Learning Integrated Image Segmentation and Object Recognition

Bir Bhanu
College of Engineering
University of California
Riverside, CA 92521-0425
email: bhanu@vislab.ucr.edu

Jing Peng
College of Engineering
University of California
Riverside, CA 92521-0425
email: jp@vislab.ucr.edu

Abstract

This paper presents a general approach to image segmentation and object recognition that learns a mapping from images with varying properties to segmentation algorithm parameters. The mapping is built using a reinforcement learning algorithm that is based on a team of generalized stochastic learning automata and operates separately in a global or local manner on an image. The edge-border coincidence is first used as an immediate reinforcement to reduce computational expenses associated with model matching during the early stage of the learning process. Since this measure however can not reliably predict the outcome of object recognition, it is used in conjunction with model matching that provides optimal segmentation evaluation in a closed-loop object recognition system. Results are presented for both indoor and outdoor color images where the performance improvement over time is shown for both image segmentation and object recognition.

1 Introduction

A model based object recognition system has three key components: image segmentation, feature extraction, and model matching. The goal of image segmentation is to extract meaningful objects from an input image. Image segmentation is an important and one of the most difficult low-level image processing and computer vision tasks [4, 9]. All subsequent image interpretation tasks including feature extraction and model matching rely heavily on the quality of the image segmentation process.

The inability to adapt the image segmentation process to real-world changes is one of the fundamental weaknesses of typical model-based object recognition systems. Despite the large number of image segmentation algorithms available, no general methods have been found to process the wide diversity of images encountered in real world applications. Typical object recognition systems are *open-loop*. Segmentation and feature extraction modules use default algorithm parameters, and generally work as pre-processing steps to the model matching component. These parameters are not reliable, since when the conditions for which they are designed are changed slightly, these algorithms generally fail without any graceful degradation in performance. To achieve reliable performance in real-world applications, a need exists to apply learning techniques that can efficiently search the segmentation parameter space and find parameter values that produce optimal results for the given recognition task. In this paper, our goal is to develop a *general* approach for learning integrated model-based object recognition system, which has the capability of inducing a mapping from input images to segmentation parameters. We emphasize at the outset that for simplicity it is assumed that only one instance of the model is present in the image. Multiple instances of the model can be recognized by slight modification of the algorithm.

1.1 Overview of the approach

We develop a general approach to learning integrated image segmentation and object recognition. The basic *assumption* is that we know the models of the objects that are to be recognized, but we do not know the number of objects and their locations in the image. The system consists of image segmentation, feature extraction, model matching, and reinforcement learning modules. The image segmentation component extracts meaningful objects from input images, feature extraction step performs polygonal approximation of connected components, and the model matching step tells us which regions in the segmented image contain the recognized object. The model matching module indirectly evaluates the performance of the image segmentation and feature extraction processes by generating a real valued matching confidence indicating the degree of success. This real valued matching confidence is then used as feedback to drive learning for image segmentation parameters in a general learning framework. The goal is therefore to maximize the matching confidence by finding a set of segmentation parameters for the given recognition task.

The particular framework adopted in this paper is reinforcement learning, which closes the loop between model matching and image segmentation. There are good reasons for using reinforcement learning in our image segmentation and computer vision system. *First*, reinforcement learning requires knowing only the goodness of the system performance rather than the details of algorithms that produce the results. It is natural to use matching confidence as reinforcement. *Second*, convergence is guaranteed for several reinforcement learning algorithms. *Third*, reinforcement learning performs efficient hill-climbing in a statistical sense without excessive demand for computational resources. Furthermore, it can generalize over unseen images. *Fourth*, reinforcement learning can systematically assign credit to different levels in a multi-level image processing and computer vision system.

The learning integrated image segmentation and object recognition system is designed to be *fundamental* in nature and is not dependent on any specific image segmentation algorithm or type of input images. To represent segmentation parameters suitably in a reinforcement learning framework, the system only needs to know the segmentation parameters and their ranges. In our approach, a binary encoding scheme is used to represent the segmentation parameters. While the same task could be learned in the original parameter space, for many types of problems, including image segmentation, the binary representation is expected to learn much faster [15, 14]. In this sense, our system is independent of a particular segmentation algorithm used.

