Learning to Perceive for Autonomous Navigation in Outdoor Environments

Jing Peng and Bir Bhanu
Center for Research in Intelligent Systems
College of Engineering
University of California
Riverside, CA 92521

Abstract

Current machine perception techniques that typically use segmentation followed by object recognition lack the required robustness to cope with the large variety of situations encountered in real-world navigation. Many existing techniques are brittle in the sense that even minor changes in the expected task environment (e.g., different lighting conditions, geometrical distortion, etc.) can severely degrade the performance of the system or even make it fail completely. In this paper we present a system that achieves robust performance by using local reinforcement learning to induce a highly adaptive mapping from input images to segmentation strategies for successful recognition. This is accomplished by using the confidence level of model matching as reinforcement to drive learning. Local reinforcement learning gives rises to better improvement in recognition performance. The system is verified through experiments on a large set of real images of traffic signs.

1 Introduction

Sensing and perception are of paramount importance to any cognitive automaton. Acquiring such abilities is a prerequisite for autonomous platforms that must operate in a dynamic environment. This objective, however, can be challenging in real world navigation applications due to the presence of clutter, object occlusion, data uncertainty, limited *a priori* model information, and changes in the environmental conditions that can not be controlled in the outdoor scenarios. As an example, suppose an autonomous platform is tasked with a mail delivery mission to each department on a typical university campus. For this scenario, the platform must recognize traffic signs and act accordingly, for example, stop at stop signs. Some sample images are shown in Figure 1. There exists a wide variety of real world conditions under which the platform may come to a stop sign. As such, it is difficult or even impossible to develop a perception strategy with fixed parameters and algorithms that performs reliably in dynamic environments. Real world considerations mentioned above must be taken into account if the platform is to act intelligently in an autonomous fashion in a dynamic environment.

In this paper we present a method, based on our research into an outdoor navigation task, that allows the landmark recognition system to adjust itself automatically to accommodate environmental variations to provide robust performance. With this method, the level of achievable recognition performance is increased through on-line acquisition/recognition of model via direct interaction with the world, thereby guiding action for navigation.

Our method, called *local reinforcement learning*, combines *local learning* and *reinforcement learning* into a single mechanism. Like local learning, the method first partitions the input space into a set of disjoint local regions using techniques that determine the number of regions based on indicators that entail intra-region cohesion and inter-region isolation. Within each region, reinforcement learning then induces a mapping from input images to recognition strategies using feedback from recognition performance. In contrast, typical case-based learning (CBL) - a local learning technique with local constant approximation - uses a representative case associated with each region to guide future performance. Such a representative case, however, might not be sufficient to accommodate within its region variations that are either inherent or caused by inadequate region indicators.

The original contribution of this paper is to provide an efficient learning method for constructing perception strategies for autonomous navigation that are highly adaptive to changes in environmental conditions. We begin with a discussion of the need for

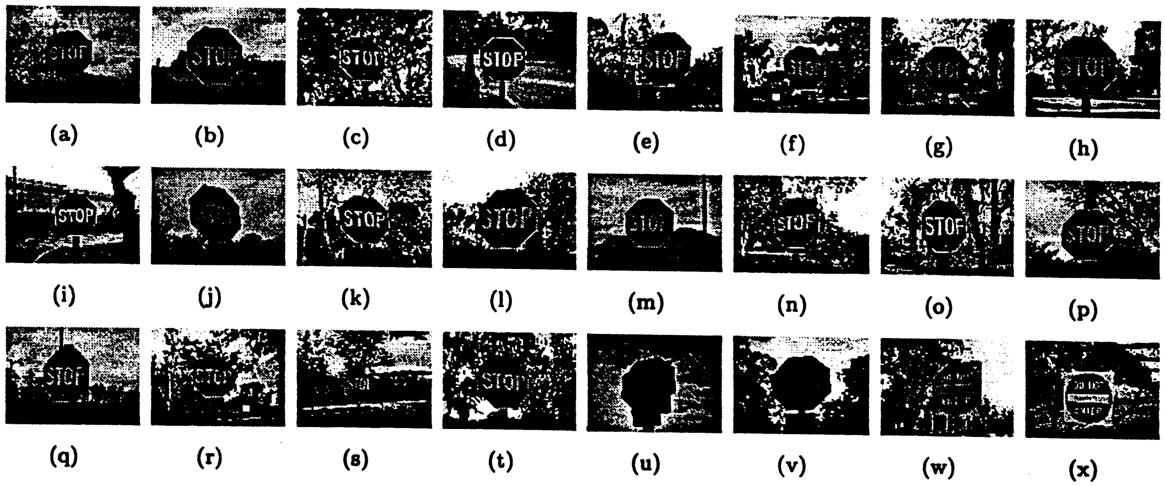


Figure 1: Sample color images with varying outdoor conditions.

learning in autonomous navigation, after which we describe the technical details of the proposed learning technique. We then present several experimental studies evaluating our method. These results are discussed and analyzed. Finally, we conclude with the key aspects of this paper.

