Reinforcement Learning Integrated Image Segmentation and Object Recognition

Bir Bhanu, Xin Bao and Jing Peng
College of Engineering, University of California, Riverside, CA92521
Email: {bhanu, bao, jp}@constitution.ucr.edu
URL: http://constitution.ucr.edu/

Abstract

This paper presents a general approach to image segmentation and object recognition that can adapt the image segmentation algorithm parameters to the changing environmental conditions. Segmentation parameters are learned using a reinforcement learning (RL) algorithm that is based on a team of learning automata and operates separately in a global or local manner on an image. The edge-border coincidence is used as a short term reinforcement to reduce the computational expense due to model matching during the early stage of object recognition. However, since this measure is not reliable for object recognition, it is used later in conjunction with model matching in a closed-loop object recognition system that uses the results of model matching as a reinforcement signal in a "biased" learning system. The control switches between learning integrated global and local segmentation based on the quality of segmentation and model matching. Results are presented for both indoor and outdoor color images where the performance improvement is shown for both image segmentation and object recognition with experience.

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 computer vision tasks [6]. All subsequent tasks including feature extraction, 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 [10], no general methods have been found to process the wide diversity of images encountered in real world applications. Usually, an object recognition system is open-loop. Segmentation and feature extraction modules use default algorithm parameters, and generally work as pre-processing steps to the model matching component. The fixed sets of algorithm parameters used in various image segmentation and feature extraction algorithms generally degrade the system performance and lack adaptability in real-world applications. These default sets of algorithm parameters are usually obtained by the system designer by following a trial and error method. Parameters obtained in this way are not robust, since when the conditions for which they are designed are changed slightly, these algorithms generally fail without any graceful degradation in performance.

The usefulness of a set of algorithm parameters in a system can only be determined by the system's output, i.e., recognition performance. To recognize different objects or instances of the same object in an image, we may need different sets of parameters locally due to the changes in local image properties, such as brightness, contrast, etc. Also the changing environmental conditions (such as the time of the day, weather conditions, etc.), affect the appearance of an image which requires the capability to adapt the representation parameters for multi-scenario object recognition. To achieve robust performance in real-world applications, a need exists to apply learning techniques which can efficiently search image segmentation and feature extraction algorithm parameter spaces and find parameter values which yield optimal results for the given recognition task. In this paper, our goal is to develop a general approach to a learning integrated model-based object recognition system, which has the ability to continuously adapt to normal environmental variations.

In the remainder of the section 1, we present an overview of the approach, related work and the contributions of the paper. Section 2 gives the details of the approach and discusses algorithms used in this research. Section 3 provides the experimental results.
for segmentation and recognition on both indoor and outdoor color images. Finally, section 4 presents the conclusions and the future work.

1.1 Overview of the approach

In this paper, we present a general approach to reinforcement learning integrated image segmentation and object recognition. A reinforcement learning system is integrated into the model-based object recognition system to close the loop between model matching and image segmentation. 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 goal of the system is to maximize the matching confidence by finding a set of image segmentation algorithm parameters for the given recognition task (We have not discussed the problem of feature extraction parameters in this paper. It is described in a separate paper by Peng and Bhanu [1]). Thus, we address the problem of adaptive segmentation as finding a set of parameters for the given model and given input image. It reflects the fact that there may not exist a single set of "optimal" parameters which can be used for recognizing different objects in a given image. Figure 1 provides an overview of the system. Basically, 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 to drive learning for image segmentation parameters in a reinforcement learning framework.

Given the computational expense for performing model matching, our approach uses edge-border coincidence [5] as a segmentation evaluation measure to find an initial point from which to begin the search through weight space. However, since this measure is not reliable as matching confidence, we use it in conjunction with model matching in a closed-loop system to adapt segmentation parameters to current input image conditions. Subsequent feature extraction and model matching are carried out for each connected component which passes through the size filter based on the expected size of objects of interest in the image. The highest matching confidence is taken as the reinforcement signal. Learning takes place as a result of interactions between segmentation and model matching.

Significant differences in characteristics exist between an image and its subimages, so operating conditions are tuned to these differences to achieve optimal performance of segmentation and model matching. For example, to recognize two objects in an image or a single object at different locations, it is often difficult, if not impossible, to meet all requirements with one process. It is essential to localize computation to meet each individual requirement. Thus, we adopt a control that switches between global and local segmentation phases based on the quality of image segmentation and model matching.

The reinforcement learning integrated image segmentation and object recognition system is designed to be fundamental in nature and is not dependent on any specific image segmentation algorithms or type of input images. Reinforcement learning requires only the goodness of the performance rather than the details of algorithms that produce the results. 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 can be expected to learn much faster [2]. In this sense, the system is independent of a particular segmentation algorithm used.