1.2 Related work and our contributions

There is no published work in image processing and computer vision on learning integrated image segmentation and object recognition using multiple feedbacks. The adaptive parameter control of segmentation algorithm and the adaptive selection and combination of different algorithms in a learning integrated system are *unsolved* problems in the field of image processing and computer vision [1]. Most threshold selection techniques in image processing and computer vision do not involve any learning to improve future performance with experience.

In a recent article Burges et al. [8] describe a method for coupling recognition and segmentation by the principle of heuristic over segmentation. The basic idea is that a segmentation algorithm generates a graph that summarizes a large number of segmentation hypotheses that are scored by a recognition algorithm. A globally optimal decision is then made that combines uncertainties in segmentation and recognition. Each time a new input comes in a over segmented hypothesis graph must be generated and traversed in order to classify the input. In contrast, the system presented in this paper uses a learned mapping to compute segmentation parameters for a given input to achieve optimal model matching. In addition, the learning of mapping in our system is driven completely by the matching confidence, whereas their graph generation is largely based on heuristics. In another work [11], graph generation is actually learned by minimizing global errors that take into account both segmentation hypotheses and recognition scores. In [2], a method is described for fitting segmentation parameters to maximize the likelihood of a model of an object. In comparison, our system attempts to maximize a classification (conditional) probability.

Bhanu and Lee [4, 5] present an image segmentation system which incorporates a genetic algorithm to adapt the segmentation process. In their approach, multiple segmentation quality measures are used as feedback. Some of these measures require ground-truth information that may not be always available. Recently, Peng and Bhanu [15, 14] develop an approach in which learning is used to close the loop between segmentation and recognition. Their approach is based on global image segmentation which is not the most appropriate way to extract objects in an image. We need the capability of performing segmentation based on local image properties (local segmentation). Another disadvantage of their method is its time complexity that makes it problematic for practical applications of image processing and computer vision.

The original contributions of the learning integrated image segmentation and object recognition system presented in this paper are: (1) Model matching confidence is used as feedback to influence the image segmentation process, thereby providing our object recognition system with adaptability in real-world applications. (2) A team of generalized stochastic learning automata is used to represent both global and local image segmentation parameters, making faster learning possible. (3) Edge-border coincidence, when combined with matching confidence, reduces overall computational costs of the learning process. (4) Explicit bias is introduced in a reinforcement learning system in order to speed up the learning process for adaptive image segmentation.

2 Technical approach

The ultimate goal of our system is to maximize the model matching confidence for a given recognition task by finding a set of segmentation algorithm parameters that are represented by a network of independent Bernoulli units. To do so, the following steps are taken.

Initial Approximation. The edge-border coincidence is first used as an immediate reinforcement to locate an initial good point in weight space from which to begin search that will give rise to high recognition performance. Once the edge-border coincidence has exceeded a given threshold, learning is then driven by the matching confidence, which requires more expensive computation of feature extraction and model matching. Although the edge-border coincidence can not reliably predict the outcome of model matching, lower edge-border coincidence values always result in poor model matching. Likewise, higher edge-border coincidence values suggest with high probability that the current set of segmentation parameters is in a close neighborhood of the optimal one.

Global Segmentation Learning. Global segmentation is carried out for the entire image. We assume that we have a prior knowledge of the size of objects of interest in the images. For those connected components that pass through the size filter based on the expected size of objects of interest in the image, we perform feature extraction and model matching. The highest matching confidence is taken as reinforcement to the learning system, which computes a global mapping from input images to segmentation parameters. Since significant differences in characteristics exist between an image and its subimages, however, this mapping may not always be possible to achieve optimal segmentation and recognition performance for individual objects in the input image. This calls for local segmentation learning.

Local Segmentation Learning. The goal of local segmentation learning is to optimize and localize computation to meet each individual requirement. The local learning process starts with the set of weights obtained from global learning. Similar to global learning, the matching confidence is used as reinforcement to update the local weights. Local learning computes a mapping from subimages to segmentation parameters, which is then applied to selected subimages to further optimize recognition performance.

Learning Bias. In order to further speed-up the learning process we introduce bias favoring such a distribution that encourages local exploitation (note that the reinforcement learning is unbiased *initially* when the edge-border coincidence is used as reinforcement). We achieve better computational efficiency of the learning system and improved recognition rates compared to the system with no bias.

