PROGRESS ON VISION THROUGH LEARNING[†]: A Collaborative Effort of George Mason University and University of Maryland

R.S. Michalski[1,3], A. Rosenfeld[2], Y. Aloimonos[2], Z. Duric[1,2], M.A. Maloof[1], Q. Zhang[1]

[1]Machine Learning and Inference Laboratory, George Mason University
Fairfax, VA 22030–4444

[2]Computer Vision Laboratory, University of Maryland
College Park, MD 20742–3275

[3]Also with GMU Departments of Computer Science and Systems Engineering, and
the Institute of Computer Science, Polish Academy of Sciences.

Abstract

This report briefly reviews research progress on vision through learning conducted as a collaborative effort of the GMU Machine Learning and Inference Laboratory and the UMD Computer Vision Laboratory. The report covers work done on the following projects:

(1) The Multi-level Image Sampling and Transformation (MIST) methodology for learning image descriptions and transformations
(2) Applying the MIST methodology to semantic analysis of outdoor scenes
(3) Recognizing objects in a cluttered environment
(4) Learning in navigation
(5) Intelligent interfaces: Learning in the RADIUS environment
(6) Learning space configuration and homing
(7) Learning object functionality

Our work aims at ultimately developing vision systems that apply a range of symbolic and parametric machine learning methods to solving vision problems.

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 difficult to obtain. 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]. During the reported period we have studied the application of symbolic, neural net and multistrategy learning methods to such problems as outdoor scene interpretation, object recognition in cluttered environments, learning in navigation, homing, and intelligent interfaces that exploit experience in the RADIUS environment.

The following sections describe specific projects and results obtained during the reported period. Detailed results are presented in separate papers in these Proceedings [Aloimonos & Fermueller, 1996; Duric et al., 1996; Maloof et al., 1996; and Michalski et al., 1996].

2 The Multi-level Image Sampling and Transformation Methodology (MIST)

Recent research on the application of machine learning to computer vision has been concerned with a wide range of problems and explored a variety of learning methods. For example, Grimson, Horn and Poggio [1994] conducted theoretical and experimental studies of neural network learning. They used radial basis functions and applied their method to face recognition.

Houzelle, Strat, Fua and Fischler [1994] considered the problem of automatically selecting a feature extraction algorithm and its parameters on the basis of past experience with similar tasks.

[†] 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, CDA-9310297 and CDA-9309725.

Bhanu [1994] applied reinforcement learning and genetic algorithms to selected problems of computer vision, such as learning the parameters of the Phoenix segmentation algorithm. He used Gabor wavelets as image features. Allen, Boult, Kender, and Nayar [1994] described work on learning object descriptions in eigenspaces. The objects are represented on a 20-dimensional manifold and recognition is done by matching new instances of object descriptions to these manifolds.

Hanson, Riseman and Weiss [1994] described the Schema learning system. Their system consists of learning programs and image understanding routines. Fischler and Bolles [1994] described work on applying learning from experience to natural object recognition. Shafer, Kanade and Ikeuchi [1994] considered the problem of learning through observation. In their project, a robot learns a task by observing a human in action. Binford, Levitt and Langley [1994] explored problems of learning object models using observation and background knowledge. Rosenfeld, Rivlin and Khuller [1994] investigated problems of learning to navigate a graph. It has also been proposed to apply learning in the RADIUS environment, e.g., [Gerson and Wood, 1994; Strat and Climenson, 1994; Bailey et al., 1994; Sargent et al., 1994; Chellappa et al., 1993].

These efforts demonstrate a wide consensus in the vision community that vision systems need learning capabilities. An open problem is what type of learning method or approach is most effective for what types of vision problems. In this context, a significant part of our research has been concerned with developing a general methodology, called *multilevel image sampling and transformation* (MIST) for applying machine learning methods to vision problems, and investigating their performance. The purpose of this methodology is to provide a researcher with an environment in which diverse machine learning methods and approaches can be flexibly applied to a wide range of vision problems. The methodology makes it easy to compare the performance of different methods and to incorporate previously developed tools.

Specifically, MIST aims at learning image descriptions or other knowledge (e.g., image transformations, site modeling operators, etc.) from training examples (e.g., labeled image samples, object examples, parameter settings, etc.). In general, the learning process can be done in one global step or in a sequence of steps. In the case of multistep learning of image descriptions, each step produces a transformed image that serves as input to the next level transformation. The process stops when the obtained *annotated symbolic image* (ASI) is sufficiently close to the target image (representing a labeling of pixels according to the visual concepts to be learned).

MIST is an extension and a generalization of an earlier multilevel logical templates methodology, originally proposed by Michalski [1972, 1973], developed further and experimented with by Channic [1988], and subsequently by Bala and Pachowicz (see, e.g., [Bala et al., 1994]).

