90-21

IEEE International Workshop on Intelligent Robots and Systems
IROS '90

Saln

LEARNING-BASED ARCHITECTURE for ROBUST RECOGNITION of VARIABLE TEXTURE to NAVIGATE in NATURAL TERRAIN

Peter W. Pachowicz

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

Abstract: Since natural terrain consists of textured objects that can be perceived under different external conditions, we apply machine learning methodology to support the recognition of variable texture. We present the results of first introductory experiments and the development of a new system architecture incorporating learning tools; i.e. conceptual clustering, learning from examples, and learning flexible concept. We present designing methodology and system architecture of three functional levels typical for the large-scale control systems; i.e. self-tuning to a given content of texture image in order to extract most sensitive features and to group them into patterns, learning a concept of new texture, and control of system adaptation (guided by vision goal, feedback verification of created hypotheses, and a plan of the environment content). We discuss the requirements for learning tools that are used to build such adaptive vision systems and we present their further development.

1. INTRODUCTION

Since the natural terrain contains textured objects and the conditions of texture perception are variable, the robust recognition of variable texture is crucial for the execution of the navigation task. We incorporate machine learning (ML) methodologies to the creation of a system that has to support great variability of texture features. We consider the following variability of texture: (i) changing resolution, (ii) lighting conditions, and (iii) surface orientation. Texture variability makes difficulties for the efficient representation of texture features in the multidimensional feature space by traditional methods. Our approach is based on the conversion of numeric features into their symbolic representation and on the learning of rule description of texture concept. The rule representation is well suited to manipulate of texture description including: optimization, representation of imprecise concept, modification by newly acquired data, merging and splitting concepts, classification into classes, representation of concept variations, etc. We present the evolution of our approach to the learning texture concept, which has been motivated by the introductory experiments [19,20].

Accurate recognition and segmentation of textured images is crucial for the robot navigation in outdoor natural terrain (especially for cross-country navigation). While the detection of objects and world modeling can be done (under some assumptions) using range scanner, the recognition of objects, their classification into certain classes (e.g. obstacles, "do not care" objects) and also the recognition of material an object has been built, pays an important role in the modeling of the environment for the navigation task [13].

The teaching phase of supervised texture classification is performed in the following three steps: (1) extraction of texture features, (2) description of feature space, and (3) evaluation performed on preclassified texture images. The recognition phase applies created descriptions of texture classes to segment textured images that was not seen to the system before. This short scenario assumes that the created description of texture classes is noise free, and it was acquired for known environment. However, in case of autonomous robotics and variable texture, the assumption of noise free teaching data cannot be fulfilled. Moreover, an autonomous vehicle must be supported by unsupervised learning techniques to allow the execution of a given task in unknown environment.

Dealing with the above reasons, the traditional non-learning approach to texture recognition problem has limited applications. Creating robots, robust and capable of navigating under changing conditions, we present the development of a new approach to the texture recognition and segmentation that incorporates ML techniques. We apply symbolic learning techniques to the numeric domain. Considering the domain of texture features, this approach allows us to code and to manipulate both of numeric and of structural features. Applying such hybrid type of texture features, we also expect to increase the recognition effectiveness. Rule representation used to code the description of texture concept gives us an opportunity to manipulate of knowledge more easily and flexibly than using other, for instance, pattern recognition methods.

2. MOTIVATION and JUSTIFICATION

2.1 Texture concept

Let us define a concept of visual texture as repetitive patterns, in which "elements" or "primitives" are arranged according to certain "placement rules" [23]. This definition is more general and very similar to the other definitions [6,7]. The starting point is that a texture is composed of patterns distributed on an image. This distribution depends on the characteristics of a given texture which can be random or order statistically and/or structurally. Repetitive patterns can differ in size and their own characteristics. They can form a group of patterns typical for a given texture. If so, then the patterns' distribution within a group can also be used to describe a texture.

Texture patterns are created and characterized by hierarchically lower texture elements (primitives). The spatial distribution of these elements is typical for a given pattern. Texture elements represent micro-characteristics of texture obtained through the analysis of small amount of local pixels. Such micro-characteristics can represent both statistical and also structural domain of texture. We can compute texture features on different levels of abstraction. So, a texture feature can represent local numeric and/or structural micro-/macro-characteristics, texture patterns, and groups of patterns.

Let us consider statistical texture domain, which is typical for outdoor natural terrain. There are numerous methods for statistical texture analysis and recognition that are in line with the above mentioned texture concept, e.g. co-occurrence matrices [6], Laws' masks [12], difference histograms [24], etc. Authors of comparative studies [29] indicated the most powerful, the co-occurrence matrices method, however, they suggest using multiple methods instead of a single one. Methods developed later include, for example, Markov random fields [4] and small structural polygons [24] applied for texture modeling.

2.2 Texture variability

The characteristics of visual texture depends both on the method used to characterize a texture and also on the variability of texture caused by external conditions. Currently, we do not have an universal method for robust texture recognition. While one method can be very sensitive to a given texture, it can be inefficient for another. The development of a universal method seems to be irrational especially after three decades of active research in computer vision. Therefore, we focus on multiple methods instead of a single one and arrange them into a system. Then, the selection of the best methods for a given variable texture must be performed automatically by the system.

