P92-23

Proc. of the 1992 IEEE
Int. Conf. on Tools with AI
Arlington, VA, Nov. 1992

A Learning-Based Evolution of Concept Descriptions for an Adaptive Object Recognition

Peter W. Pachowicz

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

Abstract

A new approach is presented to the invariant recognition of objects under dynamic perceptual conditions. In this approach, images of a sequence are used to adapt object descriptions to perceived on-line variabilities of object characteristics. This adaptation is made possible by the close-loop integration of recognition processes of computer vision together with an incremental machine learning processes. Experiments presented in this paper were run for the texture recognition problem and were limited to a partially-supervised evolution of concept descriptions (models) rather than utilizing a fully autonomous model evolution. Obtained results are evaluated using the criteria of system recognition effectiveness and recognition stability.

1: Introduction

In the area of object recognition in outdoor environments, most vision research has focused on recognizing objects through textures under stationary conditions, i.e., for stable lighting conditions, resolution and surface positioning [11]. Relatively little has been done on the problem of recognizing textures under dynamic conditions. The problem of recognizing textures under dynamic conditions occurs in most situations and is therefore of significant practical importance. For example, when an autonomous system is moving through an environment, the vision system has to adapt to changes in perceptual conditions. This adaptation to variable perceptual conditions must be applied to recognize objects on images acquired over time. In order to recognize an object on images of a sequence, the system has to iteratively update its texture descriptions (models) with regard to changes in object characteristics previously perceived.

Circumstances that cause external conditions to vary can be divided into the following two groups: (1) projection variability (e.g., changing resolution, different surface orientation) and (2) the influence of natural agents (e.g., changing illumination). Changing resolution causes both the repetition rate of texture patterns and the pattern characteristics to vary. The variability of texture features obtained for different resolutions is significant [10,12]. Changing surface orientation influences the projection of 3-D microstructure of a surface onto a 2-D image. Changing lighting is caused by the dynamics of the light source (e.g., intensity, light spectrum), its position in relation to an object's surface, and irregularities such as shadows or highlights.

We propose a novel approach to the object recognition problem that deals with the dynamics of conditions under which objects are perceived. The proposed approach integrates learning and recognition processes within a closed loop to update object models. Analysis of system recognition effectiveness, performed over a sequence of images, detects changes in object characteristics. If this effectiveness decreases then the system activates incremental learning processes of model evolution to improve the model discriminating power. The system learns initial texture descriptions from teacher-provided examples. Then, the system updates these descriptions automatically [6].

Related research work has been reported by Goldfarb [3]. Just as model evolution integrates multi-disciplinary research, Goldfarb introduces "Pattern Learning" to integrate symbol formation and recognition processes with Artificial Intelligence symbol manipulation processes. Goldfarb proposes a neural net approach for recognition and evolution of structural patterns. Since no experiments with real objects are presented, the dynamic performance of his system is still questionable.

2: System Architecture

The architecture of an adaptive texture recognition system is presented in Figure 1 integrates computer vision module (CV) with concept acquisition and evolution module (CA&E).

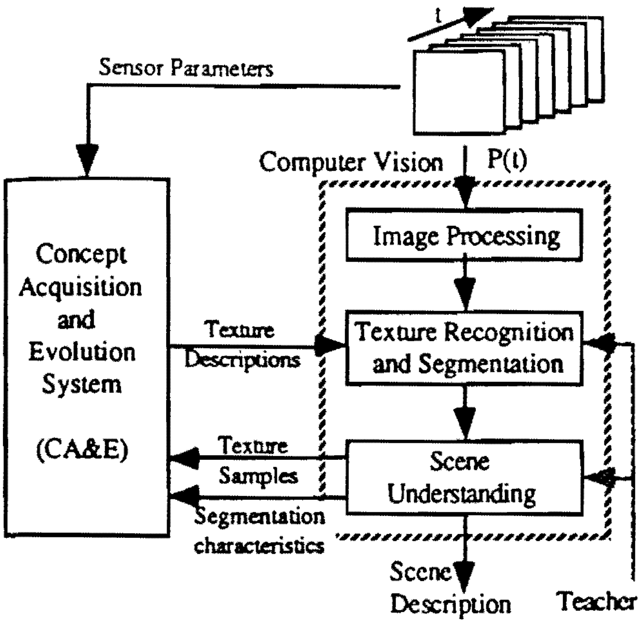


Fig.1 System architecture

The input to the vision system is a sequence of images P(t) obtained over time. These images represent differences in perceptual conditions. An incoming image is processed to extract texture attributes. Texture concept descriptions (models) are then applied to recognize extracted vectors of attributes. Segmentation processes unify homogeneous texture areas of the same class. Scene understanding, formulates a scene description and also evaluates recognition and segmentation processes using both the preceding and current scene descriptions.