2 Why Learning?

A typical 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. All subsequent image interpretation tasks including feature extraction and model matching, rely heavily on the quality of the image segmentation process. Generally, this in turn spells out the difference between success and failure in vision-based autonomous navigation.

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 serve 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. As an example, Figure 2 shows segmentation of images shown in Figure 1 obtained using the *Phoenix* algorithm [8] with default parameters. From these segmentation results, no algorithm would be able to perform model matching with sufficient confidence to recognize the stop sign, i.e., the octagon. Moreover, purely geometric or physics-based invariant approaches, *without* learning, are not sufficient to recognize objects under a wide variety of situations encountered in real-world navigation [5, 7].

One might contemplate the idea of using color information for detecting and recognizing objects, such as stop signs. However, there are three major difficulties associated with such simple, color-based techniques. ***First***, there are times at which color features can not be reliably detected. For example, the images shown in Figures 1(u) and (v) do not reveal any color information for recognizing the stop sign. This happens when one is moving towards the sun. ***Second***, there are scenarios where the same characteristic color (red) information is associated with traffic signs having completely different meaning. For example, in order to differentiate between "DO NOT ENTER" (Figures 1(w) and (x)) and "STOP" signs, color information alone is not sufficient. Additional information, such as shape, must be included. ***Third***, as shown in Figure 2, sophisticated, color-based segmentation techniques, such as *Phoenix*, can fail if they do not adapt to changes in environmental conditions including lighting, color, distance and orientation.

In ordor to achieve reliable performance in real

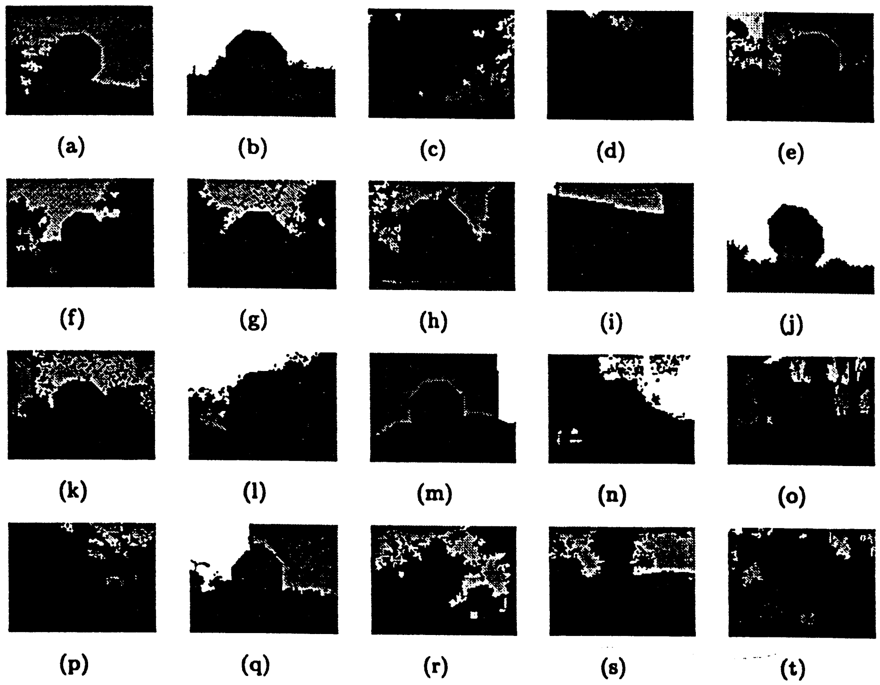


Figure 2: Segmentations of images shown in Figure 1 using the *Phoenix* algorithm with default parameters

world navigation application, therefore, a need exists to apply learning techniques that are capable of inducing a highly adaptive mapping from input images to segmentation strategies needed for successful recognition. This paper presents a computationally efficient local reinforcement learning technique that is able to compute such a mapping from actual data. It takes the output of the recognition algorithm and uses it as a feedback to influence the performance of the segmentation process. As a result, segmentation strategies, conditioned on current inputs, for performing a particular task are chosen more judiciously, i.e., so as to maximize the confidence of model matching.

Note that both adaptive and learning systems can automatically adjust their internal representations, and both make use of performance feedback information. The major differences, however, are a matter of degree, emphasis, and intended purpose. A vision system that treats every distinct input image as novel is limited to adaptation, whereas a system that correlates past experiences with past inputs, and one that can recall and exploit those experiences, is capable of learning. Since, in the process of learning, the learning system must be capable of adjusting its memory to accommodate new experiences, a learning system must, in some sense, incorporate an adaptive capability. However, the design and intended purpose of a learning system require capabilities beyond that of adaptation. In this paper, our focus is on learning systems.