The MIST methodology works in two modes: *Learning mode* and *Interpretation mode*. The learning mode is executed in four phases (which may be repeated iteratively): LP1—Description space generation and background knowledge formulation, LP2—Event generation, LP3—Learning or refinement, LP4—Image interpretation and evaluation.

The result of learning is a sequence of operators (rulesets, transformations, area descriptions, etc.) that produce a desirable performance. An explanation of the above steps and further details about this methodology are in [Michalski et al., 1996].

To interpret a new image, the system applies the learned transformations to the image. Labels corresponding to the original image areas provide an image interpretation. Learning errors are computed by comparing the target labeling (made by the trainer) with the learned labeling (produced by the system).

Advantages of this methodology include the ease of applying and testing diverse learning methods and approaches in a uniform manner, the potential for implementing very advanced and complex learning processes, the possibility for parallel image learning and interpretation, tand he ease of testing the accuracy and performance of the methods.

Most of the experiments with an earlier version of the methodology have been concerned with learning decision rules characterizing image surface classes from surface samples. The rules were determined using the inductive learning program AQ15c [Wnek et al., 1995] and represented in VL_1 (Variable-Valued Logic System 1; [Michalski, 1973]). These rules served as "logical templates" that were matched against window-size samples of the surface to classify the image, e.g., [Michalski et. al., 1993].

The initial implementation of MIST incorporates the following learning systems:

(i) Inductive rule learning program AQ15c [Michalski et al., 1986; Wnek et al., 1995].

(ii) Data-driven constructive inductive learning program AQ17-DCI, capable of automatically generating new attributes more relevant fto the given task [Bloedorn and Michalski, to appear].

(iii) Hybrid learning system, AQ-NN, that combines symbolic rule learning with neural net learning [Bala et al., 1994].

(iv) Backpropagation neural net learning system, NN [Zurada, 1992].

The initial version of the methodology has been applied to the following vision tasks:

(A) A semantic interpretation of natural scenes [Michalski et al. 1996] and

(B) Detecting objects in a cluttered environment that belong to a specific concept class, e.g., detecting blasting caps in X-ray images [Maloof et al., 1996].

The next section describes briefly the application of MIST to natural scene interpretation.

3 Applying MIST Methodology to Semantic Interpretation of Outdoor Scenes

One of the highly challenging open vision problems is how to interpret natural scenes and recognize natural objects [Fischler and Strat, 1988; Strat and Fischler, 1991]. The MIST methodology seems to offer a novel approach to these problems. In this project, we address the problem of learning to semantically interpret outdoor scenes using several learning methods. In the experiments, we used a collection of images representing selected mountain scenes around Aspen, Colorado (Figure 1).

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. The description space was defined in terms of attributes characterizing image samples in some predefined window.

We experimented with different sets of initial attributes, different images, different sizes of the training areas, and different sources of training and testing image samples (different parts of the same window in the given image for training and testing, different windows in the same image, and different images).

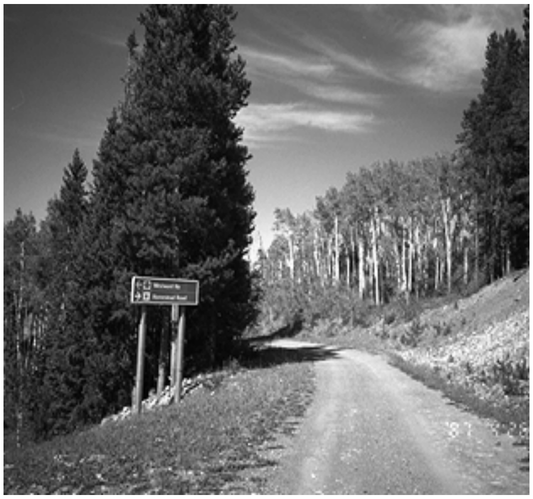

Figure 1: A typical image in the Aspen collection.

In the experiments, we used the four previously mentioned learning systems: AQ15c, AQ17-DCI, AQ-NN, and NN. The results indicated an advantage of the AQ-NN learning system over other systems in terms of both faster learning and recognition times and higher predictive accuracy. Table 1 gives a brief summary of the prediction rate of the descriptions obtained by purely symbolic learning, AQ15c, and multistrategy learning program AQ-NN.

Learning System	**Prediction Accuracy**	
	Single event	Majority voting
Symbolic learning: AQ15c	89%	96%
Multistrategy learning: AQ-NN	91%	99%

It is also worth noting that the constructive induction method, AQ17-DCI, produced several new attributes that improved the performance accuracy and simplified concept descriptions, for example, the differences between the intensities of different colors, the color of maximum intensity, and the sum of intensities. Thus, the program found transformations that in other methods would need to be given to the system explicitly. For more information about these experiments see [Michalski et al., 1996].

4 Recognizing objects in a cluttered environment

This project is conducted jointly by Maloof, Duric, and Michalski and detailed in [Maloof and Michalski, 1995c]. Here we summarize the main ideas. The goal of this research is to develop a method for identifying a given object in a cluttered environment.