1.2 Related work and our contributions

There is no published work on reinforcement learning integrated image segmentation and object recognition using multiple feedbacks. Bhanu and Lee [9] presented an image segmentation system which incorporates a genetic algorithm to adapt the segmentation process to changes in image characteristics caused by variable environmental conditions. In
their approach, multiple segmentation quality measures are used as feedback. Some of these measures require ground-truth information which may not be always available. Peng and Bhanu [2] presented an approach in which a reinforcement learning system is used to close the loop between segmentation and recognition, and to induce a mapping from input images to corresponding segmentation parameters. Their approach is based on global image segmentation which is not the best way to detect 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 which makes it problematic for practical application of computer vision.

For object recognition applications, the efficiency of the learning techniques is very important. How to add bias, a prior or domain knowledge in a reinforcement learning based system is an important topic of research in reinforcement learning [3][7][8]. For the RATLE system, Maclin and Shavlik [3] accept “advice” expressed in a simple programming language. This advice is compiled into “knowledge-based” connectionist Q-learning network. They show that advice-giving can speed up Q-learning when the advice is helpful (though it need not be perfectly correct). When the advice is harmful, back propagation training quickly overrides it. Dorigo and Colombetti [7] show that by using a learning technique called learning classifier system (LCS), an external trainer working within a RL framework can help a robot to achieve a goal. Thrun and Schwartz [8] have discussed methods for incorporating background knowledge into a reinforcement learning system for robot learning.

In our approach, the edge-border coincidence is used to locate an initial good point from which to begin the search through weight space for high matching confidence values. Although as a segmentation evaluation measure the edge-border coincidence is not as reliable as the matching confidence, 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. It is an inexpensive way to arrive at an initial approximation to a set of segmentation parameters that gives rise to the optimal recognition performance. The control switches between global and local segmentation processes to optimize recognition performance. To further speed up the learning process the reinforcement learning is biased when the model matching confidence or the edge-border coincidence is used as the reinforcement signal (note that the reinforcement learning is unbiased initially when the edge-border coincidence is used as the reinforcement signal). We achieve better computational efficiency of the learning system and improved recognition rates compared to the system developed by Peng and Bhanu [2].

The original contributions of the reinforcement learning integrated image segmentation and object recognition system presented in this paper are:

- To achieve robustness for image recognition system operating in real world, model matching confidence is used as feedback to influence the image segmentation process, and thus provide an adaptive capability.
- A RL system based on a team of learning automata is applied to represent and update both global and local image segmentation parameters. The learning system optimizes segmentation performance on each individual image and accumulates segmentation experience over time to reduce the effort needed to optimize future unseen images.
- Edge-border coincidence, as a segmentation evaluation measure, reduces computational costs by avoiding expensive model matching, especially during earlier stages of learning.
- Learning local segmentation parameters on subimages, which may potentially contain objects, improves the performance of object recognition system.
- Explicit bias is used in the RL based system to speed up the learning process for adaptive image segmentation.

2 Technical Approach

The goal of our system is to maximize the model matching confidence by finding a set of image segmentation algorithm parameters for a given recognition task. To reduce the computational expense of model matching, the edge-border coincidence is first used as evaluation function to find a set of parameters from which to begin the learning. The segmentation process has two distinct phases: global and local. While global segmentation is performed for the entire image, local segmentation is carried out only for selected subimages. For a set of input images, the system takes inputs sequentially. This is similar to human visual learning process, in which the visual stimulus are presented temporally in a sequential manner. For the first input image, since the system has no accumulated experience, we initialize the system using random value of weights in the unbiased stochastic RL algorithm. For each input image thereafter, the learning process starts from the set of segmentation parameters learned based on all the previous input images. The following are the main steps of our learning algorithm:

Initial Approximation. The edge-border coincidence is used as a short term reinforcement during earlier stages of learning to drive weight changes
without going through the expensive model matching process. Once the edge-border coincidence has exceeded a given threshold, the weight changes will be driven by the matching confidence, which requires more expensive computation of feature extraction and model matching.

**Learning Global Segmentation.** A network of biased Bernoulli units generates a set of segmentation parameters from which segmentation is performed on the entire image. The evaluation of the segmentation process is provided by the model matching confidence, which is then used to drive changes to the weights according to the reinforcement learning algorithm. We assume that we have a prior knowledge of the size of objects of interest in the images. For those connected components which 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 the reinforcement to the learning system. If the highest matching confidence level is above a given switching threshold, we focus image segmentation and model matching on the connected component and switch to the local search process.