Changing resolution varies both the repetition rate of texture features and patterns, and also the characteristics of a single pattern. The variability of the repetition rate is primarily used (as a positive effect) to determine surface orientation in space [11]. However, the influence of changing resolution on the pattern characteristics is highly negative and difficult to predict. The increase of resolution brings to the evidence new primitives, while the decrease of resolution blurs texture details toward constant tone and some pattern elements may be no longer visible. Since the analysis of statistical texture is based on the spatial variations of local pixels, the extraction of texture elements and texture patterns is affected by the resolution. The variability of texture features obtained for different resolutions has been demonstrated in [23,27]. In several cases, the variability of texture features can be approximated and used to predict features for the other resolution, however, the extrapolation of such characteristics can be inefficient.

Changing lighting is caused by the characteristics of light source (e.g. intensity, light spectrum), its position in relation to the texture surface, and irregularities like shadows. The variability of texture concept caused by changing lighting can be easily observed for specific 3-D textures composed of large 3-D material structures. These structures can reflect the propagated light differently when compared to light propagation, position, and spectrum (sunrise, sunset). Particularly, sharp shadows can create a subtexture overlapped with physical texture.

Changing surface orientation has the influence on the projection of 3-D texture onto 2-D image. Natural texture perceived from different orientations to the texture surface can bring into the evidence new texture elements and hide other elements by partial or full occlusion. For example, a texture composed of mushrooms, will be viewed differently from the top-down direction (circular hats) and the screw direction (trunks and side-viewed hats). Moreover, the screw projection of a texture surface involves the resolution variability.

2.3 Texture segmentation and recognition

The segmentation performs a discrimination test and separates areas of different texture. A test is based on the texture measure incorporating statistical or structural method. Segmentation methods based only on statistical features do not take into consideration the spatial relationship of texture features between themselves, however, they are used to derive directionality of a texture. When a texture has micro-stuctural character, then elements of local structure can be found as symbols, combined into structural patterns (as triangles, rectangles, etc.) and characterized in relation to each other [28]. Such structural approaches applied on the lowest level of vision hierarchy had significantly increased recognition effectiveness.

Segmentation processes can be executed in supervised or unsupervised mode. Supervised segmentation requires correct presentation of segmented image in order to derive characteristics of each texture. In unsupervised segmentation the system has to acquire a new texture concept itself. Since there are no training decisions available, the conceptual clustering method is helpful in finding out the distinction of texture classes [3,9].

A texture concept can be represented using a parametric or non-parametric method. To choose one of them, we have to analyze the distribution of texture features in the feature space. For texture domain, this distribution is highly irregular, contains noise and it does not represent all variable occurrences of texture. In such cases, the description of texture concept using parametric functions is not suitable. Most appropriate are non-parametric methods, e.g. information trees [8]. In our approach, a non-parametric method, i.e. rule representation, is applied to describe a texture concept. Such representation codes complicated feature clusters as covers containing teaching examples distributed through the feature space. Rule representation is also well suited for incremental learning and manipulation of concepts.

2.4 ML-based texture recognition - first experiments

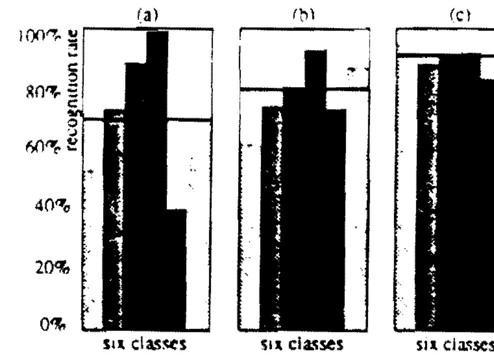
2.4.1 System architecture

The traditional open-loop architecture has been applied to learn texture descriptions from preclassified data in order to investigate novel approach before the creation of advanced methodology and a system.

In our experiment (see Figure 1), image data for the learning phase was composed of six preclassified texture images (pressed cork, lawn, wooled cloth, water, pigskin and fur). The images were characterized by irregular lighting and several of them had changed resolution caused by the screw projection. The quality of images was low. Input images were processed by well known techniques for texture features extraction; i.e., Laws' masks and coocurrence matrix method. A vector of eight features was extracted for each pixel and the method. The scaling module performed the conversion of numeric features into their symbolic intervals. The consistency of learning data was checked, and in the case of inconsistent interval, the local scaling of higher resolution was performed. The AQ inductive learning algorithm [15] was applied to learn texture attributional descriptions from symbolic events and these rules were optimized by incorporating two-tiered representation of imprecise concepts [16]. A simple two-tiered concept description generates both the Base Concept Representation as typical properties of a concept, and also the Inferential Concept Interpretation as allowed concept modifications. The SG-TRUNC method [30] was applied to optimize rules.

Table 1 Comparison of machine learning and pattern recognition approaches to texture recognition (a bar represents average recognition)

	PR approach k-NN (see a)	ML approach rules (see b)	ML approach optimal rules (see c)
average recognition	70%	80%	91%
highest recognition	99%	93%	98%
lowest recognition	40%	72%	83%
averaged deviation	16.3	7.0	4.0



The texture recognition system was composed of the following modules: feature extraction, scaling, and inductive assertion. The first two modules were similar to the corresponding modules of the learning phase. The inductive assertion module was created incorporating a software tool developed for rule base testing [21]. It performs the classification of test texture samples using learned rule description.