The learning methodology used here employs ideas of MIST (and closely parallels Michalski's [1973], Channic's [1988], and Bala's [1993]). The method proceeds in four steps: (1) Region of Interest (ROI) Determination, (2) Event Extraction, (3) Description Learning, and (4) Recognition. The image set used for these experiments are x-ray images of luggage containing blasting caps. Images were acquired by x-raying luggage containing blasting caps and varying amounts of clutter (e.g., shoes, clothes, calculators, bolts, and pens). The luggage was x-rayed much as it would be in an airport scenario: flat in relation to the x-ray source, but rotated in the plane orthogonal to the x-ray source. The image set contains 30 images, but 5 were selected and were of low to moderate complexity in terms of positional variability of the blasting cap, degree of occlusion, and clutter.

The first step involves finding image regions that likely contain blasting caps. Regions were isolated interactively yielding a set of 53 binary objects divided into two classes: blasting caps and non-blasting caps. Once several blasting cap and non-blasting cap objects were extracted from the images, several features were computed. These features included the area of the object, the length of the object's perimeter, the major and minor axes of an ellipse fitted to the object, and compactness.

Thus, each example of a blasting cap or a non-blasting cap consisted of a class label and either real or integer values for each of the computed attributes. Experiments were conducted using three learning methods and a testing methodology of 100 iterations of 2-fold cross validation [Weiss and Kulikowski, 1992]. For these experiments, the learning methods were AQ15c [Wnek et al., 1995], the Quickprop implementation of a backpropagation neural network [Fahlman, 1988], and *k*-nearest neighbor [Weiss and Kulikowski, 1992].

These methods were compared using average predictive accuracy, average learning time, and average recognition time. In the experiments, the AQ15c learning program produced significantly higher recognition accuracy than the neural net (95% vs. 79%) and learned descriptions about two orders of magnitude faster. It has also outperformed the *k*-nearest neighbor method in terms of prediction accuracy (95% vs. 69%). The predictive accuracy results for the learning methods are summarized in Table 1.

Learning Method	Average Predictive Accuracy
AQ15c	95%
Neural Network	79%
k-nn (*k* = 1)	69%

Table 1: Comparative predictive accuracy for three learning methods.

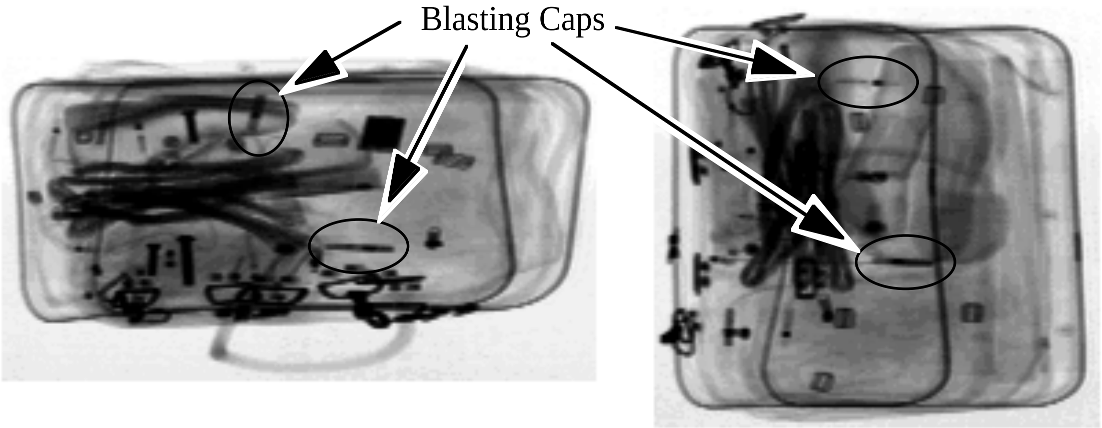

Figure 2: Example x-ray images of luggage containing blasting caps.

We also tried a different approach to this problem, described in [Maloof et al., 1996]. The approach uses attributes that combine intensity and shape information.

Shape information is expressed by compactness. Intensity depends on both on the blasting cap's orientation relative to the x-ray source, and the density of the blasting cap material. As the angle between the long axis of the blasting cap and the imaging plane increases, the compactness of the image increases, while the intensity decreases.

Learning is used to acquire the relationship between intensity and compactness of blasting cap images. Recognition proceeds in a bottom-up and top-down fashion. Low intensity blobs serve as attention-catching devices for local search and model fitting. These models, which are also learned, are used in secondary recognition processes.

Future work will involve using more detailed models, which need images acquired at a higher resolution (e.g., for detecting wires attached to the blasting cap).

5 Learning in Navigation

A robotic agent operating in an unknown and complex environment may employ a search strategy of some kind to perform a navigational task such as reaching a given goal. In the process of performing the task, the agent can attempt to discover characteristics of its environment that enable it to choose a more efficient search strategy for that environment. If the agent is able to do this, we can say that it has "learned to navigate"—i.e., to improve its navigational performance.