In the initial training phase, a teacher has to interactively extract texture samples from the first image of a sequence. These samples (attribute vectors) are then provided to the CA&E system along with texture class labels to learn texture descriptions. Next, the CV system applies texture descriptions to recognize and segment textures on the next images of a sequence. The scene understanding module analyzes segmented texture areas along with system recognition effectiveness and compares them with the results obtained on the preceding image(s). At the same time, selected texture samples are provided to the CA&E system. The CA&E system monitors the recognition effectiveness of the CV system (i.e., the discriminatory power of texture concept descriptions). The decrease in texture recognition effectiveness carries information about changes in the external perceptual conditions and is used to activate the concept evolution processes.

3: Iterative Evolution of Texture Models

The close-loop cooperation between CV and CA&E systems supports the flow of information necessary for the evolution of texture descriptions (models) over time. This evolution is by *learning from the environment* and it is performed in an incremental mode through the forthcoming images of a sequence. This requires the autonomous coexistence, indeed cooperation, of the recognition and learning parts of an integrated system. This integrated system can be implemented within a one-level or multi-level control structure.

For the one-level control structure, an evolution cycle is represented by a single loop integrating the CV system with the CA&E system. This supports system adaptation to a sequence of images. For the two-level control structure (Figure 2), this single loop is decomposed into an additional internal loop of local evolution. This supports system adaptation to both a sequence of images and each single image of a sequence.

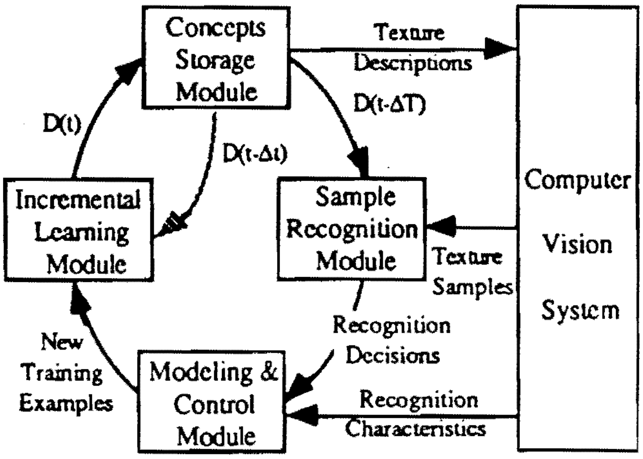


Fig.2 Two-loop evolution cycl

3.1: Initial Training

The goal of the initial training is to teach the system to recognize the most important texture classes typical for a given application domain. The training process is performed on the first image of a sequence (the starting position of a robot vehicle). The extraction of initial training data is done by a teacher selecting image areas of texture classes. These areas are then searched randomly to extract a given number of training examples.

The training examples are provided to the CA&E system. The AQ14 learning program is then applied to acquire initial concept descriptions. The AQ algorithm constructs discriminant concept description in a top-down fashion [5]. The system learns classification rules from preclassified sets of examples (events). Examples are observations from the problem domain, where each example is a list of attribute-value pairs. The set of examples of a given class for which the algorithm is learning rules is called the positive examples. Other classes are characterized by negative examples. AQ begins with the most general description (the empty, all inclusive conjunctive concept) and specializes the concept description by adding one or more particular generalized conditions from the seed (a selected positive example) through the process known as *extend against*. The specialized concept description covers fewer negative examples than the more general concept. When the concept is consistent (covers no negative events), it is saved in the current concept set by disjoining it with the previous disjuncts. Next, a new seed is selected and the process continues until all positive examples are covered.

