# The MIST Methodology and Its Application to Natural Scene Interpretation[1]

**R.S. Michalski*, Q. Zhang, M.A. Maloof and E. Bloedorn**

Machine Learning and Inference Laboratory
George Mason University
Fairfax, VA 22030-4444
{michalski, qzhang, maloof, bloedorn}@aic.gmu.edu
http://www.mli.gmu.edu
*Also with GMU Departments of Computer Science and Systems Engineering, and
the Institute of Computer Sciences, Polish Academy of Science.

## Abstract

The MIST methodology (Multi-level Image Sampling and Transformation) provides an environment for applying diverse machine learning methods to problems of computer vision. The methodology is illustrated by a problem of learning how to conceptually interpret natural scenes. In the experiments described, three learning programs were used: AQ15c—for learning decision rules from examples, NN—neural net learning, and AQ-NN—multistrategy learning combining symbolic and neural net methods. Presented results illustrate the performance of the learning programs for the chosen problem of natural scene interpretation in terms of predictive accuracy, training time, recognition time, and complexity of the induced descriptions. The MIST methodology has proven to be very useful for the presented application. Overall, the experiments performed indicate that the multistrategy learning program AQ-NN appears to be the most promising approach.

## 1. Introduction

The underlying motivation of our research is that vision systems need learning capabilities for handling problems for which algorithmic solutions are unknown or very difficult to formulate. Learning capabilities will also make vision systems more easily adaptable to different vision problems, and more flexible and robust in handling the variability of perceptual conditions[Michalski et al., 1994] .

To this end, we have developed a general methodology for applying machine learning methods to vision problems, called *Multilevel Image Sampling and Transformation*. (MIST). The purpose of this methodology is to provide a researcher with an environment in which a variety of machine learning methods and approaches can be flexibly applied to a wide range of vision problems. The methodology makes it easy to apply a variety of tools developed in computer vision and machine learning research. The central idea of the methodology is to combine advanced inductive learning techniques and the use of background knowledge in parallel multi-level image interpretation.

The MIST methodology is an extension and a generalization of an earlier methodology, called Multilevel Logical Templates, originally proposed by Michalski (1973), further developed and implemented by Channic (1988), and subsequently by Bala (1991) and Michalski et. al. (1993). Although developed independently, MIST's concept of an Annotated Symbolic Image is similar to the concept of a class map in the ALISA system (e.g., Howard and Bock, 1994).

[1] This research was supported by the Advanced Research Projects Agency under grants F49620-92-J-0549 and F49620-95-1-0462, administered by the Air Force Office of Scientific Research. The development of machine learning systems used in this research was supported in part by the Advanced Research Projects Agency under grant No. N00014-91-J-1854, administered by the Office of Naval Research, in part by the Office of Naval Research under grant N00014-91-J-1351, and in part by the National Science Foundation under grants IRI-9020266 and CDA-9309725.

This paper briefly describes the methodology and illustrates it by an application to natural scene interpretation. As pointed out in [Fischler and Strat 1988; Strat and Fischler 1991], the semantic interpretion of natural scences and recognition of natural objects is one of the most challenging open vision problems. The MIST methodology seems to offer a new approach to these problems.