The University of Maryland has conducted basic investigations into the problem of how an agent can learn to improve its goal-finding performance in a discrete space, represented by a graph. In particular, several basic search strategies on two different classes of "random" graphs were compared, and it was demonstrated that information collected during the traversal of a graph can be used to classify the graph, thus allowing the agent to choose the search strategy best suited for that graph.

In general, a *navigational task* involves finding a path in a given space that satisfies given constraints (and possibly also minimizes a given cost function). For example, in a goal finding task, the path must terminate at a given point, or at a point that has given properties; in a traversal task, the path must pass through (or close to) every point of the space. (In both of these examples, we can also introduce a cost function and require, e.g., that the path be as short as possible.) If the "layout" of the space is not known in advance, finding an acceptable (or optimal) path involves a search process. In general, there may be many applicable search algorithms, and their relative performance in finding paths may depend on the nature of the space.

A navigating agent may have to perform a navigational task in a given space repeatedly, each time with new constraints. For example, a goal-finding agent may be required to find paths to many different goals. In order to operate efficiently, such an agent should attempt to discover characteristics of the space in which it is navigating, so that it can use relatively efficient algorithms to search for the desired paths. If the agent is able to do this, we can say that it has "learned to navigate", in the sense that it has learned something about the space in which it is navigating and can use this information to improve its navigational efficiency.

In [Cucka et al., 1995], we studied this concept of "learning to navigate" using a goal-finding task in a discrete space, represented by a graph embedded in the plane. We assumed that the agent always knows its position in the plane, as well as the position of the goal; but since it does not know the structure of the graph, reaching the goal requires search. We analyzed several alternative search algorithms and showed that their relative performance differs for different classes of graphs; we also showed how the agent can discover ("learn") which type of graph it is navigating in by collecting information about the graph while it is searching for a goal.

Specifically, we studied two classes of graphs, both constructed by choosing nodes randomly in a planar region. In the first class, pairs of nodes were randomly joined by arcs; the second class were the Delaunay triangulations of the nodes. Evidently, the arcs of a Delaunay graph tend to be shorter, and to vary much less in length, than the arcs of a random graph. The agent can decide which of the two types of graphs it is navigating in by histogramming the lengths of the arcs that it traverses. We found that this decision can be made quite reliably by the time the agent has traversed a few dozen arcs.

The studies considered only a few simple goal seeking strategies. Many variations on these strategies could also be considered. The agent might also use methods other than "pure" search in seeking its goal. For example, it might try to discover "landmarks" that could be used to simplify the process of finding the goal, e.g., by moving from landmark to landmark until a landmark close to the goal is reached, and then searching for the goal. The choice of landmarks for use in navigation is another interesting research issue.

The experiments used only two simple classes of graphs, in which the nodes corresponded to points randomly chosen in a planar region; the graphs were either Delaunay triangulations of these points, or were constructed by joining randomly chosen pairs of the points. Many other classes of discrete spaces could have been used—for example, adjacency graphs of (irregular) tessellations; graphs derived from models for geographical processes; graphs derived from real city maps or road networks; and so on. Evidently, goal finding strategies may differ widely in performance for different types of graphs.

The agent's task was goal finding; as mentioned above, there are many other types of navigational tasks. Evidently, different tasks may require very different strategies; for example, strategies for traversal ("patrolling") would be very different from goal finding strategies. A variation on the goal finding task would be the path shortening problem considered in [Rosenfeld, Rivlin, and Khuller, 1994]: once a goal has been found, try to find a shorter path back to the start node, then a still shorter path back to the goal, and so on.

The agent was able to move only between neighboring nodes of the graph; it could not "jump" or "fly" (or "teleport"). It had an absolute position sense, and was also (possibly) able to sense the directions to the neighbors of a node, or even the positions of these neighbors. Evidently, the difficulty of a navigational task is highly dependent not only on the nature of the space, but also on the capabilities of the agent, as regards both mobility and sensing.

The results obtained also suggest an interesting class of questions about robotic agents that need to estimate properties of their environments. The agent was able to discover the type of graph in which it was navigating by measuring the lengths of the arcs it traversed while searching for a goal, and comparing the histogram of these arc lengths with the theoretical distribution of arc lengths for Delaunay graphs. In this way, the agent "sampled" the arc length distribution; however, it could not take a random sample, but only a "connected" sample obtained as it moved from node to node. Nevertheless, we found that even small samples collected in this way were sufficient to classify the graph (as Delaunay or random) with high confidence. We plan to study the effectiveness of this type of "robotic sampling" as applied to other types of estimation tasks—e.g., estimating the parameters of a distribution of quantities defined on the nodes or arcs of graphs of given classes—in order to better understand the limitations of estimation processes performed by real robots.