In case of linear attributes, *extent against* returns a range of values that include the value for the seed and immediately adjacent values not including the negative event's value. The results of extending the seed against each negative event are then combined by intersecting the values for each of the previous tests for each attribute. The system then selects these combined values to get a disjunct that covers the seed, but none of the negative events. The following is an example of a AQ disjunct (equality is used as a relational symbol): [x1=1..3][x2=1][x4=0][x6=1..7][x8=1].

3.2: Texture Recognition

The goal of the recognition process performed by the external control loop is (i) to assign class membership to all texture events of a given image, and (ii) to monitor system recognition effectiveness over a sequence of images. The recognition process performed by the CV system, incorporates concept descriptions D(t-ΔT) previously learned to assign class membership to each texture event and to indicate the confidence level of this decision. Next, segmentation processes unify spatially distributed classification decisions into homogeneous areas of texture classes.

The recognition goal for the internal control loop is (i) to provide guidance for learning processes, and (ii) to verify the effectiveness of the evolution processes and the accuracy of concept descriptions. The internal evolution loop is run through the ordered subgroups of texture events. These subgroups are obtained by dividing the set of events of texture samples. The recognition process is performed in the same way as for the external loop where the most recent concept descriptions D(t-Δt) are applied. The Δt represents the time interval between the current and previous subgroup of texture events. The ΔT, however, represents the time interval of image acquisition between images P(t) and P(t-ΔT), where $\Delta T = \sum \Delta t$ for the number of iterations of the internal evolution loop. The recognition processes of the internal evolution loop are applied on given consecutive subgroups of texture events. The recognition result is a set of texture events associated with recognition decisions and confidence levels.

3.3: Incremental Learning

Since the integrated system has to adapt over time, it has to modify its models dynamically according to new training data. This model modification must be supported by the model acquisition technique working in a dynamic fashion, for example, by the incremental learning methodology. The incremental learning methodology has already been implemented within the AQ family of learning programs [5,8]. It has been proven that incremental learning increases the speed of learning processes. Unfortunately, it can give slightly more complex models and somewhat worse recognition effectiveness [1,8].

Our approach to the evolution of texture descriptions incorporates incremental learning by selecting new training examples, and providing them to the system to use in modifying previously learned concept descriptions D(t-Δt). Incremental learning is performed for each texture class that has to be modified. New training data for learning a single class is both a set of positive examples representing new events typical for a given class, and a set of negative examples representing new events typical for other classes. Previously modified concept descriptions are provided to the learning system, as well. Thus, the incremental learning modifies old descriptions using the changed texture characteristics contained in the new training data.

3.4: Partial Supervision of Evolution Processes

In the first experiments with the presented method of adaptive object recognition under variable perceptual conditions (in dynamic environments), we simplified the developed model evolution approach to a "partially-supervised" evolution of object models. The experimental system included the following technical restrictions: (i) texture segmentation was performed by a teacher through the sequence of images, and (ii) learning processes were initiated every time any texture event was not recognized correctly.

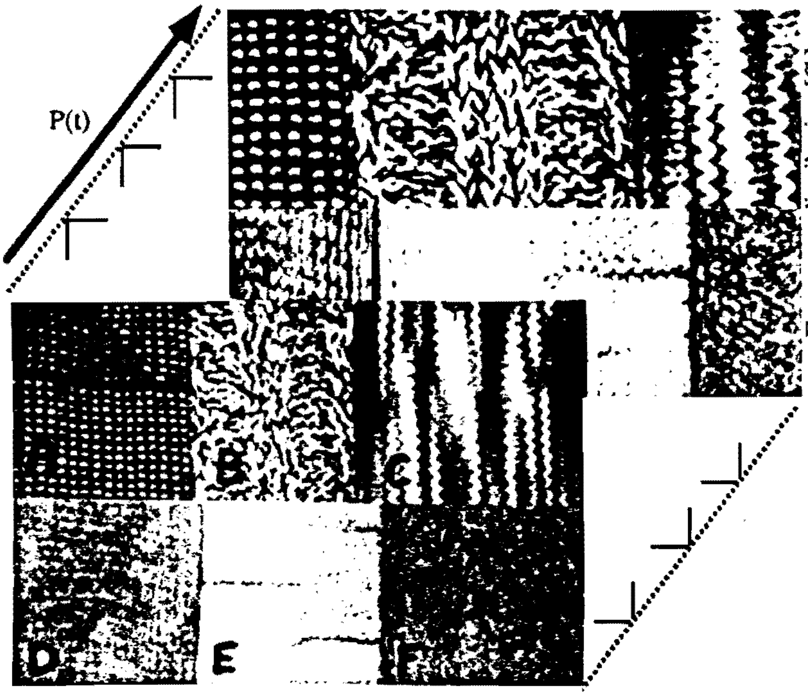
The first limitation, segmentation of texture images by a teacher, was applied to (i) secure perfect separation of different texture areas, and (ii) provide the same number of texture events for their recognition and the selection of new training data protecting the logical soundness of system behavior [9]. These assumptions allow us to focus on the behavioral analysis of evolution characteristics, where other effects influencing this analysis are eliminated. Texture segmentation and the selection of texture events for the evolution experiments were performed through a dialog with a teacher. For each image of a sequence, a teacher drew texture areas corresponding to texture classes. Then, the interface system selected randomly 200 texture events for each class, and for each image of a sequence.