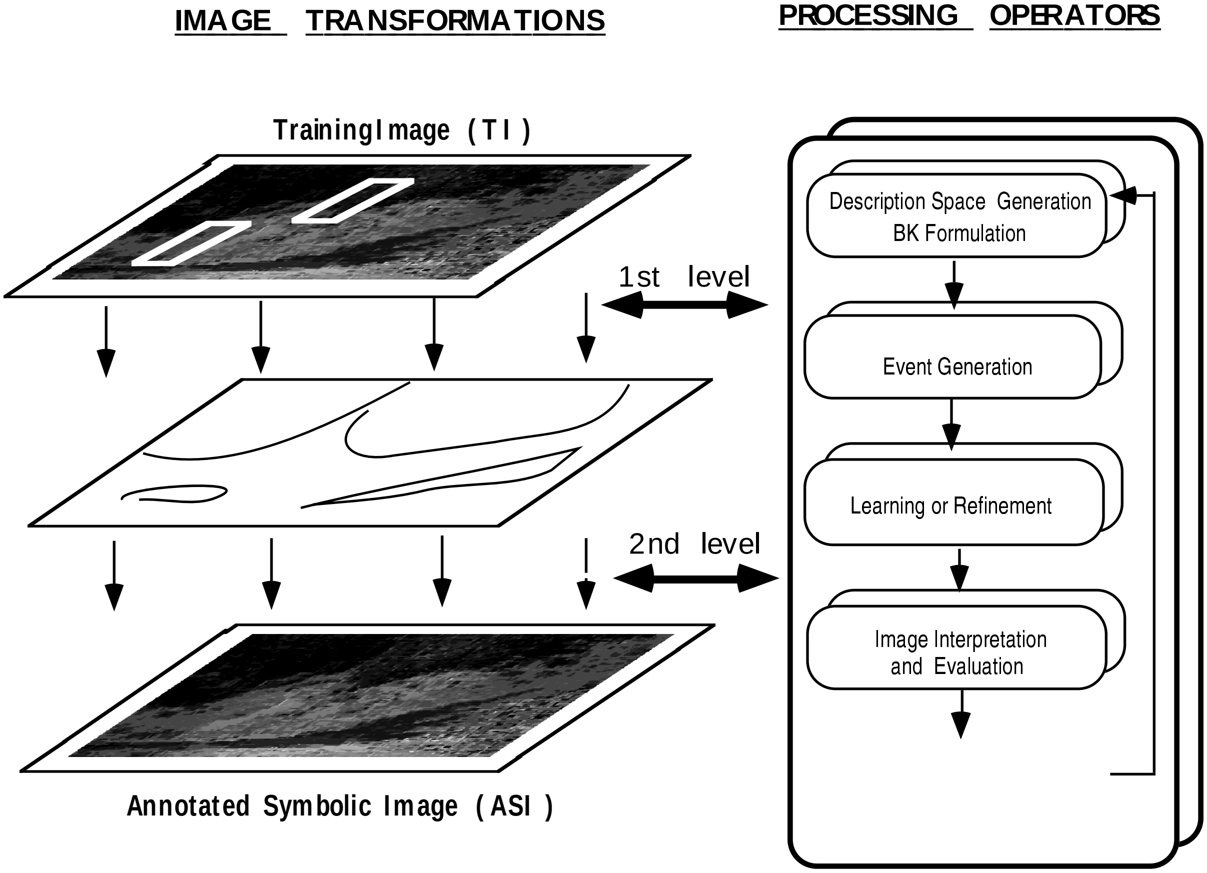
## 2. MIST Methodology

The MIST methodology works in two basic modes: *Learning mode* and *Interpretation mode*. In Learning mode, the system builds or updates the Image Knowledge Base (IKB) that contains concept descriptions, and the background knowledge relevant to image interpretation. A descritpion (or model) of a visual concept is developed by inductive inference from concept examples specified by a trainer. Concept descriptions are arranged into procedures defining sequences of image transformation operators.

In Interpretation mode, a learned (or predefined) image transformation procedure is applied to a given image in order to produce an *Annotated Symbolic Image* (ASI). In an ASI, areas that correspond to the location of recognized concepts in the original image are marked by symbols (e.g., colors) denoting these concepts, and linked to *concept annotations* (text containing additional information about that concept, such as, degree of certainty of recognition, properties of the concept, relation to other concepts, etc.). The following paragraphs describe these two modes in a greater detail.

### A. Learning Mode

This mode (Figure 1) is executed in four phases: LP1—Description Space Generation and Background Knowledge Formulation, LP2—Event Generation, LP3—Learning or Refinement, and LP4—Image Interpretation and Evaluation.



These four phases may be repeated iteratively creating images at different levels (Figure 1 shows just two levels).

