation process can be viewed as equivalent various *transmutations* (knowledge transformations; Michalski, 1993) of the conditional part of a rule.:

• specialization: [x5 = 3, 10..23]⇒ [x5 = 3, 10..20]
• generalization: [x5 = 3, 10..23]⇒ [x5 = 3, 10..24]
• variation: [x5 = 3, 10..23]⇒ [x5 = 5, 10..23]

The crossover operation is performed by splitting concept description into two parts, upper rules and lower rules. These parts are exchanged between parent concept descriptions to produce new child concept descriptions. Since the degree of match of a given tuning event depends on the degree of match of this event to each rule of concept description, this exchange process enables inheritance of information about strong rules (strongly matching) in the individuals of the next evolved population. An example of crossover applied to short, four rules description is depicted below:

Parent description 1
1 [x1=7..8] [x2=8..19] [x3=8..13] [x5=4..54]
2 [x1=15..54] [x3=11..14] [x6=0..9] [x7=0..11]
--------------crossover position ------------------
3 [x1=9..18] [x3=16..21] [x4=9..10]
4 [x1=10..14] [x3=13..16] [x4=14..54]

Parent description 2
1 *[x1=16..54] [x5=0..6] [x7=5..12]*
2 *[x1=8..25] [x3=8..13][x4=9..11] [x5=0..3]*
------------crossover position -------------------
3 *[x4=0..22] [x5=8..9] [x6=0..7] [x7=11..48]*
4 *[x2=5..8] [x3=7..8] [x4=8..11] [x5=0..3]*

The result of the crossover operation (one of two child descriptions) is the following:

1	[x1=7..8] [x2=8..19] [x3=8..13] [x5=4..54]
2	[x1=15..54] [x3=11..14] [x6=0..9] [x7=0..11]
3	*[x4=0..22] [x5=8..9] [x6=0..7] [x7=11..48]*
4	*[x2=5..8] [x3=7..8] [x4=8..11] [x5=0..3]*

The performed experiments, involving learning rules for describing texture classes, demonstrated that the classification results obtained with the hybrid learning algorithm (Figure 5) (AQ Training Data -> AQ and Tuning Data -> GAs) exceed the performance of the AQ algorithm used alone (AQ Training Data + Tuning Data -> AQ). Figure 6 presents recognition rates from one of the experiments.

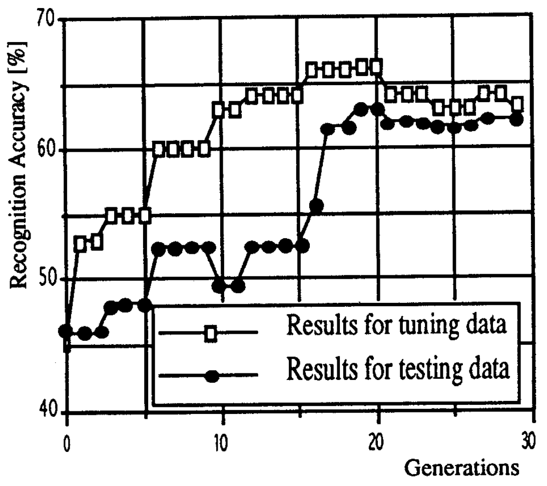


Figure 6: Recognition accuracy during the GA generations.

The result shows the recognition rates for the class that was optimized by the GA. White marks on the diagrams represent results obtained for tuning data and black marks represent recognition results for testing data.

3.5 Texture Recognition Under Varying Perceptual Conditions: Chameleon

In recognizing natural objects outdoors we have to deal with a great variability of perceptual conditions that influence changes in object visual characteristics. Most vision research on object recognition in outdoor environments, however, has been focused on recognizing objects under stationary conditions rather than dynamic conditions (varying resolution, lighting, pose, and state). Object models, particularly texture models, when learned under a given conditions are not effective in recognizing objects under different perceptual conditions.

To develop robust object recognition systems, we have to implement system adaptation capabilities that can mitigate influence of changing perceptual conditions on the effectiveness of object recognition. Our ultimate goal is to integrate learning and vision modules in such a way that learning functions can support adaptation functions of the vision system.