The second limitation, the initiation of learning processes by any data that was not recognized, simplified the complexity of control mechanisms that trigger incremental learning processes. The execution of evolution processes is based on the generation of an *evolution strategy* composed of a sequence of *evolution actions*. We implemented the simplest *evolution strategy* where the initiation of learning processes was triggered by the existence of any incorrectly recognized data. The simplest evolution action, *evolution through generalization*, was chosen for presented experiments, where the system extends by generalization the existing concept descriptions over new positive training examples.

4: Experimental Data

A sequence of five images (Figure 3) was acquired, where the images of a sequence were affected both by variable resolution and variable illumination. Each image was composed of six texture areas (sweaters). The distance between the textured objects and the camera was decreased by half during the experiment. Additionally, the light source was moved along with the camera.

Each image of a sequence was processed to extract texture attributes characterizing texture classes. We modified Laws' [4] well known method of a two step extraction of texture attributes. In the first step, the local micro-characteristics of raw texture data is computed. In the second step, the local macro-statistics are computed to derive statistical measures of the image. Methods based on a similar approach were widely applied to texture feature extraction, and they provided quite good discriminating power [2].



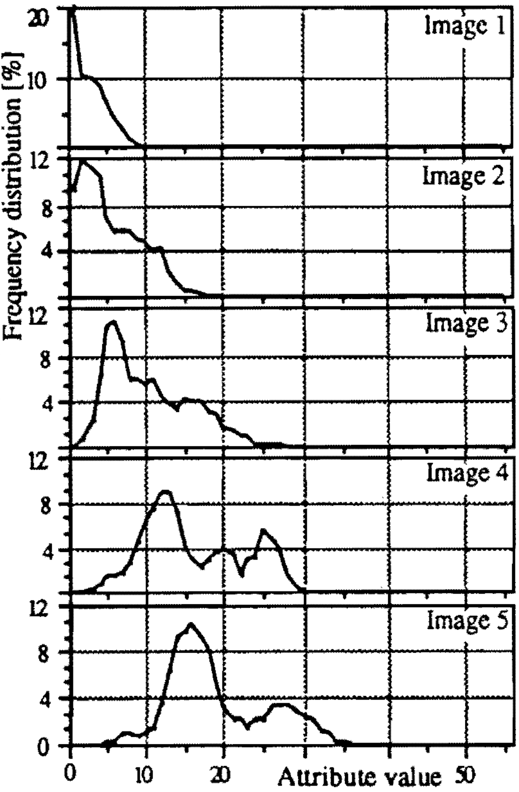


Fig.3 The first and fifth image of a sequence used in experiments, and distribution variability of example texture attribute

The modification of the Laws' method of texture feature extraction included: (i) the extension of the number of filtering masks (i.e., R5R5, E5L5, E5S5, and L5S5) by additional four masks (i.e., E5L5, E5S5, L5S5 and S3S3), and (ii) the redesigning of a window used to compute local macro-statistics from the filtered image from a square to a circular of radius R≈7.5.