**Learning Local Segmentation.** Once a connected component has been extracted from the input image, the local search begins to find the best fit parameters for the subimage. It starts from the current estimate of weights that resulted from global learning. Similar to global learning, the matching confidence is used to update the weights estimate, until the matching confidence reaches the accepting threshold (0.8 in our experiments) or the number of iterations reaches the MaxLocal (in our experiments, it is set at 20). If after MaxLocal loops, the matching confidence is still under the accepting threshold, we switch back to the global learning process, continue the learning from where we switched to the local search process. If the matching confidence reaches the accepting threshold, the learning process for the current input image is terminated.

2.1 **Phoenix image segmentation algorithm**

Since we are working with color imagery in our experiments, we have selected the Phoenix segmentation algorithm [16] [18] developed at Carnegie-Mellon University and SRI International. The Phoenix segmentation algorithm has been widely used and tested. 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. To represent 4 parameters, we need a total of 20 Bernoulli units. More details about Phoenix are given in the report by Laws [16].

### Table 1: Ranges for selected Phoenix parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sampling Formula</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hsmooth: h ∈ [0 : 31]</td>
<td>hsmooth = 1 + 2 x h</td>
<td>1 - 63</td>
</tr>
<tr>
<td>Maxmin: mm ∈ [0 : 31]</td>
<td>mm = ln(100) + 0.05 x mm</td>
<td>100 - 471</td>
</tr>
<tr>
<td>Splitmin: sm ∈ [0 : 31]</td>
<td>sm = exp(ep) + 0.3</td>
<td>9 - 71</td>
</tr>
<tr>
<td>Height: h ∈ [0 : 31]</td>
<td>height = 1 + 2 x h</td>
<td>1 - 63</td>
</tr>
</tbody>
</table>

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. To achieve this objective, we introduce a secondary feedback signal - segmentation evaluation that evaluates the image segmentation quality. There are a large number of segmentation quality measures that have been suggested. The segmentation evaluation we selected is the *edge-border coincidence* [9][17], which 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. In this approach, we use the Sobel edge detector to compute the necessary edge information. Edge-border coincidence is defined as follows. 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:

\[
\text{Edge – border coincidence} = \frac{n(E \cap S)}{n(E)},
\]

where \( n(A) \) is the number of elements in set \( A \).

Figure 2 shows the Sobel edge image of an experimental indoor color image and the boundaries of the segmented image using the Phoenix segmentation algorithm. The edge-border coincidence for the segmented image is 0.6825. Segmentation evaluation indicates the quality of the segmentation process. Matching confidence, the recognition system's output, indicates the confidence of the model matching process, and indirectly shows the segmentation quality of the recognized object. It is possible that
segmentation evaluation is high and matching confidence level is low, or segmentation evaluation is low and matching confidence is high. Figure 3(a) shows that global segmentation evaluation is not well correlated with matching confidence. However, local segmentation evaluation, which measures the overlap between the edges and region borders of a subimage, is strongly correlated to the matching confidence, as shown in Figure 3(b).

Although the global segmentation evaluation does not correctly predict the matching confidence, for our purpose it is sufficient to drive initial estimates. If the edge-border coincidence is under a 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 feedback signal until the edge-border coincidence is greater than the threshold. At that time, the segmentation performance will be determined completely by the model matching.

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. It is appropriately thought of as a class of problems, rather than as a set of techniques [4]. This type of learning has a wide variety of applications, ranging from modeling behavior learning in experimental psychology to building active vision systems. The term reinforcement comes from studies of animal learning in experimental psychology. 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 real world.

There are several reasons why we apply reinforcement learning in our 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. In the object recognition system, model matching confidence indirectly evaluates the performance of image segmentation and feature extraction processes. It is a natural choice to select matching confidence as a reinforcement signal. Second, convergence is guaranteed for several reinforcement learning algorithms. Third, reinforcement learning is well suited to the multi-level object recognition problems in image understanding. It can systematically assign rewards to different levels in a computer vision system.