6 Intelligent interfaces: Exploiting experience in the RADIUS environment

The RADIUS environment (RCDE) has proven to be a very useful tool for building site models and monitoring changes [Gerson and Wood, 1994; Sargent et al., 1994; Strat and Climenson, 1994]. However, it may require significant repetitive work by the user. A number of groups have been exploring the possibility of improving that process. In our research we are trying to exploit the use of prior experience to help in new modeling and exploitation cases. As an example, an Image Analyst (IA) performs many functions and chooses many parameters when accessing and using RCDE in the Quick-Look mode [Bailey et al., 1994]. An incremental learning process is being studied as a tool for creating an intelligent interface for IAs to simplify and speed up their use of RCDE.

Incremental learning approaches have recently been used successfully for email routing, newsgroup filtering, and calendar scheduling. The objective of using learning for these applications is to "look over the shoulder" of a user, learn patterns of behavior, and automate software functions based on these learned behaviors. The type of learning system necessary to support this class of problems must (1) learn over time, (2) learn quickly, (3) learn with low memory requirements, and (4) learn changing or evolving concepts. We are currently developing methods and systems to support this class of problems. Preliminary experiments have been reported in the domain of computer intrusion detection [Maloof and Michalski 1995a, b].

One potential application of incremental learning for automating software functions for computer vision systems (e.g., RADIUS, Khoros) is the Quick-Look concept [Bailey et al., 1994]. With the Quick-Look concept, incremental learning can be used to prioritize images and image regions for exploitation by examining which images and image regions the IA selects and the order in which they are selected. In this situation, learning could begin with no initial knowledge about a site or image collection, or could work from a user-provided profile.

A second application would be in the automation of repetitive tasks. Again, the learning system would watch the IA perform his or her duties, but would look for repeated sequences of menu actions or tasks. For instance, suppose the IA always performs histogram equalization and a measurement function after zooming. After the system notices and learns this pattern, it may ask the IA if the transition from the zooming function to the histogram equalization and the measurement function can be automated. If the IA indicates positively, then in the future, whenever the IA zooms into an image region, the system would automatically invoke the histogram equalization and the measurement functions. This eliminates the need for the IA to select this function from a menu. At any time, the IA will have the option to either override the learned function or have the system forget the function and relearn another.

7 Learning space configurations for the purpose of solving the homing problem

Another project in the marriage of vision and learning in which we have been engaged over the past year is related to learning of space configurations for the purpose of solving the homing problem. The problem of homing is defined as the process through which a system possessing visual perception can go from one place to another place in some environment on the basis of visual input. The problem, from a technical point of view, is equivalent to constructing a number of visual memories of the environment (knowledge) and then solving the indexing problem (i.e., the problem of localization — to what part of the memory does the current imaged view correspond).

The major difficulty in addressing such a task, a difficulty that is representative of the current state of the art in vision and learning, is that traditional representations of visual space, being representations of distance or depth or its derivatives — surface normals and curvature — are characterized by continuity, while machine learning techniques are by their definition of a discrete nature operating on symbolic entities. Thus, it is necessary to develop representations of visual space that can be stored in data structures consisting of discrete symbolic entities involving a small set of symbols.

A representation of visual space that satisfies these criteria, currently under development, is termed ordinal space representation. In such a representation we do not have knowledge of the values of the depth or range from the image to surfaces in the environment; our knowledge is restricted to ordinal relationships between depth values (greater, smaller or equal). It is possible to represent a retinotopic ordinal map by a set of “symbolic maps” [Aloimonos and Fermueller, 1996 — in these Proceedings]. A collection of such maps represents space. Transformation, or rather transmutation, of this knowledge into a compact format, happens through the extraction of features from these ordinal maps (see above reference) and the relating between the features through a formalism termed “spatial logic”, currently under development.

8 Learning object functionality

For robots, as for humans, recognizing the functions of objects is often a prerequisite to interacting with them. Functionality can be defined as the usability of an object for a particular purpose.

There has been considerable recent research on the problem of recognizing the functionalities of static objects. The goal of this research has been to determine the functional capabilities of an object based on characteristics such as shape, physics and causation. Little attention has been given to the problem of determining the functionality of an object from its motion.

We believe that motion provides a strong indication of function. In particular, velocity, acceleration, and force of impact resulting from motion strongly constrain possible function. As in other approaches to functional recognition, the object (and in our case, its motion) should not be evaluated in isolation, but in context. The context includes the nature of the agent and the frame of reference it uses. A robot can learn object functionality by watching the object in use. As an example, the robot might “see” a knife being used to slice a loaf of bread and learn the function of cutting and the context in which it can be used.

Our research in this area addresses the following problem: How can we use the motion of an object, while it is being used to perform a task, to determine its function? Our method of answering this question is based on motion analysis of the given image sequence. The analysis results in a few motion descriptors. These descriptors are compared with stored descriptors that arise in known motion-to-function mappings to obtain function recognition.