2.1 *Phoenix* image segmentation algorithm

Since we are working with color imagery in our experiments, we have selected the *Phoenix* segmentation algorithm [10, 13]. It works by recursively splitting regions using histogram for color features. *Phoenix* contains seventeen different control parameters, fourteen of which are adjustable. The four most critical ones that affect the overall results of the segmentation process are selected for adaptation: *Hsmooth, Maxmin, Splitmin*, and *Height*. *Hsmooth* is the width of the histogram smoothing window. *Maxmin* is the lowest acceptable peak-to-valley height ratio. *Splitmin* represents the minimum area for a region to be automatically considered for splitting. *Height* is the minimum acceptable peak height as a percentage of the second highest peak. Each parameter has 32 possible values. The resulting search space is 2^{20} sample points. Each of the *Phoenix* parameters is represented using 5 bit binary code, with each bit represented by one Bernoulli unit (see section 2.3). To represent 4 parameters, we need a total of 20 Bernoulli units.

2.2 Segmentation evaluation

Given that feature extraction and model matching are computationally expensive processes, it is imperative that initial approximation be made such that overall computation can be reduced. In order to achieve this objective, we introduce a second feedback signal - the *edge-border coincidence* (EBC)[4, 12] that evaluates the segmentation quality. EBC measures the overlap of the region borders in the segmented image relative to the edges found using an edge detector, and does not depend on any ground-truth information. Let E be the set of pixels extracted by the edge operator and S be the set of pixels found on the region boundaries obtained from the segmentation algorithm:

$$EBC = n(E \cap S)/n(E), \tag{1}$$

where $n(\cdot)$ computes the number of elements of its argument.

In this paper, we use the *Sobel* edge detector [17] to compute the necessary edge information. It is possible that EBC is high while matching confidence level is low, or EBC is low while matching confidence is high. Figure 1 shows that EBC does not correlate well with matching confidence. The model matching confidence is arguably the only measure that can conclusively evaluate the performance of the segmentation process. It must be said that although the edge-border coincidence does not correctly predict the matching confidence, for our purpose it is sufficient to drive initial estimation. If the edge-border coincidence is under a prespecified threshold, which indicates a low possibility to get a good recognition result, the system repeats the initial estimation process using the edge-border coincidence as the sole reinforcement signal. Once the threshold has been reached, the segmentation performance will be determined completely by the model matching.

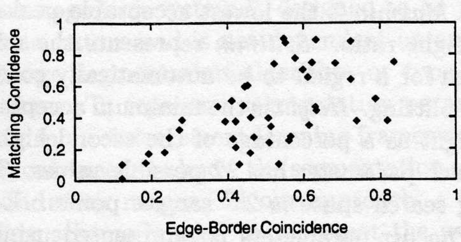

Figure 1: Edge-border coincidence vs. matching confidence.

2.3 Reinforcement learning for image segmentation

Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. This type of learning has a wide variety of applications, ranging from modeling behavior learning in experimental psychology to building active vision systems. The basic idea is that if an action is followed by a satisfactory state of affairs or an improvement in the state of affairs, then the tendency to produce that action is reinforced. Reinforcement learning is similar to supervised learning in that it receives a feedback to adjust itself. However, the feedback is *evaluative* in the case of reinforcement learning. In general, reinforcement learning is more widely applicable than supervised learning and it provides a competitive approach to building autonomous learning systems that must operate in the real world.

The particular class of reinforcement learning algorithms employed in our system is the connectionist *REINFORCE* algorithm [18], where units in such a network are *Bernoulli quasi-linear units*. The output, y_i, of a unit i is either 1 or 0 determined stochastically using the Bernoulli distribution

$$y_i = \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases} \tag{2}$$

where p_i is the probability mass function computed according to $p_i = f(s_i) = 1/(1 + e^{-s_i})$, with $s_i = \sum_j w_{ij}x_j$, w_{ij} is the weight of the jth input for unit i, and x_j is the jth input value for the unit. In the reinforcement learning paradigm, the learning component uses the reinforcement $r(t)$ to drive the weight changes. The specific algorithm we used has the following form: for each unit, at the tth time step, after generating output $y(t)$ and receiving reinforcement signal $r(t)$, increment each weight w_{ij} by