3 The Approach

We develop a general approach for achieving robust image segmentation and object recognition by using local reinforcement learning that combines local learning and reinforcement learning in a novel way. The system's functional structure is shown in Figure 3. 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 image segmentation component extracts meaningful objects from input images, feature extraction component performs polygonal approximation of connected components, and the model matching component tells us which regions in the segmented image contain the model object by generating a real valued matching confidence indicating the degree of success. Reinforcement learning module then uses this confidence value as feedback to induce a mapping from images to segmentation strategies within each local region created from partitioning of the feature space. The goal is, therefore, to maximize the matching conidence by finding a set of segmentation algorithm parameters for the given recognition task.

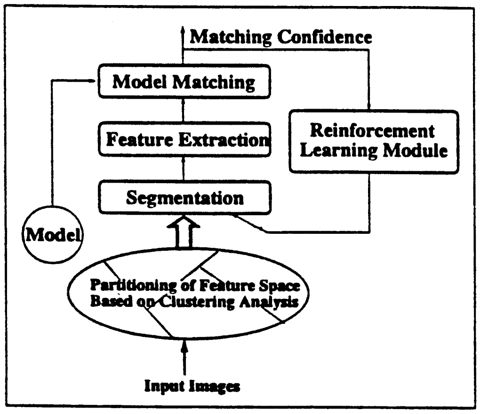


Figure 3: Reinforcement learning based system for recognition.

There are good reasons for using reinforcement learning in our object recognition 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 performs efficient hill-climbing in a statistical sense without excessive demand for computational resources. Furthermore, it can generalize over unseen images. *Fourth*, tt is feasible to construct fast, parallel devices to implement this technique for real-time applications. It thus fits our goal nicely here. Likewise, local learning has the advantage of avoidance of negative spatial cross-talk typically associated with global learning techniques, because mappings are constructed separately within each local region in the feature space. Furthermore, local learning often gives rises to better improvement in recognition performance, as we shall see later. Note that the integration of the two paradigms (local learning and reinforcement learning) at the algorithmic level makes it possible to take advantages of some of the best features of both worlds.

3.1 Related Work

Robot learning and landmark recognition are active areas of research [4, 11, 15, 18]. The challenge is to extend operating conditions of a mobile robot. Adaptation and learning play an important role for achieving the robustness of algorithms. The work presented in this paper is most closely related to earlier work by the authors [12], in which they describe a reinforcement learning system that uses recognition output as feedback to guide the segmentation process. However, their method is global in that only a single mapping is induced over the entire input space. In addition, their system was evaluated only on a small number of images. In this work, we use simulated images with controlled statistics to show that our method can indeed learn correct segmentation strategies for recognition of objects. Further, we show that local learning described here can outperform the global learning method using empirical results based on a large set of real images of traffic signs.

An adaptive approach to image segmentation is proposed by Bhanu and Lee [1]. Their system uses genetic and hybrid algorithms for learning segmentation parameters. However, the recognition algorithm is not part of the evaluation function for segmentation in their system. The genetic or hybrid algorithms simply search for a set of parameters that optimize a prespecified evaluation function (based on global and local segmentation evaluation) that may not best serve the overall goal of robust object recognition. Furthermore, their work assumes that the location of the object in the image is known for specific photointerpretation application. In our work, we do not make such an assumption. We use explicit geometric model of an object, represented by its polygonal approximation, to recognize it in the image.

The ALVINN (Autonomous Land Vehicle in a Neural Network) system [13] is a neural network based system that learns to drive in a variety of autonomous navigation scenarios, including single-lane paved and unpaved roads, multilane lined and unlined roads, and on- and off-road environments having obstacles. ALVINN's flexible operating conditions can be attributed to the fact that it does not rely on a precise model of image features for navigation. Instead, it acquires this model information through learning under various driving conditions. However, this and other supervised learning techniques rely heavily on a competent, external teacher to provide training examples. The disadvantage of this "external teacher" method is that such a competent teacher may not be always available. In addition, ALVINN is designed to follow road marks such as lines. There are situations where road marks either do not exist or are hard to follow. In contrast, our system is designed to recognize landmark such as traffic signs under changing environmental conditions.

Zheng et al. [18] describe an adaptive system for traffic sign recognition on motorways. The system uses color image segmentation and color topology analsis for traffic sign detection. It then uses weighted K nearest neighbor rules for classification. The main concern with this type of "lazy" learning is memory requirement. A memory consolidation process must be in place to satisfy practical constraints. In contrast, our system learns to induce a compact, local mapping from input images to segmentation strategies within each partition. In addition, in our system learning is driven by the output of model matching.