Acquired texture features were grouped for each pixel of an input image into a vector of attributes. A single attribute was then quantized onto 57 levels to provide compatible input of training data for the AQ14 learning program. The complexity of attribute distribution is presented graphically (Figure 3) on the example of texture class E, attribute L5E5, and through five images of a sequence. The distribution of the chosen attribute is not normal for any image of a sequence. The distribution changes both its pattern and the translational position within the attribute space. These effects cause a degradation of recognition effectiveness when texture descriptions acquired from the data of one image are applied to recognize textures on another image. This is why we are adapting the vision system to a dynamic environment using the learning-based evolution of concept descriptions.

5: Testing Data and Methodology

To obtain an objective analysis of system performance, evolution effectiveness was measured on different sets of data than data used for system evolution. Testing data used to measure system performance was obtained from different sections of an image before experiments were ran. A single dataset of testing data contained 200 testing events characteristic for a single texture class. Considering a sequence of five images, the total number of testing datasets was equal to 30 datasets (each of 200 testing events), and these datasets were grouped into 5 files corresponding to the five images of the sequence.

The testing phase was applied each time after any evolution loop (i.e., external or internal loop). During the testing phase, currently available models were applied to recognize test datasets grouped into five testing files corresponding to five images of a sequence. Thus, recognition characteristics of model evolution were created by collecting the results from each testing phase and each iteration of model evolution. The created characteristics show the recognition effectiveness of evolved models when these models are applied over and over again to the same data during consecutive testing phases.

The evaluation of system performance on a single image was based on the following three fold criteria: (1) overall system recognition effectiveness expressed by the average recognition rate, (2) stability of the recognition decision measured by the standard deviation, and (3) capability of recognizing all texture classes monitored by the minimum recognition rate.

6: Experimental Results

This section presents experimental results from the evolution of texture descriptions incorporating an incremental learning-based "partially supervised" evolution approach.

6.1: Results for One-Loop Evolution

The recognition characteristics obtained for the one-loop evolution approach are presented in the first row of Figure 4. Each diagram contains recognition characteristics for testing datasets of five images and through five iteration steps. A single curve monitors the recognition performance of evolved texture descriptions on a given image over consecutive evolution iterations.

Let us consider the recognition effectiveness, for example, of models initially acquired from the first image of a sequence. Once acquired models are then evolved over the next images of the sequence; i.e., by the second, third, fourth and fifth image. Every time the models are updated, the system measures their recognition performance on the same testing image; i.e., in our example on the first image. Completed characteristics show the recognition performance of evolved models respective to the first image only; i.e., we can analyze recognition stability of the system. On the other hand, models evolved by preceding images of a sequence can be tested on a particular image that follows these images. For example, models evolved by the first, second, third, and fourth image of a sequence can be applied to the fifth image after each evolution iteration. This means that we measure system recognition performance on a future image of the sequence.

A lack of stability of recognition characteristics is seen particularly for the first and last image of a sequence. Two negative evolution effects are observed; i.e., *lack-of-support* and *lack-of-progress.*

The first negative effect, so called *lack-of-support*, relates to the decrease in the recognition effectiveness for a given image when the evolution of models is continued through the next images. We expect that acquired object models will have similar recognition effectiveness despite the fact that they are evolved over other images characterizing object occurrences.