Following [Biederman, 1985; Rivlin et al., 1995] we regard objects as composed of primitive parts. On the most coarse level we consider four types of primitive parts: sticks, strips, plates, and blobs, which differ in the values of their relative dimensions. We can then define the four classes as follows: If all three dimensions are about the same, we have a blob. If two are about the same, and the third is very different, we have two cases: if the two are bigger than the one, we have a plate, and in the reverse case we have a stick. When no two dimensions are about the same we have a strip. For example, a knife blade is a strip, because no two of its dimensions are similar.

These primitives can be combined to create compound objects. In [Rivlin et al., 1995] the different qualitative ways in which these primitives can be combined are described — for example, end to end, end to side, end to edge, etc. In addition to specifying the two attachment surfaces participating in the junction of two primitives, we could also consider the angles at which they join, and classify the joints as perpendicular, oblique, tangential, etc. Another refinement would be to describe qualitatively the position of the joint on each surface; an attachment can be near the middle, near a side, near a corner, or near an end of the surface. We can also specialize the primitives by adding qualitative features such as axis shape (straight or curved), cross-section size (constant or tapered), etc.

Functional recognition is based on compatibility with some action requirement. Some basic "actions" are static in nature (supporting, containing, etc.), but many actions involve using an object while it is moving. To illustrate the ways in which one can interact with a primitive, consider the action of "cutting" with a sharp strip or plate. Here a sharp edge is interacting with a surface. The interaction can be described from a kinematic point of view. The direction of motion of the primitive relative to its axis defines the action — for example, slicing or chopping. We define a primitive motion to be a motion along, or perpendicular to, a main axis of a primitive object. The motion can be either a translation or a rotation.

Given a moving object as seen by an observer, we would like to infer the function being performed by the object. The object is given as a collection of primitives. In this example a knife is described as consisting of two primitives: a handle (a stick) and a blade (a strip). Given this model, the system estimates the pose of the object (as in [Rivlin et al., 1995]) and passes this information to the motion estimation module. The model and the results of the motion estimation enable the system to infer the function that is being performed by the object.

The function being performed by the object depends on the object's motion in the object's coordinate system and on its relation to the object it acts on (the "actee"; in [Kise et al., 1993; Kitahashi et al., 1991] , called the "functant"). This information gives us the relationship between the direction of motion, the main axis of the object, and the surface of the actee, and these relationships determine the intended function. For example, we would expect the motion of a knife that is being used to "stab" to be parallel to the main axis of the knife, whereas if the knife is being used to "chop" we would expect motion perpendicular to the main axis. In both cases, the motion is perpendicular to the surface of the actee. If the knife is being used to "slice", we would expect back-and-forth motion parallel to its main axes and also parallel to the surface of the actee.

In summary, perceiving function from motion provides an understanding of the way an object is being used by an agent. To accomplish this we combine information on the shape of the object, its motion, and its relation to the actee (the object it is acting on). Assuming a decomposition of the object into primitive parts, we analyze a part's motion relative to its principal axes. Primitive motions (translation and rotation relative to the principal axes of the object) are dominating factors in the analysis. We use a frame of reference relative to the actee. Once such a frame is established, it can have major implications for the functionality of an action.

Several sequences of images have been used to demonstrate the approach; the details are given in a separate paper in these Proceedings [Duric et al., 1996]. In the first three sequences, motion was used to discriminate between three cutting actions: stabbing, chopping and slicing. In still other sequences [Duric et al., to appear], we used motion information to differentiate between two different functionalities of the same object: scooping and hitting with a shovel, and hammering and tightening with a wrench. These examples of double usage are typical instances of improvisation; motion provides clear information for a correct interpretation of the action that is taking place.

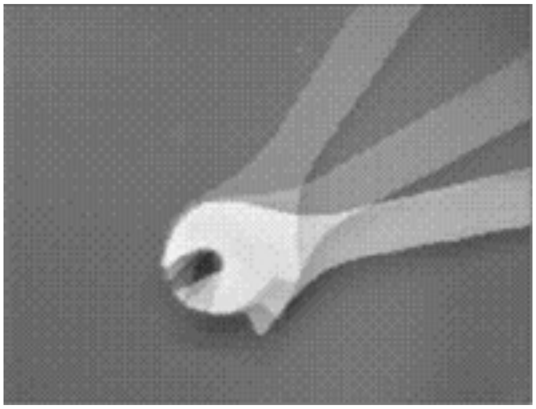

Figure 3: Tightening motion with a wrench.

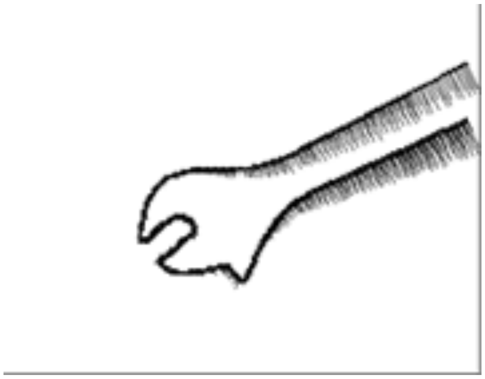

Figure 4: Flow vectors for tightening with a wrench.