*LP1: Description Space Generation and Background Knowledge Formulation*

A trainer assigns concept names to areas in the image(s) that contain objects (concepts) to be learned. These areas are divided into training and testing areas. Objects to be learned are presented in different poses and with different appearances (by changing perceptual conditions) so that the system can learn a description that is invariant to concept-preserving transformations. The trainer also defines the initial description space, i.e., initial attributes and/or terms to be measured on image samples, specifies their value sets and their types (measurement scale). This phase also involves an optimization of the *image volume*, that is, a reduction of the image resolution and intensity levels (the hue and saturation in color images) accordingly to the needs of the given problem.

The trainer may also define constraints on the description space, initial concept recognition rules, and possibly forms for expressing the descriptions (e.g., conjunctive rules, DNF, the structure of the neural net, etc.). Procedures for the measurement of attributes/terms are selected from a predefined collection.

*LP2: Event generation.*
Using chosen procedures, the system generates initial training examples ("training events") from each concept area. Concept areas are sampled exhaustively or selectively.

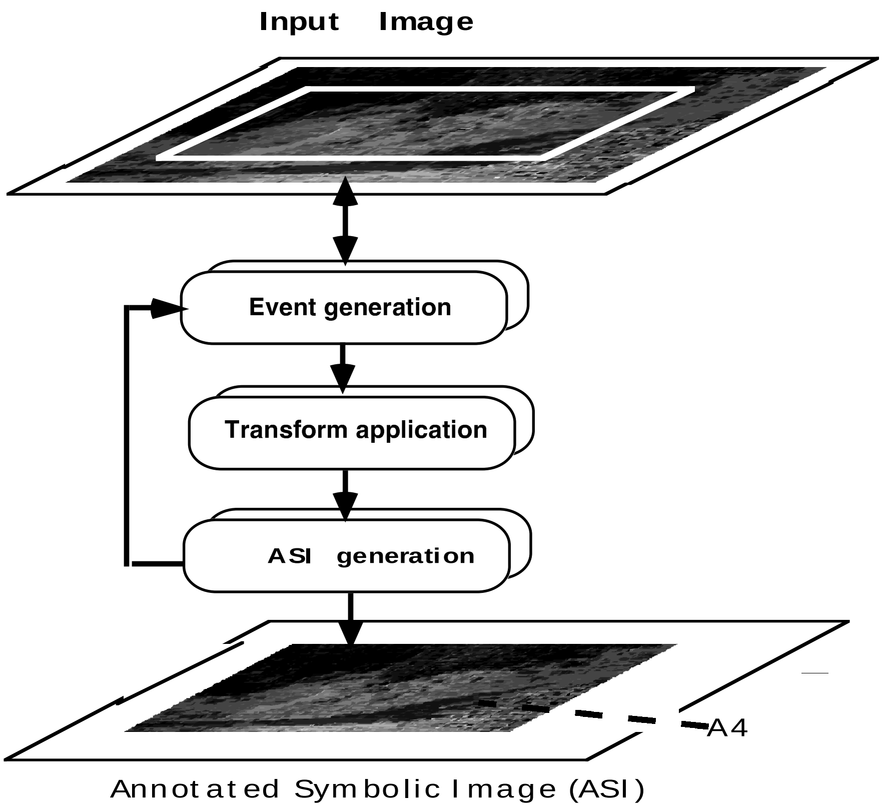
*LP3: Learning or Refinement*
The system applies a selected machine learning program to training examples to generate a concept description. Currently, we have the following programs available: AQ15c—for learning general symbolic rules from examples, NN—a neural net learning with backpropagation, and AQ-NN—a system that integrates AQ rule-learning with neural net learning[Zurada 1992].