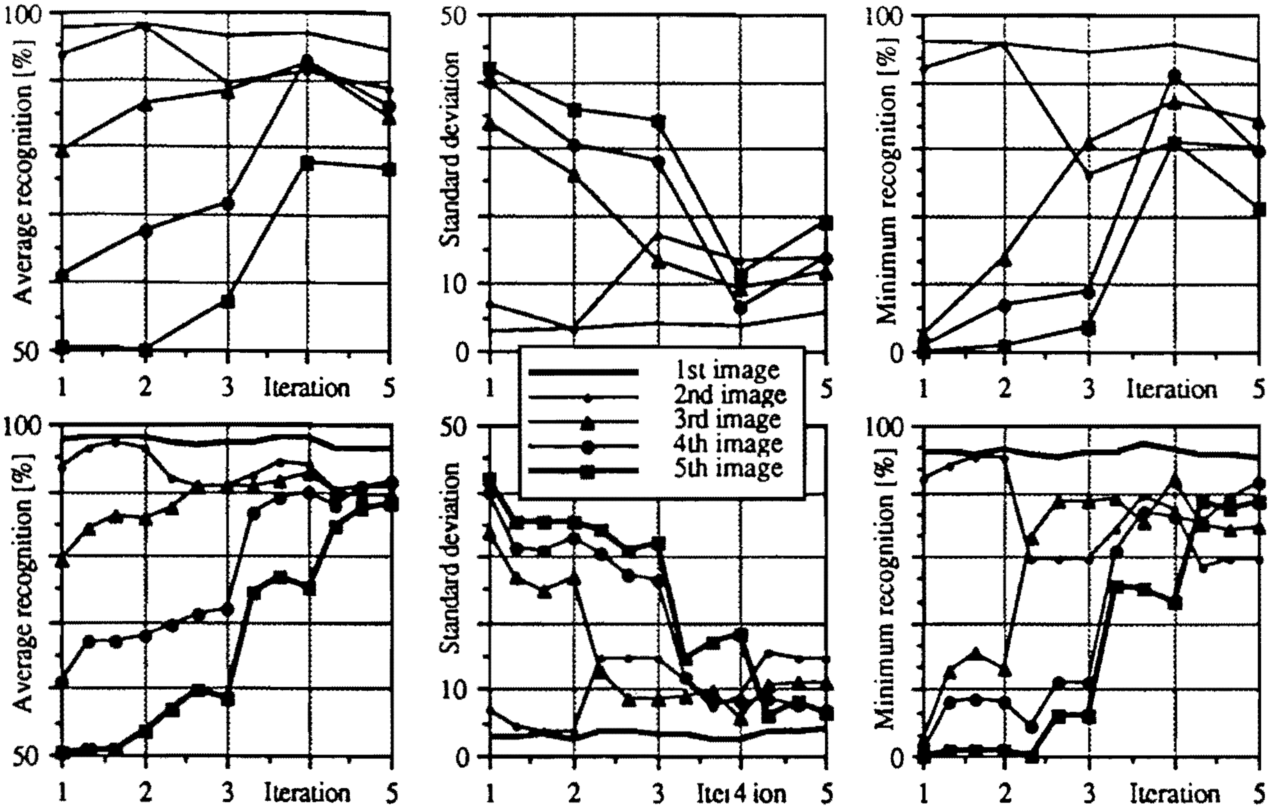


Fig.4 Evolution results for one-loop architecture (the 1st row) and two-loop architecture (the 2nd row)

The first negative effect, so called *lack-of-support*, relates to the decrease in the recognition effectiveness for a given image when the evolution of models is continued through the next images. We expect that acquired object models will have similar recognition effectiveness despite the fact that they are evolved over other images characterizing object occurrences.

The *lack-of-support* is seen for the 1st image characteristics. It is illustrated better in the left column of Figure 5 presenting selected individual recognition characteristics of three classes which influence the instability of system recognition. Figure 5 illustrates the deteriorating recognition effectiveness of initially acquired descriptions of classes C, D and F when these descriptions are evolved over the next images. As mentioned before, we would like to have models that adapt once to a given image and then do not loose their discriminating power during evolution over subsequent images. If models are sensitive to evolution iterations then a large number of such iterations can weaken their discriminating power. This could make the system unable to recognize the concept variation that it learned previously.

We find that *lack-of-support* for recognition effectiveness is caused by noise accumulation over the sequence of images. The selection of new training data does not perceive and exclude noisy examples from being provided to the learning module. This selection of new training data was performed by the simplest method possible and was simply the extraction of all texture events that were not recognized. In this way, new training datasets of selected texture events are characterized by lower Signal-to-Noise Ratio (i.e., they are more noisy) than datasets provided to the recognition phase. More careful selection of new training data is recommended.

The second negative effect, so called *lack-of-progress*, relates to the unexpected drop of the recognition effectiveness after the model adaptation is performed to a given image. In this case, we expect that the recognition effectiveness will be higher (or at least on the same level) for the current image rather than for the directly preceding one.