$$\Delta w_{ij}(t) = \alpha[r(t) - \bar{r}(t-1)][y_i(t) - \bar{y}_i(t-1)]x_j - \delta w_{ij}(t) \tag{3}$$

where α is the learning rate, δ is the weight decay rate, x_j is the input to each Bernoulli unit, and y_i is the output of the ith Bernoulli unit. $\bar{r}(t)$ is the exponentially weighted average of prior reinforcement values $\bar{r}(t) = \gamma\bar{r}(t-1) + (1-\gamma)r(t)$, where $\bar{r}(0) = 0$ and γ is the trace parameter. Similarly, $\bar{y}_i(t)$ is an average of past values of y_i computed by the same exponential weighted scheme used for $\bar{r}(t)$, $\bar{y}_i(t) = \gamma\bar{y}_i(t-1) + (1-\gamma)y_i(t)$. The algorithm has the convergence property [18] such that it statistically climbs the gradient of expected reinforcement in weight space. The weight decay (see the second term in equation (3) is used as a simple method to force the sustained exploration of the parameter space. This type of learning rule has shown greater computational efficiency [19].

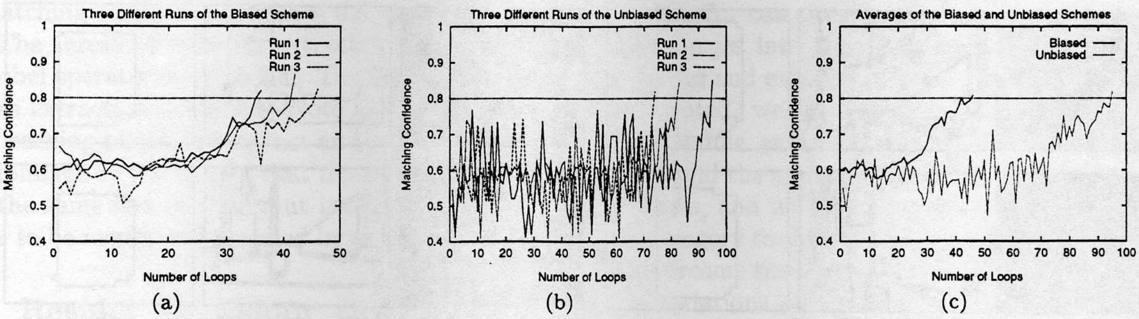

Figure 2: Matching confidence history of three runs of the biased and unbiased RL algorithms on the image shown in Figure 3. (a) Biased; (b) Unbiased (c) Average.

2.4 Biased reinforcement learning for image segmentation

The RL algorithm described in section 2.3 is "unbiased" in that the output of a bit is governed solely by the Bernoulli probability law. The advantage is that rapid changes in output values allow giant leaps in the search space, which in turn enables the learning system to quickly discover suspected high pay-off regions. However, once the system has arrived at the vicinity of a local optimum, as will be the case after the initial estimation, changes in the most significant bit will drastically alter the parameter value, often jumping out of the neighborhood of the local optimum. Ideally, once the learning system discovers that it is within a possible high pay-off region, it should attempt to capture the regularities of the region. This then biases future search toward points within it. The challenge, of course, is to have a learning algorithm that allows the parameters controlling the search distribution to be adjusted so that this distribution comes to capture this knowledge. The algorithm described here shows some promise in this regard. In order to force parameters to change slowly, after the initialization phase, we apply a *biased* RL algorithm in which the two most significant bits of a parameter are forced to change in a "lazy" fashion as:

$$y_i = \begin{cases} 1 & \text{if } p > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Other bits use the the same rule (2) as described in the unbiased RL algorithm. Figure 2 shows the experimental results of the two schemes on the first image shown in Figure 3. In this experiment, we apply the initialization followed by global learning without switching between global and local learning. The results show that the biased RL algorithm demonstrates a speed up by a factor of 2 to 3.

2.5 Feature extraction and model matching

Feature extraction consists of finding polygon approximation tokens for each connected component obtained after image segmentation. To speed up the learning process, we assume that we have the prior knowledge of the *approximate size* (area) of the object, and only those connected components whose area (number of pixels) are comparable with the area of the model object are approximated by a polygon.

