Low-level Numerical Characteristics and Inductive Learning Methodology in Texture Recognition

Peter W. Pachowicz

Center for Artificial Intelligence, George Mason University
4400 University Drive, Fairfax, VA 22030

ABSTRACT
The presented method applies inductive learning to the texture recognition problem. The method is based on a three-level generalization for texture classes description. The first step, scaling interface, transforms local texture features into their higher symbolic representation as numerical intervals. The second step incorporates the AQ inductive learning algorithm to find description rules. The third step applies the SG-TRUNC method for rule optimization. The medium recognition ratio for this method was over 90% and all classes of texture were recognized. In comparison, the k-NN pattern recognition method failed to recognize all classes of textures and had a recognition ratio of 83%.

INTRODUCTION
In this paper, we present a method that applies inductive learning to low-level texture description and recognition along with its effectiveness in texture recognition problem. We compare this method with the k-NN pattern recognition (PR) technique in imaging the complexity of texture images along with sensitivity of chosen features. An automatic scaling of quantitative data of texture features (e.g., responses to local convolution masks) has been used to transfer numeric data into their symbolic intervals. Later, these symbolic values are processed by machine learning symbolic computation.

This approach to the integration of low-level numeric computation and high-level symbolic processing is the first step in the creation of a vision system, which will be capable to adapt to dynamic environment. Such a system will acquire knowledge from its environment and modify its internal structure as well as improve its recognition performance over the system experience. Classic adaptability and learning in engineering systems use control engineering [7] and/or pattern recognition [2] techniques. These approaches work well with quantitative information. But, the typical processes of high-level vision are based on symbolic computation. The boundary between the numeric and the symbolic nature of computation has not been established. Iconic approach has been successfully applied to invariant recognition and data fusion [13]; however, the application of such a method to an object inspection task is questionable. The inspection task uses models of object (or models of object variances) that are acquired through explanation, technical diagrams, and tolerances, rather than from presentation of various real objects during a teaching phase.

This research was done in the Artificial Intelligence Center of George Mason University. Research activities of the Center are sponsored in part by the Defense Advanced Research Projects Agency under grant, administrated by the Office of Naval Research, No. N00014-87-K-0874, in part by the Office of Naval Research under grant No.N00014-88-K0226, and in part by the Office of Naval Research under grant No. N00014-88-K-0397.
Our proposed approach to robot vision uses machine learning (ML) methodologies that are applied to object location, recognition, inspection, and scene understanding tasks. The method is based on the creation of system adaptability functions. Learning capabilities allow the use of situation context in the system adaptation to dynamic environment. In our approach, models used to represent data and system knowledge must possess dynamic structure that can be modified during experience by the system itself. Using ML and cognitive processes like induction, deduction, discovery and analogy, we plan to combine elements of visual recognition and visual imagery [9] into a common system. Currently, we are investigating vision adaptability mechanisms that can be used to recognize objects in different external situations which are not within the system experience. Such situations include 3-D rotation, changed resolution, quantitative and qualitative noise, light reflectivity, and mutual relations of objects as occlusion.

The system has to work during the following vision tasks: object location, recognition and inspection. We assume that prior to the system work, it either possesses an initial model of objects or acquires itself the optimal model from the environment using background meta-knowledge. It is required that the adaptive system must be able to manage its internal 'dynamic memory' in order to increase/decrease the number of distinctive features, add new objects, and modify data processing flow.

One such typical task that must be based on such adaptive approach is texture-based object recognition for robot navigation within natural terrain. The difficulties in robust 3-D objects recognition are caused by a great variability of images projected on the sensory array. This variability is caused not only by the 3-D rotation of an object, but also by changes of resolution, positioning of light source, mutual relations with other objects at the scene, etc. In the classic approach to object recognition, which is opposite to our approach, the system is associated with a set of object distinctive features that are used for distinguishing a particular object from other objects. Such set of features is stable [1], i.e. the perceptual system is not able to verify the usefulness of the object model. Therefore, the classification is as correct as the accuracy of created recognition models.

Before applying the dynamic approach to texture-based object recognition, we must test the usefulness of the inductive learning methodology [5] as a kernel of our future system. This introductory work tests both the application of inductive learning to texture recognition problem, and also the scaling method as an early generalization phase and an interface of numeric to symbolic data transformation.