4 Local Reinforcement Learning

4.1 Local Learning

Local learning first partitions the space of input variables into a set of local regions (clusters). The method then learns a separate mapping individually in each local region. The partitioning procedure used in this paper is the K-means method [9]. The number of regions, K, is determined experimentally using the Calinski-Harabasz Index [3] as an indicator. The Calinski-Harabasz Index is chosen because it gives the best performance among 30 indicators [10]. The Index is defined as

$$I = (Trace(B)/(K-1))/(Trace(W)/(n-K)) \quad (1)$$

where n is the number of sample data, and $Trace$ denotes the trace operation of a matrix. B and W are the between and within cluster sum of squares and cross product matrices from multivariate statistics, respectively. If d^2 denotes the mean of all $n(n-1)/2$ squared distances, and d_l^2 that of the $n_l(n_l-1)/2$ squared distances within the lth class ($l = 1, 2, \cdots, K$), then (1) can be computed according to

$$I = (d^2 + \frac{n-K}{K-1}A_K)/(d^2 - A_K) \quad (2)$$

where

$$A_K = \frac{1}{n-K}\sum_{l=1}^{K}(n_l - 1)(d^2 - d_l^2). \quad (3)$$

Larger Index values indicate greater class cohesion and external isolation. Once the number of regions (clusters) has been determined, a local mapping in each region is constructed using connectionist reinforcement learning. For a given input, generalization is made by searching for the nearest cluster and then applying the mapping associated with the cluster to compute segmentation parameters.

4.2 Connectionist Reinforcement Learning

The particular class of reinforcement learning algorithms employed in each local region for our object recognition system is the connectionist REINFORCE algorithm [16], where units in such a network are *Bernoulli semilinear units*, in that the output of such a unit, i, is either 0 or 1, determined stochastically using the Bernoulli distribution with parameter $p_i = f(s_i)$, where f is the logistic function

$$f(s_i) = 1/(1 + \exp(-s_i)) \quad (4)$$

and $s_i = \sum_j w_{ij}x_j$ is the usual weighted summation of input values to that unit. For such a unit, p_i represents its probability of choosing 1 as its output value. The left graph in Figure 4 depicts a connectionist reinforcement learning system and the right graph shows a Bernoulli semilinear unit in such a system.

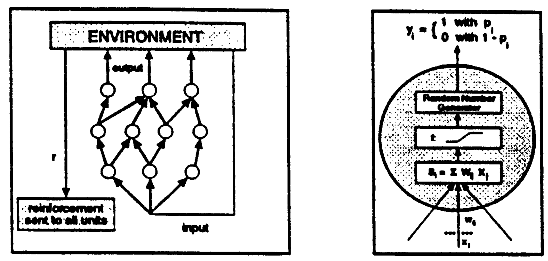


Figure 4: Left: Connectionist reinforcement learning system. Right: Bernoulli semilinear unit.

In the general reinforcement learning paradigm, the network generates an output pattern and the environment responds by providing the reinforcement r as its evaluation of that output pattern, which is then used to drive the weight changes according to the particular reinforcement learning algorithm being used by the network. For the Bernoulli semilinear units used in this research, the REINFORCE algorithm prescribes weight increments equal to

$$\Delta w_{ij} = \alpha(r - b)(y_i - p_i)x_j \quad (5)$$

where α is a positive learning rate, b serves as a *reinforcement baseline*, x_j is the input to each Bernoulli unit, y_i is the output of the ith Bernoulli unit, and p_i is an internal parameter to a Bernoulli random number generator.

It can be shown [16] that, regardless of how b is computed, whenever it does not depend on the immediately received reinforcement value r, and when r is

- Partition training data into K classes
- For each class do:
  - LOOP:
    1. $rr = 0$ (rr: average matching confidence)
    2. For each image i in the training set do
    (a) Segment image i using current segmentation parameters
    (b) Perform noise clean up
    (c) Get segmented regions (also called blobs or connected components)
    (d) Perform feature extraction for each blob to obtain token sets
    (e) Compute matching of each token set against stored model, return the highest matching confidence r
    (f) Update $\mathbf{W}$ according to eq. (7) using r as reinforcement
    (g) Compute new parameters for the segmentation algorithm
    (h) $rr = rr + r$
  - UNTIL number of iterations is equal to N or $rr/n \geq R_{th}$ (Threshold)

Figure 5: Main steps of the local reinforcement learning algorithm.

sent to all the units in the network, such an algorithm satisfies

$$E\{\Delta\mathbf{W}|\mathbf{W}\} = \alpha\nabla_{\mathbf{W}}E\{r|\mathbf{W}\} \quad (6)$$

where E denotes the expectation operator, $\mathbf{W}$ represents the weight matrix of the network, and $\Delta\mathbf{W}$ is the change of the weight matrix. A reinforcement learning algorithm satisfying the above equation has the convergence property that the algorithm statistically climbs the gradient of expected reinforcement in weight space. For adapting parameters of the segmentation algorithm, it means that the segmentation parameters change in the direction along which the expected matching confidence increases.