The polygon approximation is implemented by calling the polygon approximation routine in *Khoros* [17]. The resulting polygon approximation is a vector image to store the result of the linear approximation. The image contains two points for each estimated line. Model matching employs a cluster-structure matching algorithm [7]. It is based on forming the clusters of translational and rotational transformations between the object and the model. The algorithm takes as input two sets of tokens, one of which represents the stored model and the other represents the input region to be recognized. It then performs topological matching between the two token sets and computes a real number that indicates the confidence level of the matching process. For further details, see [7].

2.6 Algorithm description

There are three distinct learning phases. The initial estimation does not involve the polygonal approximation and model matching. The segmentation parameters are computed based on (2), and weights are updated according to (3) using EBC

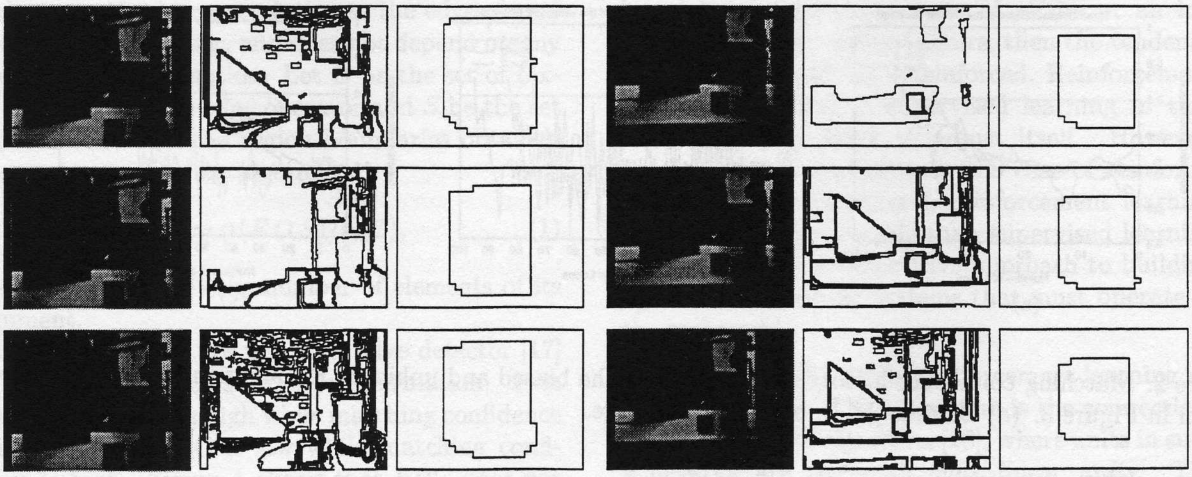

Figure 3: Column 1: input color images 1, 2, 3; column 4: input color images 6, 7, 8; column 2, 3, 5, 6: corresponding segmented image and recognized object.

(1) as reinforcement. The initial estimate terminates when EBC exceeds a given threshold at which point global segmentation learning begins. In both global and local segmentation learning, the segmentation parameters are computed using the lazy strategy (4), and the matching confidence is used as reinforcement that requires polygon approximation and model matching. The training images for local segmentation learning consist of subimages extracted from target areas in the training images for global segmentation learning. The global learning procedure calls the local learning routine when the matching confidence has reached a prespecified level. Training terminates when the average matching confidence is above a given threshold.

3 Experimental evaluation

The system is verified through twelve indoor and twelve outdoor color images. The indoor images are acquired at different viewing distances with varying lighting conditions. The outdoor images are collected every 15 minutes over a three-hour period using a JVC GXF700U color video camera. These images simulate a photo interpretation/surveillance scenario in which the camera positions are fixed and the image undergoes significant changes over time due to changing environmental conditions (time of the day, position of the sun in the sky, and cloud cover). Varying light level is the most prominent change throughout the outdoor images. Although the environmental conditions also created varying object highlights, moving shadows and many subtle contrast changes between the objects in the image. The size of indoor images is 120 by 160 pixels, and the size of outdoor images is 120 by 120 pixels. Each image is decomposed into 4 images for *Phoenix* segmentation – red, green, blue components, and the Y component of YIQ model of color images. For the indoor images, the desired object is the cup in the image, and in the outdoor images, the target object is the traffic sign. The expected size of the cup and the traffic sign are 200 to 450 pixels and 36 to 100 pixels, respectively.