The variability of texture characteristics under changing resolution, lighting and pose has been investigated. It was found that texture attribute distribution (for different attribute extraction methods) can vary significantly when these conditions are changed. The shape of the distribution often contains a multimodality of texture characteristics. In most cases studied, the distribution cannot be determined *a-priori*. We also observe that the variability of perceptual conditions causes a significant translation of the attribute distribution within the attribute space.

Relatively little has been done on the application of machine learning methods to the adaptability of vision systems to the dynamic environment. Bhanu et al. [1989, 1990] apply genetic algorithms to image segmentation problems with an extension towards segmentation under variable perceptual conditions.

The variability of object appearance (particularly texture characteristics) requires the development of system capabilities that will dynamically reconfigure and update object models (knowledge). In our approach [Pachowicz, 1991], system adaptation is applied to recognize objects on images acquired over time.

In order to recognize an object on images sequentially, the system has to iteratively update the object model with regard to changes in object characteristics. In this approach, a time sequence of images monitors slight changes in resolution, lighting and surface positioning from one image to the next one -- a sequence of images is affected by continuous changes in perceptual conditions.

We integrate the learning and recognition processes within a closed loop to update texture models. Analysis of system recognition effectiveness, performed over a sequence of images, detects changes in textures. If this effectiveness decreases then the system activates incremental learning processes of model modification to improve the model discriminating power. The system learns initial texture models from teacher-provided examples. Then, the system updates these descriptions automatically without teacher help.

Two systems were developed: CHAMELEON '91 and CHAMELEON '92. The first system had only some of the adaptability functions implemented [Pachowicz et al., 1992, Pachowicz, 1992]. In this system, a teacher segments each image in a sequence. The system was useful for investigating stability problems and for the modification of object models performed on-line. The second system is more autonomous, and needs much less help from a human operator. The underlying methodology, system architecture, and experimental results are presented in a separate paper in the Proceedings [Pachowicz, 1993].

4 Summary

We have presented a general approach, called Multilevel Logical Templates (MLT), and several implemented systems for inductive learning descriptions of texture classes from texture samples. These systems represent different variations and extensions of the general approach oriented toward various types of learning problems:

- TEXTRAL —for multilevel learning to maximally improve the recognition accuracy of new textures,
- AQ-NT—for learning from data with noise,
- PRAX — for learning from large numbers of texture classes,
- AQ-GA — for automatic tuning and enhancement of concept descriptions obtained by a standard AQ inductive learning method,
- Chameleon — for model evolution under changing perceptual conditions.

The systems have been tested on several texture recognition problems, and the results were very encouraging. The recognition rates for even relatively high number textures (36) were frequently near 100%

The developed methodology is quite general and can be potentially applied not only to the problem of learning of the texture descriptions, but also to other types of problems in vision.

There are several limitations of the current methods. We have not investigated issues of the robustness and the sensitivity of the methods to various invariant texture transformations (e.g., a significant changes in the illumination). Also, it is unclear how the performance of the methods depends on the number of texture classes.

There are several other major topics to be investigated in future research:

(i) the enhancements to the current learning methodology to include capabilities for automatically generating higher level problem-relevant attributers (constructive induction)

(ii) the applicability of multistrategy learning (e.g., combining symbolic rule learning with neural network learning; the issue of representing and learning of imprecisely defined visual concepts).

(iii) the extensions of the methodology to other problems in vision, e.g., learning of shape classes.

(iv) learning new visual concepts in terms of differences and similarities form known concepts, and developing a calculus for representing symbolic differences between visual concepts.

Acknowledgments

The GMU research on machine learning and vision is conducted in collaboration with the Center for Automation Research at the University of Maryland.

Authors wish to thank Eric Bloedorn for his contribution to the development of the PRAX system, and for constructive comments about the paper. They also thank Ken De Jong for his collaboration in the development the AQ-GA system.

References

Bala, J., DeJong K., and Pachowicz P., "Multistrategy Learning From Engineering Data by Integrating Inductive Generalization and Genetic Algorithms," in *Machine Learning: A Multistrategy Approach Volume IV*, R.S. Michalski and G. Tecuci, (Eds.), Morgan Kaufman, San Mateo CA., 1993 (in press).