The specific algorithm we use here has the following form: At the t^{th} time step, after generating output $\mathbf{y}(t)$ and receiving reinforcement $r(t)$, i.e., the confidence level indicating the matching result, increment each weight w_{ij} by

$$\Delta w_{ij}(t) = \alpha(r(t)-\overline{r}(t-1))(y_i(t)-\overline{y}_i(t-1))x_j-\delta w_{ij}(t) \quad (7)$$

where α, the learning rate, and δ, the weight decay rate, are parameters of the algorithm. The term $(r(t) - \overline{r}(t-1))$ is called the *reinforcement factor* and $(y_i(t) - \overline{y}_i(t-1))$ the *eligibility* of the weight w_{ij} [16]. Generally, the eligibility of a weight indicates the extent to which the activity at the input of the weight was connected in the past with unit output activity. Note that this algorithm is a variant of the one described in equation (5), where b is replaced by $\overline{r}$ and p_i by $\overline{y}_i$.

$\overline{r}(t)$ is the exponentially weighted average, or *trace*, of prior reinforcement values

$$\overline{r}(t) = \gamma\overline{r}(t-1) + (1-\gamma)r(t) \quad (8)$$

with $\overline{r}(0) = 0$. The trace parameter γ was set equal to 0.9 for all the experiments reported here. Similarly $\overline{y}_i(t)$ is an average of past values of y_i computed by the same exponential weighting scheme used for $\overline{r}$. That is,

$$\overline{y}_i(t) = \gamma\overline{y}_i(t-1) + (1-\gamma)y_i(t). \quad (9)$$

Note that equation (6) does not depend on eligibility. Note also that p_i in (5) is the theoretical mean of y_i, whereas $\overline{y}_i$ in (7) and (9) is the actual estimate. In addition, The weight decay (the second term in equation (7) is used as a simple method to force the sustained exploration of the parameter space. Empirical study shows superior performance with this form of weight update [17]. The local learning algorithm is shown in Figure 5, where n is the number of training images in each local region.

5 Empirical Evaluation

This section describes empirical results evaluating the performance of our system on a large set of outdoor color images. The system has been implemented on a SUN Ultra-1 workstation. For the real images the segmentation algorithm takes about one quarter of per iteration time. Programming optimizations can reduce the expense per iteration further for real-time performance.

The *Phoenix* algorithm [8] was chosen as the image segmentation component in our system. *Phoenix* works by splitting regions using a histogram for color features. We have chosen the *Phoenix* algorithm because it has been widely used, refined and well documented. *Phoenix* has been extensively tested on color imagery and has become an important part of the DARPA's Image Understanding testbed at SRI International. Note that our system is designed to be independent of segmentation algorithms and images. It is a basic learning framework that can be applied to any object recognition system.

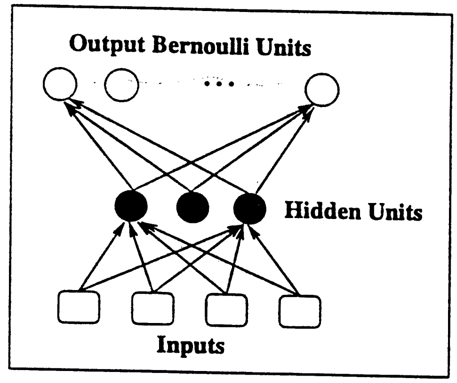
The *Phoenix* algorithm has a total of fourteen adjustable parameters. The four most critical ones that affect the overall results of the segmentation process are used in learning. These parameters are *Hsmooth*, *Maxmin*, *Splitmin*, and *Height*. *Hsmooth* is the width of the histogram smoothing window, where smoothing is performed with a uniformly weighted moving average. *Maxmin* defines the peak-to-valley height ratio threshold. Any interval whose peak height to higher shoulder ratio is less than this threshold is merged with the neighbor on the side of the higher shoulder. *Splitmin* defines the minimum size for a region to be automatically considered for splitting. This is an absolute value, not a percentage of the image area. *Height* is the minimum acceptable peak height as a percentage of the second highest peak. Table 1 shows sample ranges for each of these parameters. The resulting search space is about one million sample points.

The feature extraction consists of finding polygon approximation tokens for each of the regions obtained after image segmentation. The polygon approximation is obtained using a split and merge technique [2] that has a fixed set of parameters. Object recognition employs a cluster-structure matching algorithm [2] that is based on the clustering of translational and rotational transformations between the object and the model for recognizing 2-D and 3-D objects. It outputs a real number indicating the confidence level of the matching process. This confidence level is then used as a reinforcement signal to drive learning. These algorithms were chosen simply because they are avaiable in house.

5.1 Experimental Results

The experiment described here consists of 500 images, some of which are shown in Figure 1. These images are collected in late afternoon over several days (including a rainy day) using a Canon PowerShot 600 digital camera. They are taken in a variety of locations in Southern California. These images simulate an autonomous navigation scenario in which an autonomous vehicle must be able to recognize the stop sign. The size of the images is 78 by 104 pixels.

Eighty images are randomly selected as training data, and the rest (420) as testing data. A principal component analysis is carried out using the red color component. Red component of each image is projected onto the subspace spanned by the first 4 eigen vectors corresponding to four largest eigen values. These inputs are normalized to lie between 0 and 1.