TEXTURE FEATURES EXTRACTION
Six input images have been obtained from the black and white camera, where each pixel of a 512x512 image has been coded on 64 gray levels. They represent six classes of texture: pressed cork, grass lawn, woolen cloth, water, pigskin, and fur (see Figure 1). They have been illuminated irregularly. Two of them (grass lawn and water) have smoothly changing texture resolution, caused by the screw direction of image projection. Each of these input images has been divided into two files of image data. The left part of each input image has been designated for the learning phase, while the right part has been applied for the recognition phase.

Two classic methods, Laws's masks and co-occurrence matrices, have been used for texture features extraction and sample data preparation [12] (see Figure 2). For the Laws's masks method we have computed 200 random learning samples for each class of texture, where each sample is a vector of eight features. A single sample, characterizing the macrostatistics of a 20x20 window, was computed as the medium absolute value of local single feature for all pixels within the window. A single
Fig. 1 Sample subimages of six classes of texture (each subimage is composed of 200x200 pixels each printed in 5 grey-levels); a) pressed cork, b) grass lawn, c) woolen cloth, d) water, e) pigskin, f) fur

Feature used for the computation of the macrostatistics is a response for a single Laws's convolution mask [4]. There are eight local features of texture computed for one 3x3 and seven 5x5 convolution masks. For the co-occurrence matrix method, we have also computed 200 random learning samples for each class of texture. The sample was also composed of eight features computed for four co-occurrence matrices. Each matrix was characterized both by the angular second momentum (matrix uniformity) and by the contrast as a measure of the spread of values away from the main diagonal [11]. A single matrix was computed for a 20x20 window of pixels and for two parameters: the distance between two pixels (that are compared) and the angle as the direction of pixel displacement. A single position within the matrix is the frequency of the occurrence of couple of pixels, where gray-levels of pixels correspond to the coordinates of the matrix.

Fig. 2 Numerical features extraction for texture description phase
FIRST GENERALIZATION PHASE -
Scaling Interface
The application of inductive learning to texture description and recognition requires the creation of a special interface of numeric-to-symbolic data conversion. To fulfill this requirement, we have applied the simplest scaling of numeric texture features into their symbolic intervals. In this way, the scaling process is an early generalization phase that generalizes numerical samples of the feature space into their symbolic representation as a cubic cell of a feature space interval. In this work, we used static scaling that assumes the constant number of texture classes and features. Static performance has been determined by a priori given set of events that could not be changed or extended by additionally acquired data from the environment. The future application of dynamic scaling assumes that a system will be able to extend the number of texture classes, to add new events that are characteristic for a single class, and to change the number of attributes or to modify them.

The scaling interface classifies a numerical feature as one of 50 symbolic intervals. An automatic computation of the border values for each symbolic interval is discussed in a separate report [8]. After the execution of first scaling, the system checks the consistency of created symbolic data. For those events that were inconsistent (i.e., when events of two or more classes have the same symbolic values), the system creates an additional class of inconsistent events and the scaling process is repeated on the lower-level. The application of lower-level rescaling requires that each class of events be separated. Considering this task, the system predicts that non-limited recursive rescaling would create a very complicated hierarchical structure (running to high resolution structure), making it difficult to execute object recognition processes. Therefore, we have set up a criterion for the scaling applied on x-th level of the scaling hierarchy, which states that the scaling of the lower-level must be applied if more than 5% of learning events for a given class are inconsistent. Such events are removed on the given x-th level of scaling hierarchy and placed into the additionally created class of inconsistent events. For our textures, the scaling processes have not been applied on the lower levels because the described criterion has not been fulfilled.

SECOND GENERALIZATION PHASE -
Inductive Learning
The inductive incremental learning program, AQ14 [3,5], has been applied to learn texture attributional descriptions from examples. The AQ program performs a heuristic search through a space of symbols expressions to find the most preferred expression according to the specified criterion. The input to the AQ program is a string of learning events with each event as a vector of attribute values. The set of events obtained for one class, and to change the number of attributes or to modify them. class is called a set of positive examples. Regarding this particular class, all other events are negative examples. The program finds optimal cover of all positive examples and no negative ones, and it produces decision rules to discriminate all classes of texture images between themselves. The conditional part of a rule, defined as a cover, is a disjunction of the complexes (by the use of OR operator). Then, each complex is decomposed into selectors which is a variable or a disjunction of values (by the use of AND operator).