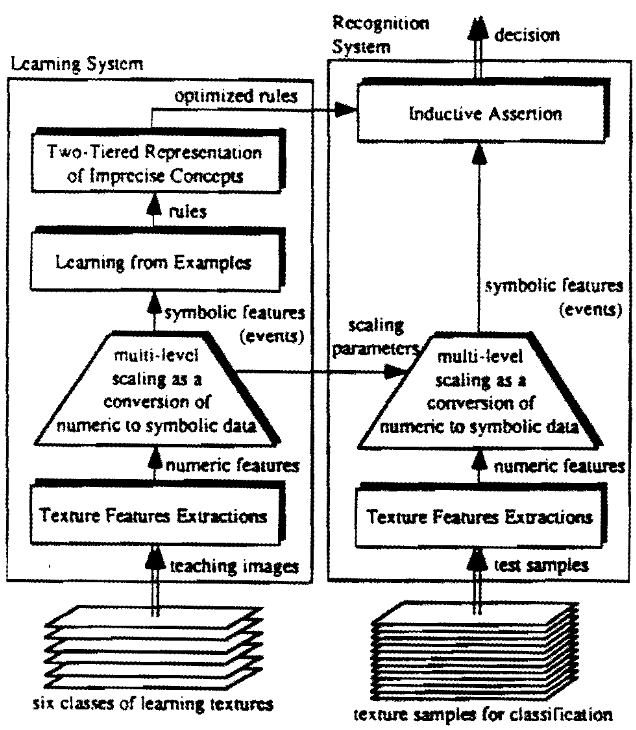


Fig.1 Open-loop architecture for texture recognition

2.4.2 Results and their interpretation

The best results (see Table 1) were obtained for eight Laws' masks. The average recognition was equal to 91% and the system recognized all test textures correctly. The highest recognition rate was equal to 98% and the lowest recognition rate was equal to 83%. The averaged deviation ($d = 1/N \sum |\bar{x} - x_i|$) was the lowest and equal to 4.0 . The experiments that applied texture classes description using inferred non-optimized rules shew the average, highest and lowest recognition rates equal to 80%, 93% and 72%, respectively. The averaged deviation was greater and equal to 7.0. In comparison, the average, highest and lowest recognition rates for the k-NN (k=10) pattern recognition method were equal to 70%, 99% and 40%, respectively. The averaged deviation was greatest and equal to 16.3. Main conclusions from the experiment are as follow:

1. The application of ML methodology increases the recognition effectiveness and optimizes the description of texture concept.
2. The improvement of recognition effectiveness possesses specific character; i.e. there is the increase of recognition rates for those classes that were classified before with lower rates, and the decrease of recognition rates for those classes that were recognized before with very high recognition rates.
3. Optimal rule description smooths recognition rates; i.e. the deviation of the recognition rate is reduced.

In summary, the most precious advantage of the applied ML approach is a four fold decrease of the deviation of the recognition rate. This effect makes the system more stable and the texture recognition more uniform, because the recognition rates are on the similar level of effectiveness for each texture class.

3. ADAPTIVE TEXTURE RECOGNITION

3.1 Introductory assumptions

The creation of the adaptive system for the recognition of natural objects depends on the nature of data and the robotics domain. We define the input data to the system as a series of images obtained from the moving robot. Thus, the variability of 3-D texture projected onto 2-D images is affected by the resolution, lighting and surface orientation. Considering robot mobility through the environment, an image can contain unknown texture, a texture that has been learned before, and an unknown variation of texture. All these cases of the texture appearance can be simultaneously observed on a single image.

Since, a single image contains more than one texture, the recognition and image segmentation into homogeneous areas is required. Considering autonomous robotics, these processes must work in an unsupervised mode. However, running a system without any initial training or verified knowledge is either impossible or inefficient. We assume that the system must be trained before the execution of a given mission, at least to verify background and domain specific knowledge about most characteristic and important texture that can be observed in the environment. The background knowledge represents a general concept of texture and environment, and also a principle of concept acquisition, transformation, and classification. This knowledge allows the system to perform the following tasks: (i) to acquire a concept of unknown texture, (ii) to acquire variations of the known texture concept, (iii) to optimize/modify concepts, and (iv) to verify the concept description. We apply symbolic learning methodology to perform such tasks that gives us advantages in manipulation of concept description. Since, a texture concept includes both numeric and structural texture features, the symbolic learning must be also able to learn from the numeric data.

A closed-loop system architecture is proposed to support system adaptability (see Figure 2). Such architecture is well suited for the incremental learning and the verification of learned descriptions. Considering the speed of system performance, there is a time restriction for the recognition and for the control of adaptation. These two modules must fulfil near real-time performance. Speed requirement for the learning module is less restrictive. Even a human needs time to learn new concepts and to verify them. Because symbolic learning algorithms are computationally very expensive, some modifications of them are necessary.

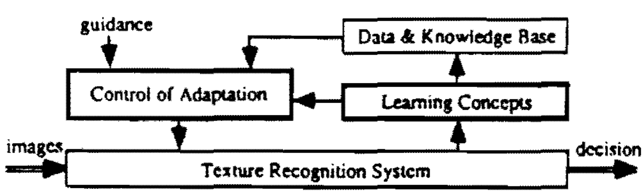


Fig.2 Closed-loop architecture for adaptive texture recognition

3.2 Learning tools

Learning tools that we apply to develop our adaptive texture recognition system consist of programs representing three groups of learning techniques: conceptual clustering, learning from examples, and learning flexible concepts.