(a)

Figure 6: A connectionist reinforcement learning network.

(1) Local Learning: The training data are first clustered using the K-means algorithm based on the eigen inputs. The K-means algorithm was repeatedly applied to the training data with varying K. The K value that attained the largest Calinski-Harabasz Index was selected as the final cluster number (4 in this experiment). The resulting clusters contain 25, 26, 13, and 16 training images, respectively. Within each cluster, a network having 3 hidden Bernoulli units and 20 output Bernoulli units that encode the four *Phoenix* parameters was trained using the local learning algorithm described in Figure 5. Each hidden unit takes four eigen inputs and there are no connections from inputs to output units. Because of the independence of the output units, the effective number of weights in the network is 19 (4 (input weights) $\times$ 3 (hidden units) + 3 (hidden to output weights) + 4 (biases)). Figure 6 depicts such a network.

(2) Global Learning: A global network, simi-

Table 1: Sample ranges for selected *Phoenix* parameters.

Parameter	Sampling Formula	Test Range
Hsmooth: hsindex $\in [0:31]$	hsmooth=1 + 2 $*$ hsindex	1 – 63
Maxmin: mmindex $\in [0:31]$	ep=ln(100) + 0.05 $*$ mmindex maxmin = exp(ep) + 0.5	100 – 471
Splitmin: smindex $\in [0:31]$	splitmin=9 + 2 $*$ smindex	9 – 71
Height: htindex $\in [0:31]$	height=1 + 2 $*$ htindex	1 – 63

lar to the one shown in Figure 6, is trained on the entire training data to construct a single mapping. The network has 8 hidden Bernoulli units, and 20 output units. The number of hidden units is determined experimentally that achieves the best performance among several trials. In comparison with local learning, the effective number of free parameters in the global network is 49 (4 (input weights) $\times$ 8 (hidden units) + 8 (hidden to output weights) + 9 (biases)).

(3) Simple Case-Based Learning: Instead of constructing a local mapping within each cluster, as is done in the local learning method, the simple CBL method first learns, for each cluster, a set of segmentation parameters achieving the best performance for the image closest to the cluster center. It then stores the set of segmentation parameters in a memory location associated with the cluster center. For a given test image, CBL returns the set of segmentation parameters associated with the cluster that is closest to the input image.

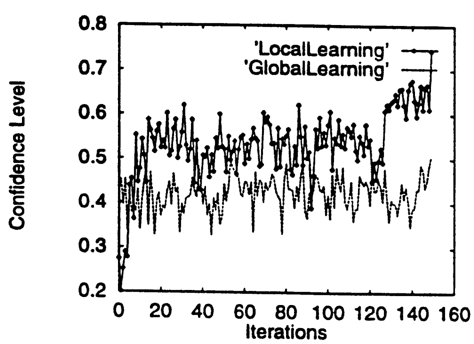


Figure 7: Confidence level received during learning.

Comparison of Learning Methods: In order to evaluate our local learning system for object recognition described here, its performance was compared against several other methods.

(a) Comparison of Local and Global Learning: In the first experiment, each local network was allowed 150 iterations, i.e., 150 sweeps through local training data, and then four confidence values over the four clusters were averaged. In contrast, the global network was given 600 iterations through the entire 80 training images. Confidence values received over every 4 iterations were averaged and plotted. Figure 7 shows the average matching confidence received over time by the two methods. It can be seen that, given the same amount of computation, the local method learned much faster and its confidence value exceeded 0.7, whereas the global confidence value was slightly above 0.5. Furthermore, when applied to 420 test images, the local method achieved an average confidence value of 0.71, whereas the global method only managed to achieve an average value of 0.59. Parameter values used in these experiments are: $\delta = 0.005$ (weight decay rate (7)), $\gamma = 0.85$ ((8) and (9)), and the learning rate $\alpha = 0.2$ for local learning and $\alpha = 0.1$ for global learning.

(b) Comparison of Local and Simple Case-Based Learning: In the second experiment, case-based learning (CBL) was applied to the same task. The CBL method achieved an average confidence value of 0.21 on the testing data, which is far worse than the local learning method. This demonstrates that the local learning approach has the ability to compensates not only variations within each cluster, but also inadequate cluster characterization.