Bala, J. and Pachowicz P., "Recognizing Noisy Pattern Via Iterative Optimization and Matching of Their Rule Description," *International Journal on Pattern Recognition and Artificial Intelligence* , vol. 9, issue 4 , pp. 513-538. 1992.
Bala, J., Michalski R.S., and Wnek J., "The Principal Axes Method for Constructive Induction," *Proc. of the 9th International Conference on Machine Learning*, Aberdeen, Scotland, July 1992.

Bala, J. and Michalski R., "Recognition of Textural Concepts Through Multilevel Symbolic Transformations," *Proc. of the 3rd International Conference on Tools for Artificial Intelligence*, San Jose C.A., Nov. 1991.

Bergadano, F., Matwin, S., Michalski, R. S., & Zhang, J., "Learning Two-tiered Descriptions of Flexible Concepts: The POSEIDON System,". *Machine Learning*, No. 8, 5-43, 1992.

Bhanu, B., S. Lee and J. Ming, "Adaptive Image Segmentation Using a Genetic Algorithm," *Proc. of Image Understanding Workshop*, Palo Alto, CA, pp.1043-1055, 1989.

Bhanu, B., S. Lee and J. Ming, "Self-Optimizing Control System for Adaptive Image Segmentation," *Proc. of Image Understanding Workshop*, Pittsburgh, PA, pp.583-596, 1990.

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

Michalski, R. S., "A Variable-Valued Logic System as Applied to Picture Description and Recognition," in Graphic Languages, Nake, F. and Rosenfield, A. [Eds.], North Holland, 1972.

Michalski R. S., "AQVAL/1--Computer Implementation of a Variable-Valued Logic System VL1 and Examples of Its Application to Pattern Recognition," in *Proc. of the 1st International Joint Conference on Pattern Recognition*, October 30, 1973, Washington DC.

Michalski, R.S., "Pattern Recognition as Knowledge-Guided Computer Induction," Report No. 927, Department of Computer Science, University of Illinois, Urbana, June 1978. (Subsequently published under the title "Pattern Recognition as Rule-Guided Inductive Inference," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 4, pp. 349-361, July 1980.)

Michalski, R. S., "Inferential Theory of Learning: Developing Foundations for Multistrategy Learning," in *Machine Learning: A Multistrategy Approach Volume 4*, R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufman Publishers, San Mateo, CA, 1993.

Michalski R.S, Mozetic, I., Hong, J., and Lavrac, N., "The multipurpose incremental learning system AQ15 and its testing application to three medical domains," *Proc. of the 5th National Conference on Artificial Intelligence*, 1986.

Pachowicz, P.W., "Invariant Object Recognition: A Model Evolution Approach," in these Proceedings, 1993.

Pachowicz, P.W., "A Learning-Based Evolution of Concept Descriptions for an Adaptive Object Recognition," *Proc. of the IEEE of Int. Conf. Tools with AI*, Arlington, pp.316-323, 1992.

Pachowicz, P.W., "Recognizing and Incrementally Evolving Texture Concepts in Dynamic Environments: An incremental model generalization approach," Report MLI-91-10, Artificial Intelligence Center, George Mason University, 1991.

Pachowicz, P. and Bala J., "Improving Recognition Effectiveness of Noisy Texture Concepts Through Optimization of Their Descriptions", *Proceedings of the 8th International Workshop on Machine Learning*, Evanston, Ill, June, 1991.

Pachowicz, P.W., M. Hieb and P. Mohta, "A Learning-Based Incremental Model Evolution for Invariant Object Recognition," *Proc. of the 6th International Conference on System Research, Informatics and Cybernetics*, Baden-Baden, August 1992.

Reinke, R.E., "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.

Wnek J. and Michalski, R.S., "Hypothesis-driven Constructive Induction in AQ17: A Method and Experiments," *Proc. of the IJCAI-91 Workshop on Evaluating and Changing Representations in Machine Learning*, K.Morik, F.Bergadano and W.Buntine (Eds.), pp.13-22, Sydney, Australia, 1991.