Conceptual clustering is a form of unsupervised learning applied to order given observations into a hierarchy of classes [26]. A single class represents a concept using structural attributes and relations rather than a group of objects created using predefined numeric measure of similarity. Conceptual clustering algorithm generates conceptual descriptions of object classes, and then classifies these objects using created descriptions. An important aspect of this approach is the distinction of background knowledge and its role in the learning class descriptions. This background knowledge consists of classification goals, evaluation criteria, deductive and inductive inference rules. The Goal Dependency Network of the background knowledge, represents general-purpose and domain-specific background knowledge. The general-purpose knowledge contains fundamental constraints and classification criteria. The domain-specific knowledge contains inference rules to guide the process to the goal of classification.

Learning from examples is "a process of acquiring knowledge by drawing inductive inferences from teacher- or environment-provided facts" [14]. The process applies a heuristic search through the feature space incorporating generalization, transformation, correction and refinement of knowledge representation. Input data to the inductive learning is composed of preclassified sets of teaching examples. Theory of inductive learning assumes that these sets must be consistent. This is a very strong assumption because texture data is very noisy and appears under different variations. The inductive processes generate concept description from teaching data using background knowledge. This background knowledge consists of information about attributes, preference criteria, inference rules, heuristics, and program dependent procedures. The result of inductive learning is a rule description of object classes.

Learning flexible concept has been proposed [2,18] to deal with imprecise concepts of fluid boundaries and context-dependent meaning. The idea of flexible concept assumes the creation of two-tiered description of a concept: the Base Concept Representation (BCR) and the Inferential Concept Interpretation (ICI). The BCR represents typical properties of a concept inducted from examples and optimized by the application of generalization and specialization operations. In this way, the BCR contains a common characteristic of a concept; i.e. it does not include the variations of concept appearances. The second tier (ICI) contains allowed concept modifications and has been proposed as a procedural flexible matching. Such procedure represents such training examples that have not been included in the BCR as a typical concept description. For example, a typical property of tables is the horizontal position of the top of the table. However, a drawing table can be characterized by the screw position of the table's top. The implementation of the learning flexible concept to the texture recognition problem has been done successfully [20] (see section 2.4) in the part of concept optimization; i.e. extraction of the BCR.

4. LEARNING-BASED VISION MACHINE

4.1 Global architecture

In Figure 3, we present the position of the adaptive texture recognition and segmentation system within an integrated "learning-based vision machine". This vision machine consists of the following modules used to process and analyze image data: image understanding modules (color, texture, depth, and contour), data fusion module, 3-D modeling system, and a module of planning environmental content.

The texture module is one of dedicated modules designed to process certain class of data and to derive information necessary for the higher fusion and modeling. The output data from the texture module is fused with data from other modules of image understanding, and the final decision is sent to the 3-D modeling module. This final decision is also analyzed on the same level (by fusion module) comparing the initial decisions received from the contour, color, texture and depth modules. The verification of these decisions is fed back to improve system performance and to arrange possible exchange of information between these modules. We assume that mutual connections between distributed processes (not seen in Figure 3) can be very useful to improve unsupervised segmentation and recognition within a texture module. For instance, color, contour and depth information can help to cut texture area into smaller areas that will be grouped later based on their characteristics. The module of planning environment content is used to manage the image understanding and data fusion processes.

4.2 Active Perception and Control

Active Perception (AP) is an integrated element of our vision machine. We strongly agree that the AP is a kind of modeling and creation of control strategies for perception [1], while perception in that sense is intelligent data acquisition and understanding. In our approach to the AP, we focus on the intelligent data understanding. Implementing AP within the

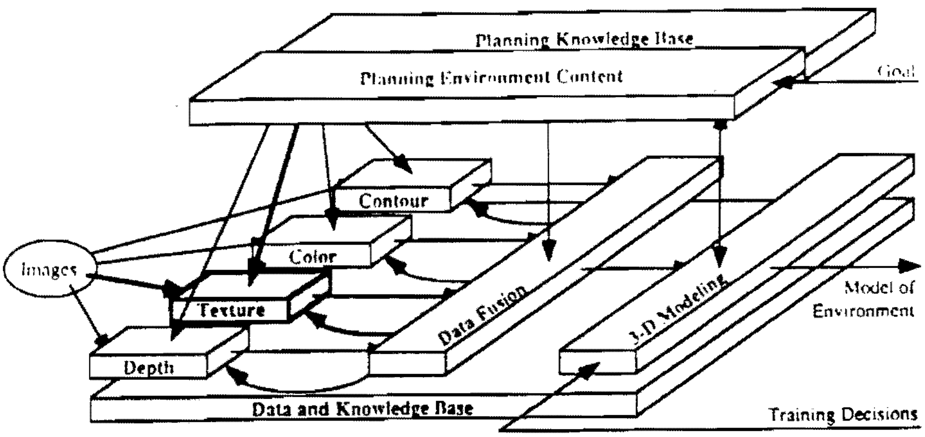


Fig.3 Learning-Based Vision Machine

system, we integrate both ML that allows acquisition of new concepts and experience, and also the theory of large scale control systems [10] for building complex dynamic and adaptive systems including tuning, verification and guidance processes.