The particular class of reinforcement learning algorithms employed in our system is the connectionist REINFORCE algorithm [11], where units in such a network are Bernoulli quasi-linear units. Figure 4 shows the basic structure of a Bernoulli unit. A team of five independent Bernoulli units represent a segmentation parameter with 32 possible values. The output of each unit is either 1 or 0, determined stochastically using the Bernoulli distribution with probability mass function $p = f(s)$, where $f$ is the
logistic function. For such an unit, $p$ represents the probability of choosing 1 as its output value.

$$f(s) = \frac{1}{1 + e^{-s}}, \quad \text{where} \quad s = \sum_j w_{ij} x_j$$

where $w_{ij}$ is the weight of the $j$th input for unit $i$, and $x_j$ is the $j$th input value for each unit. In the reinforcement learning paradigm, the learning component uses the reinforcement $r(t)$ to drive the weight changes according to a particular reinforcement learning algorithm used by the network. The specific algorithm we used has the following form: for each unit, at the $t$th time step, after generating output $y_i(t)$ and receiving reinforcement signal $r(t)$, increment each weight $w_{ij}$ by

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

where $\alpha$ is the learning rate, $\delta$ is the weight decay rate, $x_j$ is the input to each Bernoulli unit, $y_i$ is the output of the $i$th Bernoulli unit. The term $r(t) - \bar{r}(t-1)$ is called the reinforcement factor, and $\bar{y}_i(t) - \bar{y}_i(t-1)$ is the eligibility of the weight $w_{ij}$. $\bar{r}(t)$ is the exponentially weighted average of prior reinforcement values,

$$\bar{r}(t) = \gamma \bar{r}(t-1) + (1 - \gamma)r(t), \quad \text{with} \quad \bar{r}(0) = 0$$

$\gamma$ 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 [11] such that it statistically climbs the gradient of expected reinforcement in weight space. The weight decay is used as a simple method to force the sustained exploration of the weight space.

Note that a team of 20 Bernoulli units represents the four image segmentation parameters selected for learning. Each bit of a parameter is independent of each other. Thus, it allows us to search the parameter space thoroughly.

2.4 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. In Figure 1, the region filter selects those connected components whose areas are in the expected range. For example, in our experiment on indoor images, the cup is the target object. The expected area is from 200 to 450 pixels. Figure 5 shows the boundaries of a segmented image, selected regions whose areas are in the expected range, and the polygon approximation of these regions. The polygon approximation is implemented by calling the polygon approximation routine in Xhoros [12]. 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. The polygon approximation has a fixed set of parameters:

- Minimal segment length for straight line - 5. When the estimated straight line has a length less than this threshold, it is skipped over.
- Elimination percentage - 0.1. Percentage of line length rejected to calculate parameters of the straight line.
- Approximation error - 0.6. Threshold Value for the approximation error. When the calculated error is greater than this value, the line is broken.

Model matching employs a cluster-structure matching algorithm [14] which 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. Basically, the technique consists of three steps: clustering of border segment transformations; finding continuous sequences of segments in appropriately chosen clusters; and clustering of sequence average transformation values. More details about this algorithm are given in [14].

2.5 Biased reinforcement learning for image segmentation

In the RL algorithm as described in section 2.3, each of the bits of each of the parameters is independent. The output of each bit depends on the value of $p$, which represents the probability of an unit to choose
1 as its output. In the initialization phase, we use the unbiased RL algorithm in which the output of each bit of a parameter is determined in the following way:

\[
y_i = \begin{cases} 
1 & \text{with probability } p \\
0 & \text{with probability } 1 - p
\end{cases}
\]

It 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-payoff 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-payoff 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 slower fashion as:

\[
y_i = \begin{cases} 
1 & \text{if } p > 0.5 \\
0 & \text{otherwise}
\end{cases}
\]

and other bits use the same rule as described in the unbiased RL algorithm. Figure 6 shows the experimental results of the two schemes on the image shown in Figure 2(a). In this experiment, we only apply the initialization followed by global learning without switching between global and local learning. The results show that the biased RL algorithm demonstrates a speedup of 2 – 3.

### 2.6 Algorithm description

Figure 7 shows the implementation of our algorithm. The algorithm works by switching between global and local segmentation. Initially, if the system has no accumulated knowledge, the edge-border coincidence is used as the evaluation function to search a set of image segmentation parameters using unbiased reinforcement learning algorithm. Otherwise, the input image is segmented using the set of parameters learned from previous images. EB1 and EB2 are two thresholds for edge-border coincidence. During the initial unbiased reinforcement learning phase, if the edge-border coincidence is greater than EB1 ( = 0.5 in our experiments), then we can start.
the learning process with a high expectation to generate good recognition results. During the global segmentation phase, if the segmentation quality is less than $EB2 (= 0.4$ in our experiments), the object is less likely to be present in the segmented image, and choosing another set of parameters using the biased RL algorithm with the current reinforcement signal can speed up the process.