*LP4: Image Interpretation and Evaluation.* The developed descriptions are applied to the testing areas to generate an *Annotated Symbolic Image* (ASI). In ASI, the areas corresponding to given concepts are marked by symbols representing these concepts (numbers, colors, etc.). These areas are also linked to texts that include additional information about concept descriptions. The quality of generated descriptions is determined by comparing the ASI with testing areas in the original image. Depending on the results, the system may stop, or may execute a new learning process (iteration), in which the ASI is the input (hence the term "multilevel" in the name of the methodology). If the generated descriptions need no further improvement, the process is terminated. This occurs when the obtained symbolic image is "sufficiently close" to the target image labeling (indicating the "correct" labeling of the image). Complete object descriptions are sequences of image transformations (defined by descriptions obtained in each iteration) that produce the final ASI. Learning errors are computed by comparing the target labeling (made by the trainer) with learned labeling (produced by the system).

## B. The Interpretation Mode

In this mode (Figure 2), the system applies descriptions from the Image Knowledge Base to semantically interpret a new image. To do so, the system executes a sequence of operators (defined by the description) that transform the given image into an ASI.

A given "pixel" in ASI is assigned a class on the basis of applying operators to a single event, or to a sample of events and applying a majority voting schema (typically within a 3x3 window). In ASI, different concepts are denoted by different colors and/or textures. The simplest form of annotation is to associate the degree of confidence with the ASI pixels denoting a given concept.


Input Image
Event generation
Transform application
ASI generation
A4
Annotated Symbolic Image (ASI)

descriptions was in the form of decision rules, which were discovered by the inductive learning program AQ15 [Michalski et al., 1986] and represented in the $VL_1$ logic-style language (Variable-Valued Logic System 1) [Michalski, 1973]. Such decision rules can be applied to an image in parallel or sequentially.

## 3. Experimental Results

A simple version of MIST methodology was applied to problems of semantically interpreting outdoor scenes using several learning methods. In the experiments, we used a collection of images representing selected mountain scenes around Aspen, CO (Fig. 3).

The input to the learning process was a training image in which selected examples of the visual concepts to be learned have been labeled by a trainer, for example, trees, sky, ground, road, and grass. We experimented with different sets of attributes defining the description space, with images obtained under different perceptual conditions, with different sizes of training areas, and different sources of training and testing image samples (from different parts of the same image area, from different areas of the same image, from different images).

In the experiments described here, the description space was defined by such attributes as: hue, saturation, intensity, horizontal and vertical lines, high frequency spot, horizontal and vertical V-shape, and Laplacian operators. These attributes were computed for the 5x5 windowing operator (sample size) that scanned the training area. Vectors of attribute values constituted training events. Three learning methods were used: AQ15c, AQ-NN, and NN. Three different training areas were used: 10 x 10, 20 x 20, and 40 x 40 pixels. The validation methodology used here was a hold-out method in which a random selection of 60% of the samples from the training area were used for training, while the remaining 40% were used for testing[Weiss and Kulikowski, 1992].