Control strategies within the system presented in Figure 3 include the following tasks: (i) to arrange a feedback verification of decision making and learning processes performed by the data understanding modules - this verification is provided from the data fusion module, (ii) to arrange the cooperation between distributed modules of texture, color, depth and contour understanding to support and improve unsupervised learning of new concepts, and (iii) to develop a high-level guidance using planning the environment content.

Planning module has a special purpose within the presented system. Since planning is a kind of prediction, the content of an image can be modeled in advance. If the system predicts content of the environment, then it can use simpler and faster image processing and analysis to verify this prediction. So, the frequency of complete scene understanding processes can be decreased depending on the vision task, content of the environment, motion parameters, and unexpected differences between the current visible and predicted scene. Such reduction of interactions can be used by the system to learn new concepts and to provide adaptation to the environment. The planning module provides control parameters to the image understanding modules in order to optimize their performance and to implement the task-driven balance between image understanding and adaptation processes. Planning in the texture recognition problem is also applied to control system memory divided into segments representing descriptions of: currently visible textures, highly possible textures for a given class of environment, textures representing dangerous objects (like sand, water), and

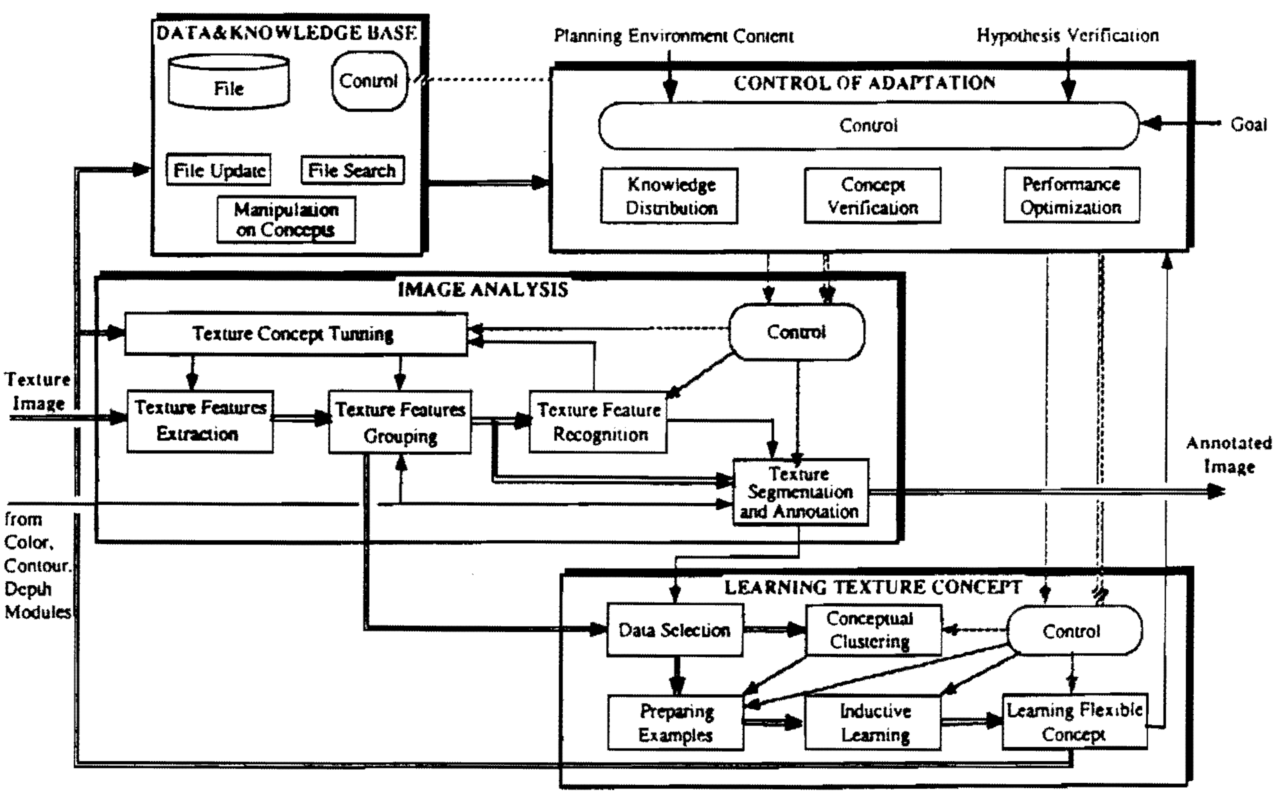


Fig. 4 System architecture for learning-based adaptive texture recognition and segmentation

acquired unknown textures. Let us suppose there is a texture that has not been recognized for a given resolution: during the movement of a vehicle the system will derive a description of this texture for other resolution and will modify the description dealing with texture variable concept. If this texture is suddenly recognized (e.g., because the resolution was decreased to such a one that the object was learned before), then the system will address stored descriptions for the merging task.

5. ADAPTIVE TEXTURE RECOGNITION and SEGMENTATION

The development of the machine learning approach to the robust texture recognition problem is based on the closed-loop principle. The system architecture is presented in Figure 4, and it consists of the four modules described below.