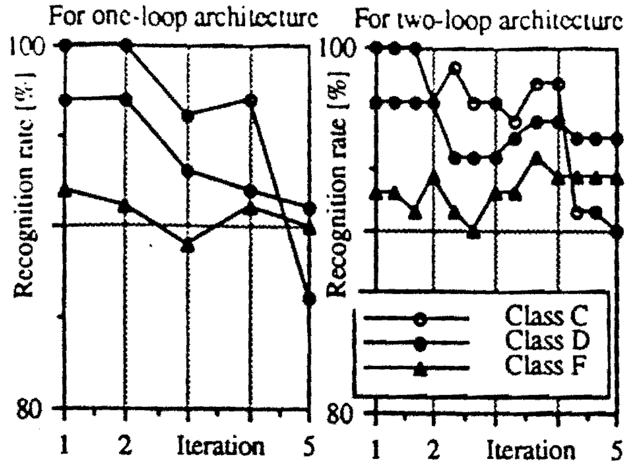


Fig.5 Improvement of the recognition characteristics on the 1st image (lack-of-support effect)

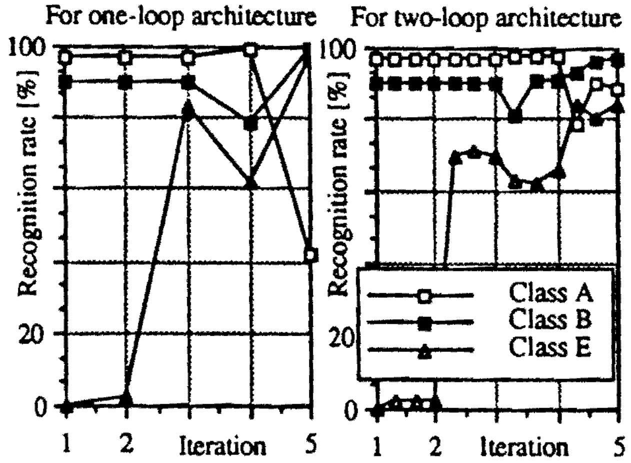


Fig.6 Improvement of the recognition characteristics on the 5th image (lack-of-progress effect)

The *lack-of-progress* is seen for the 5th image. It is illustrated better in the left column of Figure 6 presenting selected individual recognition characteristics of classes A, B and E. In this case, evolved models were applied over and over again to recognize textures on the last image of the sequence; i.e., on the 5th image. It is seen that the predictive power of the description of class A was very high from the first to the fourth iteration. Because of this high prediction accuracy, class A was not evolved over preceding iterations (i.e., over the first, second, third and fourth images) and thus the model of class A did not compete with other models. Finally, it was not evolved over the fifth iteration (i.e., over the fifth image of a sequence) due to the set of new training data being empty for class A – class A was recognized perfectly by models after the fourth evolution iteration. Simultaneously, descriptions of classes B and E were aggressively evolved during the last iteration, eroding the description of class A. Classes B and E were recognized at a much lower recognition level in the preceding iterations, causing the new training datasets for these classes to consist of data that significantly moved concept descriptions through the attribute space.

Two solutions to the elimination of this negative effect (i.e., *lack-of-progress*) were proposed; i.e., (i) the development of a two-level control system that evolves models through split training datasets (see next section), and (ii) the design of a new evolution-oriented learning kernel that allows for continuous evolution of model descriptions.

6.2: Results for Two-Loop Evolution

The recognition characteristics obtained for the two-loop evolution of texture concepts are presented in the second row of Figure 4. Each line representing the recognition of an image has two types of measurement points; i.e., points corresponding to the external evolution loop (integer iteration values), and points corresponding to the activated internal evolution loop (fractional iteration values). There were five iterations of the external evolution loop. For each external loop there were three iterations of the internal evolution loop.

The two-loop evolution of texture models has smoother recognition curves than the one-loop evolution. This demonstrates that two-loop evolution improves the general stability of the system. It is seen that the two-loop evolution of texture models reduces the two negative effects, *lack-of-support* and *lack-of-progress.*, that cause recognition instability when one-loop evolution is applied.