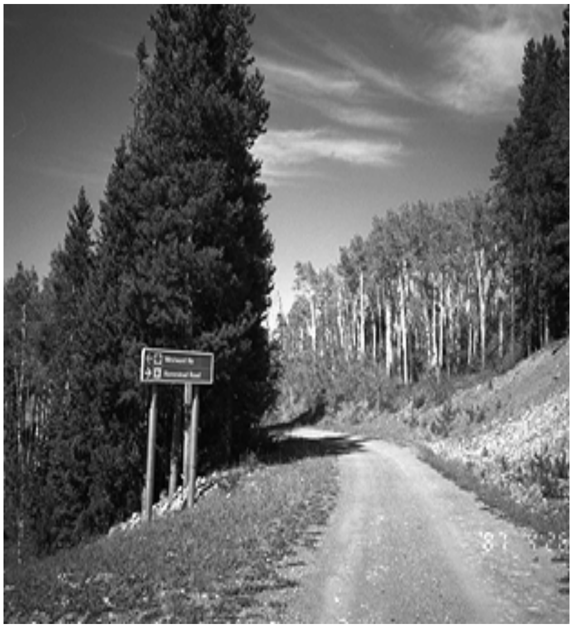


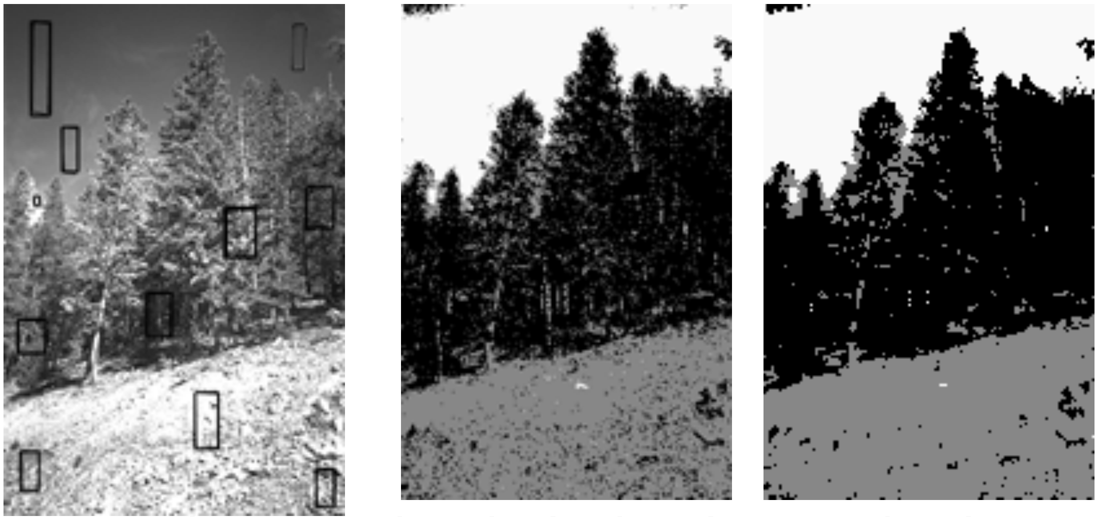
*Figure 3:* A typical image of a natural scene used in the experiments.

Table 1 gives results from an experiment involving only one level of image transformation using different learning programs. In this experiment, the training area for each concept was only 10x10 pixels. When the training area was enlarged to 20x20, the training time was significantly longer, but the correctness of the interpretation of the areas of the whole image was approximately the same.

| | **Learning Method** | | |
|---|---|---|---|
| | AQ15c | AQ-NN | NN |
| **Training time** | 0.43 s | 10.93 s | 4.38 s |
| **Recognition time** | 1.0 s | 0.016 s | 0.033 s |
| **Accuracy** | 94.0% | 99.98% | 99.97% |

(Statistics computed for 161 training events, 150 testing events selected from the 10 x 10 training area.)

*Table 1*: A summary of results from learning to interpret the image in Figure 4.



(a) An image with training area for sky, tree, and ground.
(b) ASI based on the single-event evaluation scheme.
(c) ASI obtained using a majority voting scheme.

Concept denotation: ■ tree area □ sky area ▨ ground area

*Figure 4.* An example of the image interpretation process based on the rules learned from the indicated training areas.

Figure 4 presents an example of a training image and ASIs (annoted symbolic images) obtained from applying the learned one-level descriptions to the whole image using two different evaluation schemes.

As one can see in Figure 4c, most of the areas in the image were correctly interpreted, although the system learned concept descritpions from relatively small training areas (Figure 4a). In this experiment, AQ-NN produced a slighly smaller neural net and the interpretation time of the image was about 50% shorter than with NN method.

We also tested the application of the data-driven constructive induction method, AQ17-DCI, in this experiment and got some new attributes and comparable results [Bloedorn et al., 1993].

Table 2 presents a summary of the performance accuracy of the descriptions obtained by AQ15c and AQ-NN. The AQ15c program was run on a Sparc 2 workstation and AQ-NN on Sparc2 and MATLAB neural network toolbox.

| **Learning System** | **Recognition accuracy**<br>Single event scheme | **Recognition accuracy**<br>Majority voting scheme |
|---|---|---|
| Symbolic learning: AQ15c | 89% | 96% |
| Multistrategy learning: AQ-NN | 91% | 99% |

*Table 2.* A comparison of recognition rates from symbolic learning (AQ15c) and multistrategy learning (AQ-NN).