5.1 Image Analysis

The image analysis module performs (i) the extraction and grouping of texture features, (ii) texture recognition, and (iii) image segmentation into homogeneous areas. Texture feature extraction is based on the concept of multi-level repetitive patterns, in which elements or primitives are arranged according to certain placement rules. The extraction of texture concept incorporates a hierarchical representation of numeric and micro-structural texture features. Numeric features represent texture statistics, while micro-structural features represent spatial relationship. Numeric features are converted into their multi-value symbolic representation. The conversion process applies multi-level scaling of numeric data into their symbolic intervals [20]. If a texture has a micro-structural characteristics, then elements of local structure are combined into structural patterns (like blobs, triangles, rectangles), and are characterized in relation to each other. Extracted elements are grouped using background knowledge into the higher-level features and patterns.

The system knows how to extract texture elements and how to group them into patterns, because such knowledge is included within a rule description of an already learned texture concept. For unknown texture, special tuning is arranged to extract only most significant features and groups of them for the learning step. This tuning incorporates feedback influence on early processes of texture transformation provided both by the recognition unit that directly evaluates the recognition results, and also by the learning module that asks for the verification of learned descriptions. There are the following two advantages of such feedback interaction: (i) operator parameters are tuned to extract and group texture areas, and (ii) the number of operators is decreased to optimize processing time. Since this feedback mechanism must support the variability of texture concept, chosen operators and their parameters must cover a wide range of texture variable appearances instead of a particular point in the parameter space.

The texture recognition unit performs inductive assertion on the multi-level texture features and patterns using a rule description of texture concepts. These descriptions are provided by the control unit and they are selected from the data and knowledge base by the control module of system adaptability. Finally, texture segmentation is performed on the annotated texture features stepping up through the pyramid of texture areas [25].

5.2 Learning Texture Concept and Description

Texture features and some groups of them are transferred into the learning module. At first, the selection unit organizes multi-level features as the input data to the conceptual clustering unit by separation from already recognized texture area and by decreasing the resolution. Conceptual clustering [26] supports unsupervised learning of a multi-level concept of a single example incorporating background and domain specific knowledge. During this learning step, higher-level features like structural patterns are extracted. For example, structural edges can be organized into a triangle [28], and this triangle represents a new feature. This is a kind of data self-organization and learning higher-level features. Next, teaching examples are process to find out homogeneous areas. We will develop a reasoning system to deal with the problem of unsupervised separation of homogeneous areas based on the description and structure of neighboring examples. Roughly organized groups of examples are the input to the learning of their description. The learning from examples, creates a rule description of acquired texture subarea from provided examples [14]. We assume that the system will be working incrementally, and for a single pass it will acquire a description of only one small area of texture. The module of learning flexible and optimal concept applies the idea of two-tiered representation of imprecise concepts [2,18] to transfer an already inferred description into its two components; i.e., the BCR and the ICI (see section 3.3). The first component contains common characteristics of a concept, while the second component contains allowed concept modifications. Such two-tiered concept representation is more optimal, supports concept variability and filters the noise [18,19].

5.4 Data and Knowledge Base

The module of data and knowledge base is responsible both for the knowledge storage, and also for the manipulation of concepts in order to merge, split and update them. We separate such manipulation of texture concepts from the learning module, where these concepts have been acquired. This manipulation is caused by the dynamics of external influences (e.g. lighting, weather), system movement through the environment (e.g. changing resolution, surface orientation), and the appearance of unknown textures that represent either new objects or an unknown variation of a known texture. Such manipulation is performed on the two-tiered representation of the flexible concept as an optimized rule description. We implement both, forgetting mechanisms that protect the knowledge base against the overloading, and also a special storage of concept descriptions in the base for their optimal selection.

5.5 Control of Adaptation

The connections between above described modules of the texture recognition system are determined by the system hierarchical structure. Considering external connections between the texture system and other systems presented in Figure 3, we design a higher-level control module for the adaptation of the texture recognition system. This module pays a key role in the knowledge selection and distribution. It controls the acquisition of a new concept, the verification of generated output annotated image and classification decisions, and the optimization of system performance. This control incorporates the top-down system guidance and verification. It analyses a vision task, input modeling data from the module of planning environment content, and fed-back evaluation of the recognition and segmentation decisions provided from the fusion module. It also controls the cooperation (data exchange) of the texture system with other distributed modules of contour, color and depth processing.

6. SYMBOLIC LEARNING for ENGINEERING SYSTEMS

6.1 Requirements for engineering systems

We have already explained tools of symbolic learning applied in the first experiments and included within the learning-based architecture of the final system. Currently, we point out engineering requirements to stress the conditions in which symbolic learning and system adaptation must be performed.

First of all, the *speed* of learning algorithms must allow the system to work in the near real-time regime. It means that the learning capabilities of the system cannot limit system performance in such degree that the system would spend most time on the adaptation instead of on the task execution. The performance and frequency of activated adaptation must be balanced due to the top task. Specification of such task should include a parameter that would be used to configure the adaptation process in the balance with the task execution process. For example, the adaptation pays a secondary role in the case of an emergency task (e.g. "run out of the area as fast as possible") but it pays a crucial role for such a task as terrain exploration.

The next requirement, system *robustness*, is caused mainly by the robotics and data domain. Adaptation mechanism must be created in such a way that it will allow the system to work even under unexpected circumstances. For example, if two very similar textures characterize different objects, then the control of adaptation must be capable of increasing the sensitivity of the concept acquisition and learning processes. One of the solutions is the application of multi-level control and the implementation of *closed-loop cooperation* between different system elements.