(c) Comparison of Local Learning vs. No Learning: In the final experiment, the *Phoenix* algorithm with default parameters was used. The system was only able to achieve an average confidence value of 0.04, which is extremely poor. Figure 8 shows the segmentation results of the images shown in Figure 1 using the local learning method, from which the sys-

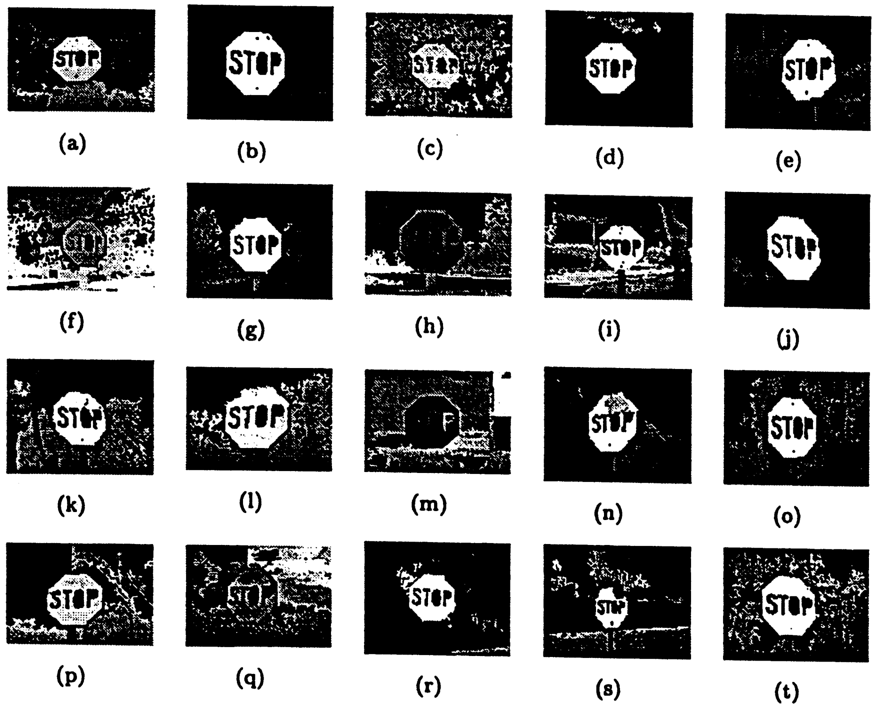


Figure 8: Segmentation of images shown in Figure 1 using the *Phoenix* algorithm with learned parameters

tem was able to recognize the stop sign (octagon) with sufficient confidence (0.71 confidence level on 420 test images). Note that these images are part of the testing data.

5.2 Discussion

In this paper, we have focused our attention on the subject of landmark recognition, an important area of perception for mobile agents. Detection and mobility can be addressed in the following way. Typically, a landmark is viewed from different distances. The landmark representation and associated matching techniques have to correlate with what is observed. As an example, when distances are relatively large, we can use feature clues (e.g., red color for stop sign, yellow for pedestrian crossing, etc.) to localize traffic signs. When we are relatively close we can then use geometric model matching (this is what is done here).

Finally, The images shown in Figure 1 are already "presegmented" in the sense that they all contain a stop sign in roughly the center of the image. It can be argued, however, that normal navigation scenarios in which one reacts to a stop sign often contain the stop sign as a stand alone object. While it is true that a sign taken in a standard urban view with other signs nearby would provide a more compelling test, this would depend, to a large extent, on the capabilities of the segmentation and model matching algorithms used.

6 Conclusion

We have presented a general approach to achieving robust image segmentation and object recognition for autonomous navigation in outdoor environments. The approach systematically uses model matching confidence as feedback in a novel reinforcement learning framework to efficiently learn segmentation parameters and perform object recognition simultaneously. Experimental results demonstrate that the simple approach is very promising in accommodating the wide variety of images encountered in outdoor autonomous navigation.

Acknowledgements

This work was supported by DARPA/AFOSR grant F49620-95-1-0424 and F49620-97-1-0184.

References

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

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

[3] T. Calinski and J. Harabasz, "A denrite method for cluster analysis," *Communications in Statistics 3*, pp. 1-27, 1974.

[4] J.H. Connell and S. Mahadevan (Eds), *Robot Learning*, Boston MA: Kluwer Academic Publishers, 1993.

[5] D. Forsyth, J. Mundy and A. Zisserman, "Transformational invariance: A Primer," *Image and Vision Computing*, Vol. 10, pp. 39-45, Jan. 1992.

[6] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley Publishing Co., 1977.

[7] G. Healey and A. Jain, "Retrieving multispectral satellite images using physics-based invariant representation," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 18, No. 8, pp. 842-846, August 1996.

[8] K. Laws, "The *Phoenix* image segmentation system: Description and evaluation," SRI International Tech. Rep. TR289, December 1982.

[9] J.L. Marroquin and F. Girosi, "Some extensions of the K-means algorithm for image segmentation and pattern classification," A.I. Memo No. 1390, MIT AI Lab, 1993.

[10] G.W. Milligan and M.C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika* 50, pp. 159-179, 1985.

[11] H. Nasr and B. Bhanu, "Landmark recognition for autonomous mobile robots," IEEE Int. Conf. on Robotics and Automation, pp. 1218-1223, 1988.

[12] J. Peng and B. Bhanu, "Closed-loop object recognition using reinforcement learning," Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 538-543, San Francisco, June, 1996.

[13] D.A. Pomerleau, *Neural Network Perception for Mobile Robot Guidence.* Boston, MA: Kluwer Academic Publishers, 1993.