In the global segmentation procedure, if the global segmentation loops more than $MaxGlobal$, we conclude that the object does not appear in the image and terminate the learning process for the given input image. For each connected component which passes the region filter, if the matching confidence is greater than $Switch$, then we can switch the control from global to local segmentation. During local segmentation, if the matching confidence reaches $Accept$, we conclude that the connected component is the recognized model object. If the local segmentation loops more than $MaxLocal$, the control will switch back to global segmentation since the object is not likely to be extracted in the subimage and we resume the global segmentation process.

3 Experimental Results

The system is verified through a set of 12 indoor and a set of 12 outdoor color images. These images are acquired at different times and different viewing distances with varying lighting conditions. 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 standard deviations of those 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 indoor images, and 720 weights for outdoor images.

For the team of 20 Bernoulli units, the parameters $\alpha$, $\gamma$, and $\delta$ are determined empirically, and they are kept constant for all images. In our experiments, $\alpha = 0.02$, $\gamma = 0.9$, and $\delta = 0.01$, $EB1 = 0.5$, $EB2 = 0.4$, $MaxGlobal$, $MaxLocal$, and $MaxSeg$ are all set to 20. The threshold for matching confidence $Switch = 0.6$, and $Accept = 0.8$. Threshold used for extracting edges using Sobel operator is set at 200.

3.1 Results on indoor and outdoor images

Figure 8 and 9 show the experimental results on the
set of 12 indoor color images and the set of 12 outdoor color images. For each indoor image, the globally segmented image using the set of learned parameters and the extracted object which has been finally 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 which produces high edge-border coincidence. Figure 8 and 9 also show the global edge-border coincidence, local edge-border coincidence, model matching confidence, and the four learned segmentation parameters after local learning process for each input image.

Figure 10 shows the CPU time for the 12 indoor images and 12 outdoor images for five different runs, and the number of loops for each input image, which is the sum of all the loops involved in the global learning and local learning processes. These two curves show the learning capability of the system, i.e., the system uses less and less CPU time with experience to find a set of segmentation parameters and correctly recognizes the object. The number of learning loops decreases with the accumulation of experience.

### 3.2 Comparison of the two approaches

In this section we compare the performance of our system as shown in Figure 1 with the approach discussed in the paper by Peng and Bhanu [2]. We show the effect of incorporating segmentation evaluation using the edge-border coincidence into the learning system and the impact of global and local segmentations on model matching.

The key differences between the two methods are the introduction of the local segmentation process, the biasing of RL algorithm, and the use of edge-border coincidence as an evaluation of the segmentation performance during earlier stages of learning in order to reduce the computational expense stemming from model matching. The segmentation process alternates between the whole image and its subcomponents. The local segmentation is highly desirable when there are multiple targets or a single target at multiple locations with different local characteristics. It can dramatically improve the recognition performance. The biasing of RL algorithm reduces computational time as illustrated in Figure 6.

In the paper by Peng and Bhanu [2], the matching confidence is the only feedback that drives learning. Although it is undoubtedly the most reliable measure, it is relatively expensive to compute. Here the edge-border coincidence provides us with a cheap way to find a good point from which to begin the more expensive search for high matching confidence values. Figure 11 shows the comparison results of the two schemes: our scheme (scheme 1) and Peng and Bhanu’s scheme (scheme 2). Although good initial estimates may not always result in faster discovery of high matching confidence values, the edge-border coincidence seems to work well in practice for all the problems we have experimented.
4 Conclusions and future work

We have presented a proof-of-the-principle of a general approach for adaptive image segmentation and object recognition. The approach combines a domain independent simple measure for segmentation evaluation (edge-border coincidence) and domain dependent model matching confidence in a reinforcement learning framework in a systematic manner to accomplish robust image segmentation and object recognition simultaneously. Experimental results demonstrate that the approach is suitable for continuously adapting to normal changes encountered in real-world applications.

For adapting to the wide variety of images encountered in real-world applications, we can develop an autonomous gain control system which will allow the matching between different classes of images taken under significantly different weather conditions (sun, cloud, snow, rain) and adapt the parameters within each class of images. We use image context to divide the input images into several classes based on image properties and external conditions, such as time of the day, lighting condition, etc. [9]. 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 start learning process from that set of parameters. This will overcome the problem of adapting to large variations between consecutive images.

The real significance of using a learning network to select segmentation parameters to optimize model matching performance is that interconnections within the network can enforce coordination of the choices made by the output units in order to concentrate the search in suspected high-payoff regions of the parameter space. A network that can coordinate the choices made by the output units should be able to generate certain combinations of bits with greater probability than if their individual components were selected independently. If the network operates in this way it should expect to find high matching confidence values much more quickly than without coordination. We plan to explore these issues in the future.

References