The third requirement deals with texture domain. The system architecture and applied methodologies must support *noisy data*, *incomplete data*, and the *variability of texture*. Since our approach combines numeric and microstructural texture features, the system must deal with numeric and structural noise. Because the noise can be propagated through the system hierarchy, the noise filtering must be applied in most system modules and on all levels of the system hierarchy.

Additionally, the system must work with the *incomplete list of texture classes*; i.e. the total number of classes cannot be defined or predicted. This requirement is very important for the support of unsupervised learning, when the system collects the rule descriptions of unknown textures continuously. The protection is required against such a situation when the number of texture classes is to large to manage the knowledge base and to effectively select applicable rule descriptions. The system has to look permanently for merging concepts corresponding to the same texture but obtained for different external conditions.

6.2 Selected modifications

Dealing with the above requirements, we develop learning tools and modifications of already used algorithms. We assume that all learning programs must be designed for *parallel execution* and must apply *noise filtering*. Early filtering, data selection and parallel execution will increase the execution speed. Moreover, the learning tools must be able to work with *noisy* data (because the filtering can be insufficient), *incomplete* and *inconsistent* data. Since texture concept possesses multi-level hierarchy, learning algorithms must be suited for such data representation. Each learning technique must be executable in a closed-loop *incremental* learning mode [22] that also gives the increase in For example, we implement the following selected modifications to the AQ inductive learning program [15]: (i) to learn a description of only one class during a single run and under control of a given accuracy coefficient, (ii) to acquire a parametric distribution of examples within a single cover (a rule is the disjunction of covers representing a cluster of examples, and a cover is the conjunction of selectors as attributes describing this cluster) used later for incremental modification and forgetting mechanisms, (iii) to select only those (to the set of learning positive events) descriptions representing the nearest classes of negative examples, (iv) to apply parametric fast filtering of teaching examples, and (v) to perform the learning task on parallel processors.

6.3 Integrated control system

The system architecture and hierarchical control (see Figure 5) illustrate the integration of various techniques within a system. Such integration implies the development of theoretical and experimental studies for the creation of a new class of control and learning systems; i.e., systems that can learn how to control. In the area of symbolic learning, we integrate together synthetic learning techniques [17]; i.e., conceptual clustering, learning from examples and learning flexible concept.

Multistrategy learning [17] as the integration of various learning methods and techniques within our system also includes genetic algorithms as a tool applied to find the optimal low-level operators to extract and group texture features. Since genetic algorithms methodology [5] works by the execution of tens to hundreds iterations (where the execution of a single loop can be time consuming), we are looking for modifications of genetic algorithms applying elements of guidance. On the other hand, we also consider an application of neural networks in the early phase of image processing and texture feature extraction. We incorporate the theory of large scale systems [10] to arrange levels of control and the connections of the texture module with other modules.

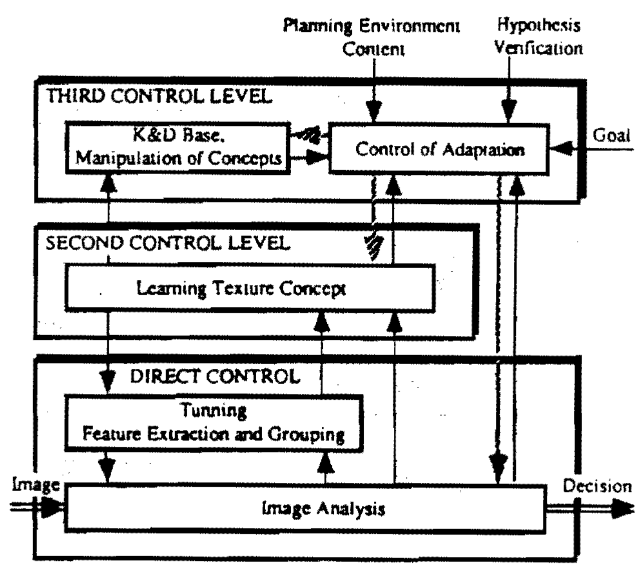


Fig. 5 Control levels of adaptive texture system

7. SUMMARY

This paper has presented our approach to the creation of a robust texture recognition system for robot navigation in natural environment. The learning-based system architecture has been proposed to support great variability of texture concept caused by changing resolution, lighting conditions, and surface orientation. Proposed architecture integrates tools of different areas of computer science and systems engineering. We build the hierarchy of the system based on the theory of large-scale control systems, where each level of the hierarchy includes elements of artificial intelligence and machine learning. Conducted research and the creation of the system bridges different methodologies and tools of systems adaptability. We believe that such an approach to adapt to variable characteristics of the environment and to acquire new knowledge through learning, reasoning, experimentation and verification will increase the autonomy of intelligent systems. Presented adaptation is required to support the execution of a given mission by an autonomous system.

Acknowledgement: The author wishes to thank Ryszard Michalski, Harry Stephanou and Janping Zhang for their valuable comments. This work was done at 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, administered 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.

8. REFERENCES

(1) Bajcsy, R., "Active Perception", Proc. IEEE, Vol.76, No.8, pp.996-1005, 1988.