An example of applying our method to an image sequence is shown in Figures 3, and 4. Figure 3 shows a motion sequence of a wrench tightening a screw and Figure 4 shows the normal flow field for one frame of the sequence.

Natural extensions of this work include the analysis of more complex objects. Complexity can be expressed in terms of either the shapes of the parts or the way in which the parts are connected. An interesting area is the analysis of articulated objects. The different types of connections between the parts constrain the possible relative motions of the parts. A pair of pliers or a pair of scissors is a simple case, with only a single articulated connection (one degree of freedom in the relative motion of the parts).

Work is in progress in which the methods developed on this project are used to demonstrate how a robot can learn the functionality of an object by observing image sequences in which the object is performing actions which accomplish its function(s).

Acknowledgments

The authors thank Jerzy Bala, Eric Bloedorn, Ken Kaufman and Janusz Wnek for their help in performing experiments with MIST methodology described in this report.

References

Allen, P.K., Boult, T.E., Kender, J.R., and Nayar, S.K., “Image understanding research at Columbia University”, *Proceedings of the 1994 Image Understanding Workshop*, pp. 21–35, 1994.

Aloimonos, Y., and Fermueller, C. “Ordinal vision,” *Proceedings of the 1996 Image Understanding Workshop*, Palm Springs, CA, February 12-15, 1996.

Bailey, J.S., Kelly, M.D., and Sargent, J.D., “Quick-Look: a new way to prioritize imagery for exploitation,” *Proceedings of the 1994 Image Understanding Workshop*, pp. 247–249, 1994.

Bala, J.W. “Learning to recognize visual concepts: development and implementation of a method for texture concept acquisition through inductive learning,” *Reports of the Machine Learning and Inference Laboratory*, MLI-93-3 (Ph.D. Dissertation), George Mason University, Fairfax, VA., 1993.

Bala, J.W., Michalski, R.S., and Pachowicz, P.W., “Progress on vision through learning at George Mason University,” *Proceedings of the 1994 Image Understanding Workshop*, pp. 191–207. 1994.

Bhanu, B., “Image understanding research at UC Riverside”, *Proceedings of the 1994 Image Understanding Workshop*, pp. 3–8, 1994.

Biederman, I., “Human image understanding: recent research and a theory”, *Computer Vision,* *Graphics and Image Processing*, 32, pp. 29–73, 1985.

Bloedorn, E., and Michalski, R.S. “The AQ17-DCI system for data-driven constructive induction. Submitted to ISMIS ‘96, June 10–13, 1996, Zakopane, Poland.

Bloedorn, E., Wnek, J., Michalski, R. S., and Kaufman, K., “AQ17—A multistrategy learning system: the method and user’s guide,” *Reports of the Machine Learning and Inference Laboratory*, MLI 93–12, George Mason University, Fairfax, VA. 1993.

Bolles, R.C., and Fischler, M.A., “Image understanding research at SRI International”, *Proceedings of the 1994 Image Understanding Workshop*, pp. 53–68, 1994.

Channic, T., “TEXPERT: An application of machine learning to texture recognition,” *Machine Learning and Inference Reports*, No. 88-27, George Mason Universrity (representing a M.S. thesis prepared at the University of Illinois, Urbana), 1988.

Chellappa, R., Zheng, Q., Davis L.S., Liu C.L., Zhang, X., Rodriguez C., and Rosenfeld, A., “Site model based image registration and change detection,” Contract DACA 76-92-C-0024, First Annual Report on Radius Project for the period 30 Sept. 92 – 29 Sept. 93, 1993.

Cucka, P., Netanyahu, N.S., and Rosenfeld, A. “Learning in navigation: goal finding in graphs”, *Center for Automation Research Technical Report* CAR-TR-759, University of Maryland, College Park, MD, 1995.

Duric, Z., Fayman, J., and Rivlin, E., “Function from motion,” *IEEE Transactions on PAMI*, to appear.

Duric, Z., Rivlin, E., and Rosenfeld, A., “Learning an object’s function by observing the object in action”, *Proceedings of the 1996 Image Understanding Workshop*, Palm Springs, CA, Febrau;ry 12-15, 1996.

Fahlman, S.E., *An empirical study of learning speed in back-propagation networks*. Technical Report CMU-CS-88-182. Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA. 1988.

Fischler, M.A. and Strat, T.M., “Recognizing trees, bushes, rocks and rivers,” *Proceedings of the AAAI Spring Symposium Series: Physical and Biological Approaches to Computational Vision*, Stanford University, pp. 62–64, March 1988.