Figure 5 shows a slight improvement in the support (decrease in *lack-of-support*) of chosen concept descriptions when evolved through other images. The *lack-of-support*, however, was not eliminated completely. Further reduction is expected by filtering new training datasets to reduce the influence of noise on the evolved models.
On the other hand, Figure 6 presents a significant reduction of the *lack-of-progress* when two-loop evolution approach is applied. The model evolution through the internal loop even recovered the recognition effectiveness for the description of class A. This recognition effectiveness did not drop to a dangerously low level (indicating that class A was not recognized at all.) The recognition effectiveness for other classes (i.e., class B and class E) was stabilized as well.

7: Conclusions

This paper presented a new approach to the object recognition under variable perceptual conditions, where objects have variable characteristics. The proposed approach to invariant object recognition has been illustrated through the evolution of texture descriptions. The approach integrates computer vision with machine learning and assumes that (i) the system has to recognize objects on each image of a sequence, (ii) the images demonstrate the variability of conditions under which objects are perceived, (iii) an observer and objects can move, (iv) the extraction of texture attributes and training examples can be imperfect, and (v) the system has to work autonomously or semi-autonomously.

We utilized images of a sequence to adapt system models to perceived variabilities of texture characteristics. The experiments presented in this paper were run in a "partially-supervised" mode as opposed to a fully autonomous model evolution. The experiments were compared based on the following two approaches: (i) a one-loop evolution, and (ii) a two-loop evolution of object models.

Future experiments involve further modifications of the system architecture. These modifications include: elimination of partial supervision through the development of a scene understanding module for automatic selection of texture samples, filtering of new training data, and precise guidance of the evolution of concept descriptions.

Acknowledgements

This research was done in the GMU Center for Artificial Intelligence. Research of the Center is supported in part by the Defense 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, in part by the National Science Foundation Grant No. IRI-9020266, and in part by the DARPA Machine Learning Program.

References

[1] Bentrup, J.A., G.J. Mehler and J.D. Riedesel, "INDUCE 4: A Program for Incrementally Learning Structural Descriptions from Examples", UIUCDCS-F-87-958, Computer Science Department, University of Illinois, Urbana, 1987.

[2] DuBuf, J.M.H., M. Kardan and M. Spann, "Texture Feature Performance for Image Segmentation", *Pattern Recognition*, Vol.23, No.3-4, pp.291-309, 1990.

[3] Goldfarb, L., "On the Foundations of Intelligent Processes - I. An Evolving Model for Pattern Learning", Pattern Recognition, Vol.23, No.6, pp.595-616, 1990.

[4] Laws, K.I., "Textured Image Segmentation", PhD Thesis, Dept. of Electrical Engineering, University of Southern California, Los Angeles, 1980.

[5] Michalski, R.S., "A Theory and Methodology of Inductive Learning", in Machine Learning: An Artificial Intelligence Approach, TIOGA Publishing, Palo Alto, CA, pp.83-134, 1983.

[6] Pachowicz, P.W., "Learning Invariant Texture Characteristics in Dynamic Environments: A model evolution approach", Report MLI-2-91, Center for Artificial Intelligence, George Mason University, 1991.

[7] Pachowicz, P.W. and J. Bala, "Texture Recognition through Machine Learning and Concept Optimization", Report MLI-6, George Mason University, Center for Artificial Intelligence, 1991 (submitted also to the IEEE Tr. on Pattern Analysis and Machine Intelligence).

[8] Reinke, R.E. and R.S. Michalski, "Incremental Learning of Concept Descriptions: A Method and Experimental Results", Machine Intelligence 11, J.E. Hayes, D. Michie and J. Richards (Eds), Clarendon Press, Oxford, pp.263-288, 1988.

[9] Rine, D, "Software Maintenance: by means of retrainable software", to appear in the *Int. J. of Software Maintenance*, 1992.

[10] Roan, S. J., J.K. Aggarwal and W.N. Martin, "Multiple Resolution Imagery and Texture Analysis", *Pattern Recognition*, vol.20, No.1, pp.17-31,1987.

[11] Rosenfeld, A. and L. Davis, "Image Segmentation and Image Models", *Proc. of IEEE*, Vol.67, No.12, pp. 7646-7772, 1979.

[12] Unser, M. and M. Eden, "Multiresolution Feature Extraction and Selection for Texture Segmentation", *IEEE Trans. Pattern Analysis and Mach. Intel.*, Vol.PAMI-11, No.7, pp.717-728, 1989.