(2) Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J., "Learning Flexible Concept Descriptions Using a Two-tiered Knowledge Representation: Ideas and a Method", Report No.88-4, Artificial Intelligence Center, George Mason University, 1988.

(3) Coleman,G.B. and Andrews, H.C., "Image Segmentation by Clustering", Proceedings of the IEEE, Vol.67, No.5, pp.773-785, 1979.

(4) Cross, G. R. and Jain, A. K, "Markov Random Field Texture Models", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.PAMI-5, No.1, pp.149-163, 1983.

(5) DeJong, K., "Learning with Genetic Algorithms: An Overview", Machine Learning, Vol.3, pp.121-138, 1988.

(6) Haralick, R.M., Shanmugan, K. and Dinstein, I., "Texture features for image classification", IEEE Trans. System, Man and Cybernetics SMC-3, pp.610-621, 1973.

(7) Hawkins, J. K.,"Textural Properties for Pattern Recognition", in Picture Processing and Psychopictorics, B.C. Lipkin and A. Rosenfeld (eds.), Academic Press, New York, pp.347-370, 1970.

(8) Hougen, D. R., "Visual Texture Recognition Using Information Trees", UILU-ENG-88-1734, Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, 1988.

(9) Hsiao,J.Y. and Sawchuk, A., "Unsupervised Textured Image Segmentation Using Feature Smoothing and Probabilistic Relaxation Techniques", Computer Vision, Graphics and Im. Processing, Vol.48, pp.1-21, 1989.

(10) Jamshidi, M., Large-Scale Systems: Modeling and Control, North-Holland, 1983.

(11) Kanatani, K.-i. and Chou, T.-C., "Shape from Texture: General Principle", Artificial Intelligence, Vol.38, pp.1-48, 1989.

(12) Laws, K.I. "Textured Image Segmentation", PhD Thesis, Dept. of Elecrical Engineering, Univ. of Southern California, Los Angeles, 1980.

(13) Lawton, D.T., Levitt, T.S., McConnell, Ch.C., Nelson, Ph.C. and Glicksman, J., "Environmental Modeling and Recognition for an Autonomous Land Vehicle", Proc. DARPA Image Understanding Workshop, pp.107-121, 1987.

(14) Michalski, R.S., "A Theory and Methodology of Inductive Learning", Artificial Intelligence, Vol.20, pp.111-161, 1983.

(15) Michalski, R.S., Mozetic, I., Hong, J., Lavrac, N., "The AQ15 Inductive Learning System: An Overview and Experiments", ISG 86-23, UIUCDCS-R-86-1260, Department of Computer Science, University of Illinois, Urbana, 1986.

(16) Michalski, R.S., "Two-Tiered Concept Meaning, Inferential Matching and Conceptual Cohesiveness", in "Similarity and Analogy", S. Vosniadou and A. Orton (Ed.), 1987.

(17) Michalski, R.S., "Multistrategy Constructive Learning: Toward a Unified Theory of Learning", Proc. Office of Naval Research Workshop on Knowledge Acquisition, Arlington, VA, Nov. 1989.

(18) Michalski, R.S., "Learning Flexible Concept: Fundamental Ideas and a Method Based on Two-tiered Representation", Machine Learning, Morgan Kaufman, 1990 (in press).

(19) Pachowicz, P.W., "Integrating Low-Level Features Computation with Inductive Learning Techniques for Texture Recognition", International Journal of Pattern Recognition and Artificial Intelligence, 1990 (in press).

(20) Pachowicz, P.W., "Low-Level Numerical Characteristics and Inductive Learning Methodology in Texture Recognition", Proc. IEEE International Workshop on Tools for AI, Fairfax, VA., pp.91-98, 1989.

(21) 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.

(22) Reinke, R.E. and Michalski, R.S., "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.

(23) Roan, S. J., Aggarwal, J. K. and Martin, W. N., "Multiple Resolution Imagery and Texture Analysis", Pattern Recognition, Vol.20, No.1, pp.17-31, 1987.

(24) Rosenfeld, A. and Kak, A., Digital Picture Processing, Academic Press, New York, 1976.

(25) Sher, C.A. and Rosenfeld, A., "Detecting and Extracting Compact Textured Regions using Pyramids", Image and Vision Computing, Vol.7, No.2, pp.129-134, 1989.

(26) Stepp, R.E. and Michalski, R.S., "Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects", in Machine Learning: An Artificial Intelligence Approach, Vol.II, Morgan Kaufman, pp.471-498, 1983.

(27) Unser, M. and Eden, M., "Multiresolution Feature Extraction and Selection for Texture Segmentation", IEEE Trans. Pattern Analysis and Mach. Intell., Vol.PAMI-11, No.7, pp.717-728, 1989.

(28) Vistnes, R., "Texture Models and Image Measures for Texture Discrimination", International Journal of Computer Vision, Vol.3, pp.313-336, 1989.

(29) Weska, J. S., Dyer, C. R. and Rosenfeld, A., "Comparative Study of Texture Measures for Terrain Classification", IEEE Trans. Syst. Man and Cybernetics SMC-6, pp.269-285, 1976.

(30) Zhang, J., Michalski, R.S., "Rule Optimization via SG-TRUNC Method", Proc. of the Forth European Working Session on Learning, Pitman, Morgan Kaufman, pp.251-262, Montpellier, Dec. 1989.