Based on the size of the object to be recognized in the image, we divide the Y component image into 48 subimages for the indoor images, and 36 subimages for the outdoor images. Each subimage's size is 20 by 20 pixels. The images are first histogram equalized using the Khoros routines [17]. The standard deviations of these subimages serve as inputs to each Bernoulli unit, i.e., each Bernoulli unit has a total of 48 inputs (and therefore, 48 weights) for the indoor image, and has a total of 36 inputs (36 weights) for the outdoor image. To learn the four selected *Phoenix* segmentation parameters, we need 20 Bernoulli units. So there is a total of 960 weights for the indoor images, and 720 weights for the outdoor images. It should be noted that, because of independence of these units, the effective number of free parameters is forty-nine for the indoor images and thirty-seven for the outdoor images, respectively.

For the team of 20 Bernoulli units, the parameters α, γ, and δ are determined empirically, and they are kept constant for all images. In our experiments,

$\alpha = 0.02$, $\gamma = 0.9$, and $\delta = 0.01$. The threshold for matching confidence *Switch* = 0.6, and *Accept* = 0.8. The threshold used for extracting edges using the Sobel operator is set to 200. The parameters for feature extraction are fixed. Note that during the local learning phase, we extract and enlarge the local subimage by a factor so that the enlarged image is of the same size as the input image. The stored model to be recognized is scaled by the same factor.

3.1 Results on indoor and outdoor images

Figure 3 shows the experimental results on the first six of the twelve indoor color images while Figure 4 shows the results on the first six of the twelve outdoor color images. For each indoor image, the globally segmented image using the set of learned parameters and the extracted object that has been successful recognized are presented. For each set of images, the 12 images are taken sequentially. Except for the first image, the learning process for each image starts from the global segmentation parameters learned from all the previous images. For the first input image, the learning system is initialized using the unbiased RL algorithm. Usually, it takes less than 45 iterations to find a set of segmentation algorithm parameters that produce high edge-border coincidence values. The final matching confidence values obtained are 0.87, 0.93, 0.86, 0.91, 0.92 and 0.97 for the indoor images, and 0.82, 0.91, 0.86, 0.90, 0.90 and 0.91 for the outdoor images. The results demonstrate clearly that our system works well on images exhibiting condition changes.

4 Conclusions

We have presented a general approach to learning integrated image segmentation and object recognition. The approach systematically combines a domain independent simple measure for segmentation evaluation (edge-border coincidence) and domain dependent model matching confidence in a novel reinforcement learning framework to efficiently learn segmentation parameters and perform object recognition simultaneously. Experimental results demonstrate that the simple approach is promising in real-world applications.

In order to accommodate the wide variety of images encountered in real-world applications, we can develop an autonomous gain control system that will allow switching between different classes of images taken under significantly different weather conditions (sunny, cloudy, snowy, rainy) and learn separate segmentation parameters within each class of images. We can use image context to divide the input images into several classes based on image properties and external conditions. When an image is presented, we use an image property measurement module and the available external information to find the stored information for this category of images, and use the parameters associated with that category to perform image segmentation. This will overcome the problem of adapting to extremely large variations among images. Further, our learning approach can be readily extended to systems using different segmentation algorithms.

Acknowledgements

This work was supported by DARPA/AFOSR grants F49620-97-1-0184 and F49620-95-1-0424. The contents of the information do not necessarily reflect the position or the policy of the U.S. Government. Authors would like to thank Xin Bao for generating experimental results.

References

[1] W. Au and B. Roberts. Overview of a self-adaptive ATR system via context-based configuration and control. In *ARPA Image Understanding Workshop*, pp. 585-588, Palm Springs, CA, Feb. 1996.

[2] Y. Bengio and Y. Le Cun. Word normalization for on-line handwritten word recognition. In *Proc. of Int. Conf. on Pattern Recognition*, pp. 409-413, Jerusalem, 1994.

[3] B. Bhanu, S. Lee and S. Das. Adaptive image segmentation using genetic and hybrid search methods. *IEEE Trans. on Aerospace and Electronic Systems*, 31(4):1268-1291, Oct. 1995.

[4] B. Bhanu and S. Lee. *Genetic Learning for Adaptive Image Segmentation.* Boston MA: Kluwer Academic Publishers, 1994.