In our experiment, the input data were composed of six sets of learning events according to six texture classes, with an additional seventh set of inconsistent events. A single event was composed of eight attributes, representing one of two approaches to texture characteristics obtained from Laws's masks or co-occurrence matrices. As mentioned above, each attribute was coded on 50 levels. The AQ inductive incremental learning program was run in two modes, producing intersecting
or disjoint covers. Discrimination processes of the intersecting mode produce covers that can logically intersect with covers of other class over "do not care" areas of the event space. On the other hand, discrimination processes of the disjoint mode produce only such covers that do not intersect at all with covers of other classes. As a consequence, rules produced by the intersecting mode are more general than rules produced by the disjoint mode.

THIRD GENERALIZATION PHASE -
Rules Optimization
Rules optimization has been applied as the post-processing phase of rules generation. The theory of a two-tiered description of imprecise concepts [6,14] has been used to truncate rules generated in the second generalization phase. A simple two-tiered concept description generates both the Base Concept Representation (BCR) as a typical properties of a concept, and also the Inferential Concept Interpretation (ICI) as allowed concept modifications. The SG-TRUNC method has been used to obtain BCR through a sequence of generalization and specialization operations [14]. First of all, the SG-TRUNC method performs generalization to remove selectors from the complexes. This makes a complex is more general, i.e., covering more examples. Then, a specialization operation removes a number of complexes making the description less general and covering less examples.

Processes of the rule optimization are based on rule characteristics. Such characteristics are composed of two coefficients, $t$-weight and $u$-weight. $T$-weight is the total number of examples covered by a complex, $u$-weight is the number of examples covered by only one complex. The SG-TRUNC method preserves those complexes that have high $t$-weights and high $u$-weights and modify complexes with low $t$-weights and low $u$-weights. The degree of rule optimization is controlled by two parameters (each of which can vary from 0 to 1.), controlling the removal of selectors and the reduction of complexes. An increase of the parameter value causes an increase of rule generalization. In our experiments these control parameters were relatively very low, i.e., they were equal to 0.05, indicating that the optimization of rules was low.

TEXTURE RECOGNITION
Inductive descriptions of texture classes have been tested in comparison with the classical pattern recognition approach. The recognition process was applied on the right parts of input images that have not been seen by the system before. The same two methods used for the learning phase were applied to texture feature extraction. A set of 100 samples was obtained for each class of texture and for each of two texture description methods, Laws's masks and co-occurrence matrices. Those events were scaled by the data conversion interface using scaling parameters calculated during the learning phase.

Recognition by inductive assertion
The ATEST program [10] has been applied to support texture recognition by inductive assertion. This program evaluates the overall performance of the rule base. In our case, the program was working on separate sets of events, where each set was obtained for a single class of texture as described above. Each event was classified into one of six classes of texture. There were three possible classification decisions of a single event: an event belongs only to the correct class (unique-classification), an event does not belong to the correct class (miss-classification), and an event belongs to more classes where one of them is the correct class (multiple-classification). The final recognition decision was made, based on counting unique- and multiple-classified events for a single tested image. Such classification is called first rank assertion, our main measure of the recognition effectiveness.
Table 1 Recognition results for two-phase generalization of texture features

<table>
<thead>
<tr>
<th>Class</th>
<th>Features extraction method</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Class 5</th>
<th>Class 6</th>
<th>average recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Laws’s masks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Co-occurrence matrices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of complexes</td>
<td>recognition ratio</td>
<td>number of complexes</td>
<td>recognition ratio</td>
<td>number of complexes</td>
<td>recognition ratio</td>
<td>number of complexes</td>
<td>recognition ratio</td>
<td>number of complexes</td>
</tr>
<tr>
<td>Class 1</td>
<td>38</td>
<td>72%</td>
<td>26</td>
<td>88%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 2</td>
<td>35</td>
<td>74%</td>
<td>31</td>
<td>78%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td>23</td>
<td>81%</td>
<td>24</td>
<td>90%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 4</td>
<td>8</td>
<td>93%</td>
<td>6</td>
<td>99%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 5</td>
<td>31</td>
<td>73%</td>
<td>46</td>
<td>57%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 6</td>
<td>23</td>
<td>87%</td>
<td>13</td>
<td>84%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>average recognition</td>
<td></td>
<td>80%</td>
<td></td>
<td>83%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Recognition by pattern classification

We have considered several classic parametric and non-parametric PR methods for texture description and recognition [2]. The parametric methods have been excluded after testing the feature space because parametric models of features distribution were not representative, due to irregular distribution of teaching and difficulty in estimating by parametric curves. Therefore, we used the k-NN non-parametric statistical pattern recognition method to provide a simple comparison of static ML and PR approaches to texture recognition, as well as to imagine the texture complexity and effectiveness of the features extraction. The main advantage of this method deals with the case of irregularity and complexity of teaching data domain. However, the requirement to store all teaching data (or selected data as most representative samples of feature domain distribution) for the recognition phase is a great disadvantage.