[14] A. Ram and J.C. Santamaria, "Continuous case-based reasoning," To appear in: *Artificial Intelligence*, 1998.

[15] P.E. Trahanias, S. Velissaris and T. Garavelos, "Visual landmark extraction and recognition for autonomous robot navigation," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Grenoble, France 1997.

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

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

[18] J.Y. Zheng, M. Barth and S. Tsuji, "Autonomous landmark selection for route recognition by a mobile robot," IEEE Int. Conf. on Robotics and Automation, pp. 2004-2009, 1991.

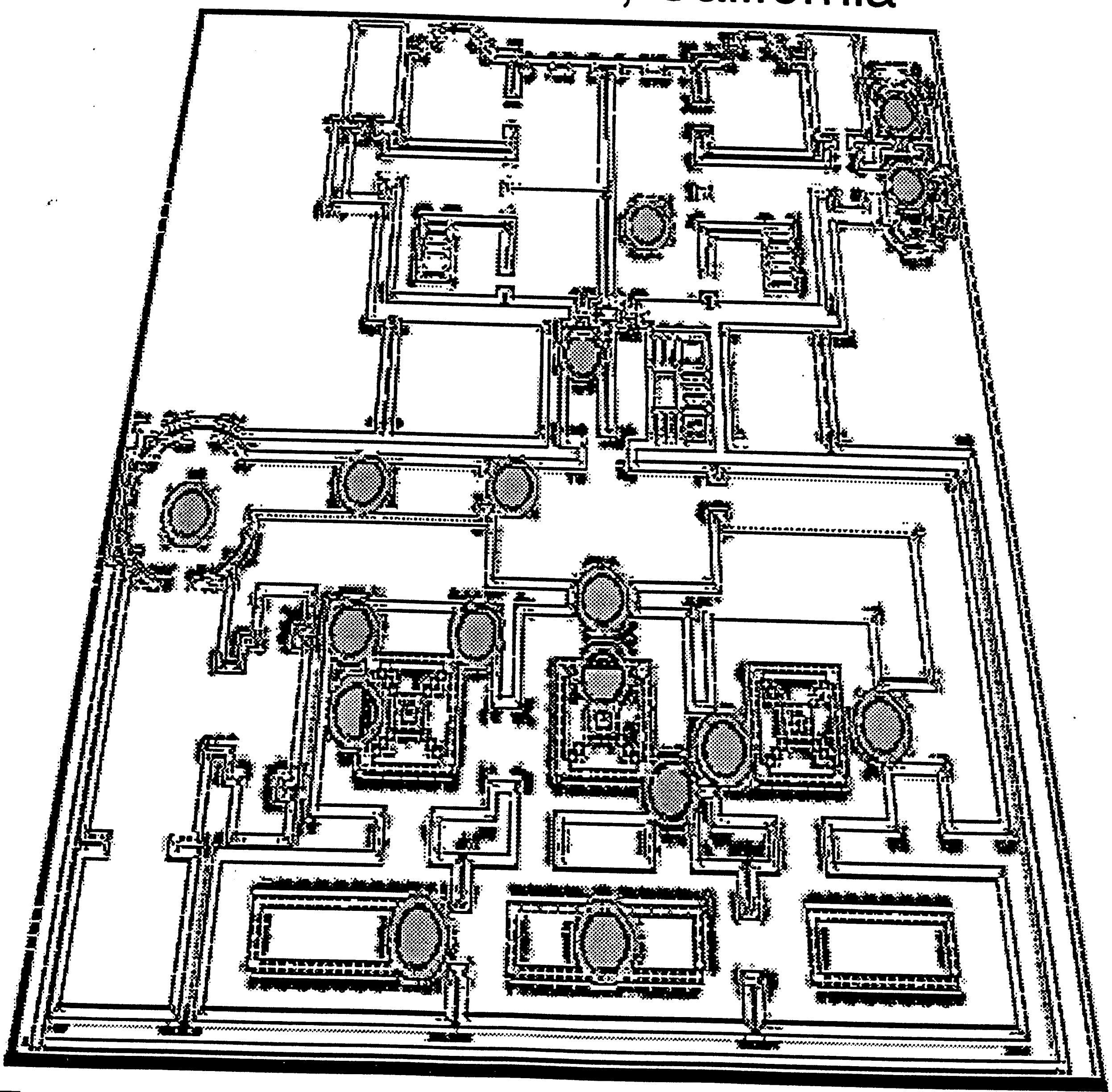
Proceedings of the

IEEE Workshop on

Perception for Mobile Agents

June 1998

Santa Barbara, California



Edited by G. Dudek, M. Jenkin and E. Milios

Proceedings

Workshop on Perception for Mobile Agents (WPMA)

University of California, Santa Barbara

June 26, 1998

Sponsored by

IEEE Computer Society Technical Committee for Pattern Analysis and Machine Intelligence (PAMI)