[5] B. Bhanu, S. Lee, and J. Ming, "Adaptive image segmentation using a genetic algorithm," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 25, No.12, pp. 1543-1567, Dec. 1995.

[6] B. Bhanu and T. L. Jones. Image understanding research for automatic target recognition. *IEEE Aerospace and Electronics Systems* , 8(10):15-23, Oct. 1993.

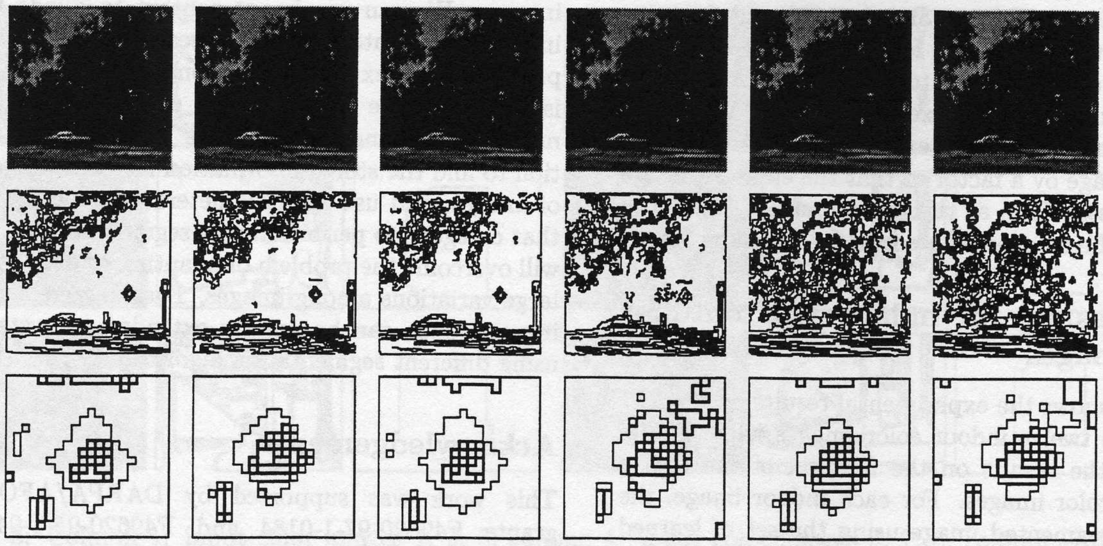

Figure 4: Row 1: input color images 1, 2, 3, 4, 5, 6; rows 2 and 3: corresponding segmented image and recognized object.

[7] B. Bhanu and J. Ming. Recognition of occluded objects: A cluster- structure algorithm. *Pattern Recognition*, 20(2):199-211, 1987.

[8] C. Burges et al. Shortest path segmentation: A method for training a neural network to recognize character strings. In *Proc. of Int. Joint Conf. on Neural Network*, Vol. 3, pp. 165-172, Baltimore, 1992.

[9] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29:100-132, 1985.

[10] K. Laws. The *Phoenix* image segmentation system: Description and evaluation. SRI International Technical Report TR289, Dec. 1982.

[11] Y. LeCun, L. Bottou and Y. Bengio. Reading checks with graph transformer networks. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing* (ICASSP 97), Vol. 1, pp. 151-154, 1997.

[12] D. L. Milgram. Region extraction using convergent series. *Computer Graphics and Image Processing*, 11:1-12, 1979.

[13] R. Ohlander, K. Price and D. R. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8:313-333, 1978.

[14] J. Peng and B. Bhanu. Closed-Loop Object Recognition Using Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20,. No. 2, pp. 139-154, 1998.

[15] J. Peng and B. Bhanu. Closed-loop object recognition using reinforcement learning. In *ARPA Image Understanding Workshop*, pp. 727-731, Monterey, CA, Nov. 1994.

[16] J. Peng and B. Bhanu. Delayed reinforcement learning for closed-loop object recognition. In *Proc. of the 13th Inter. Conf. on Pattern Recognition*, pp. 310-314, Vienna, Austria, Aug. 1996.

[17] J. Rasure and D. Argiro. Khoros User's Manual, University of New Mexico, 1991

[18] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229-256, 1992.

[19] R. J. Williams and J. Peng, "Function optimization using connectionist reinforcement learning algorithms," *Connection Science 3(3)*, 1991.