RESULTS COMPARISON

The results comparison have been studied in two phases. In the first phase, rules obtained from the second generalization step have been applied to recognize texture classes. The second phase tests rules were obtained from the third generalization step. Results from these two experiments are presented in Tables 1 and 2, while Table 3 presents results for the k-NN pattern recognition method.
The analysis of the results presented in Table 1 and 3 gives us the following conclusions for the application of two-phase generalization method in texture recognition:

- The average recognition effectiveness factor is the same for the ML and the PR approaches when the texture features were acquired by the co-occurrence matrices method. For the Laws's masks method, the ML approach to texture recognition is comparably better than the PR approach.

- Recognition effectiveness has been significantly decreased by the low recognition rate obtained for Class 5 of the texture category. Considering the minimum threshold on the 50% level, the inductive learning approach recognized all textures, but the k-NN method was not able to recognize Class 5 texture. In this case, ML approach proved to be much better as well as in the case of recognition of Class 2 texture.

- Neither the Laws's mask method nor the co-occurrence matrices method of texture features extraction were stable when compared to each other for each class of texture. The Laws's mask method gave generally worse results, considering both the number of generated complexes of the rules and the recognition results. But in case of Class 5 texture, the number of complexes was significantly lower and the recognition ratio was higher. This conclusion suggests the future application of a mixed feature extraction method.

Presented in Tables 1 and 2, recognition results were obtained for the intersection cover mode of the inductive learning algorithm, where covers generated for different classes cannot intersect. In this case, the generation of rules for the intersecting mode was much faster than for the disjoint cover mode, where covers generated for different classes cannot intersect themselves. The average recognition effectiveness factor was also higher. The detailed analysis shows that for the similar number of complexes generated in the intersecting mode and in the disjoint mode, the recognition results are better for the disjoint mode. This case is very interesting for the application of dynamic texture recognition and will study it in future.

Recognition results for rules obtained after the three-level generalization processes yielded a significant increase in the value of recognition ratio (see Table 2). These results are summarized below:

- The average recognition ratio has increased, as much as over 90% in the case of the Laws's masks method of features extraction.

- The recognition ratio for Class 5 texture has been significantly increased up to 83% (this texture class was not recognized by the PR approach). Analysis of the variation of recognition ratios presented in Table 1 and 2 is very interesting. This variation has been reduced, i.e., classes with previously lower ratios are recognized after the application of SG-TRUNC method of rules optimization with higher recognition ratios. On the other hand, there is the decrease of recognition ratios for classes with previously very high ratios.

The number of selectors and the number of complexes have been reduced significantly. The recognition rules both for the Laws's masks method and the co-occurrence matrices method are much more optimal. The secondary effect of this optimization is the smoothing of recognition ratio through out all texture classes as well as the increase of recognition speed.
CONCLUSIONS

The main goal of this work has been the implementation and testing of the inductive learning approach in texture recognition, where textures have been characterized by well-known methods of low-level features extraction. We have applied three hierarchical levels of the generalization processes: the scaling generalization, the generalization by inductive learning, and the generalization by rule optimization. We have shown that the scaling method can be applied as an interface for numeric-to-symbolic data conversion, allowing the use of symbolic computation (typically executed on higher levels) on the lower levels of the vision hierarchy. The comparison with the simple k-NN pattern recognition method presents the complexity of our textures and the difficulties associated with their accurate recognition. This work has shown that the ML (inductive learning) approach can be applied successfully to typical pattern recognition problems. Obtained recognition results for each of the texture classes and the medium recognition ratio (equal to 91%) are quite satisfactory at this stage of our research work. We underline that ML approach has recognized Class 5 texture with the 83% ratio, while it has not been recognized by applied PR approach. The three-level generalization has smoothed the recognition ratios for all classes of textures.

REFERENCES