Gerson, D.J. and Wood, Jr., S.E., “RADIUS phase II—the RADIUS testbed system,” *Proceedings of the 1994 Image Understanding Workshop*, pp. 231–237, 1994.

Grimson.W.E.L., Horn, B.K.P., and Poggio, T., “Progress in image understanding at MIT,” *Proceedings of the 1994 Image Understanding Workshop*, pp. 37–42, 1994.

Hanson, A.R., Riseman, E.M, and Weiss, R., “Progress in computer vision at the University of Massachusetts,” *Proceedings of the 1994 Image Understanding Workshop*, pp. 43–51, 1994.

Houzelle, S. Strat, T.M., Fua, P., and Fischler, M., “Using contextual information to set control parameters of a vision process,” *Proc. of the Twelfth International Conference on Pattern Recognition*, vol.1 pp. 830-832, Jerusalem, September 1994.

Kise, K., Hattori, H., Kitahashi, T., and Fukunaga, K., “Representing and recognizing simple hand-tools based on their functions,” In *Proceedings of the Asian Conference on Computer Vision*, pp. 656–659, 1993.

Kitahashi, T., Abe, N., Dan, S., Kanada, K., and Ogawa, H., “A function-based model of an object for image understanding,” In *Advances in Information Modelling and Knowledge Bases*, IOS Press, Jaakkola, H. and Ohusuga, S. editors, pp. 91–97, 1991.

Maloof, M.A., and Michalski, R.S., “A partial-memory incremental learning methodology and its application to intrusion detection,” *Reports of the Machine Learning and Inference Laboratory*, MLI 95–2, George Mason University, Fairfax, VA, 1995a.

Maloof, M.A., and Michalski, R.S., “A partial-memory incremental learning methodology and its application to intrusion detection,” *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence*, pp. 392–397. 1995b.

Maloof, M.A., and Michalski, R.S., “Learning symbolic descriptions of 2d shapes for object recognition in x-ray images,” *Proceedings of* *the 8th International Symposium on Artificial Intellignce*, 1995c.

Maloof, M.A., Duric, Z., Michalski R.S., and Rosenfeld, A., "Recognizing blasting caps in x-ray images," *Proceedings of the '96 Image Understandning Workshop*, Palm Springs, CA, February 12-15, 1996.

Michalski, R.S., "A Variable-valued logic system as applied to picture description and recognition," in *Graphic Languages*, F. Nake, F. and A. Rosenfeld (eds.), North Holland, 1972.

Michalski, R.S., "AQVAL/1 — computer implementation of a variable-valued logic system VL_1 and examples of its application to pattern recognition," *Proceedings of the First International Joint Conference on Pattern Recognition*, pp. 3–17. 1973.

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

Michalski, R.S., Rosenfeld, A., and Alomoinos, Y., "Machine vision and learning: research issues and directions, *A report on the NSF/ARPA Workshop on Learning and Vision*, Harpers Ferry, WV, October 15-17, 1992. *Reports of the Machine Learning and Inference Laboratory*, MLI 94-6, George Mason University, Fairfax, VA, 1994.

Michalski, R.S., Zhang, Q., Maloof, M.A., and Bloedorn, E., "The multi-level image sampling and transformation methodology and its application to natural scene interpretation," *Proceedings of the '96 Image Understandning Workshop*, Palm Springs, CA, February 12-15, 1996.

Rivlin, E., Dickinson, S.J., and Rosenfeld, A., "Recognition by functional parts," *Computer Vision and Image Understanding*, 62, pp. 164–176, 1995.

Rosenfeld, A., Rivlin, E., and Khuller, S. "Learning to navigate on a graph," *Proceedings of the 1994 Image Understanding Workshop*, pp. 789–795, 1994.

Sargent, J.D., Kelly, M.D., and Bailey, J.S., "RADIUS concept definition experiments," *Proceedings of the 1994 Image Understanding Workshop*, pp. 239–245, 1994.

Shafer, S.A., Kanade, T., and Ikeuchi, K., "Image understanding at CMU," *Proceedings of the 1994 Image Understanding Workshop*, pp. 81–92, 1994.

Strat, T.M. and Climenson, D.W., "RADIUS: site model content," *Proceedings of the 1994 Image Understanding Workshop*, pp. 277–286, 1994.

Strat, T.M. and Fischler, M.A., "Natural object recognition: a theoretical framework and its implementation," *Proceedings of IJCAI 91*, August, 1991.

Weiss, S.M. and Kulikowski, C.A., *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning and expert systems*, Morgan Kaufmann, San Mateo, CA, 1992.

Wnek, J., Kaufman, K., Bloedorn, E., and Michalski, R.S., "Inductive learning system AQ15c: the method and user's guide," *Reports of the Machine Learning and Inference Laboratory*, MLI 95–4, George Mason University, Fairfax, VA. 1995.

Zurada, J.M., *Introduction to artificial neural systems*, West Publishing, St. Paul, MN, 1992.