Local Reinforcement Learning for Object Recognition

Jing Peng and Bir Bhanu
College of Engineering
University of California, Riverside, CA 92521
{jp,bhanu}@vislab.ucr.edu

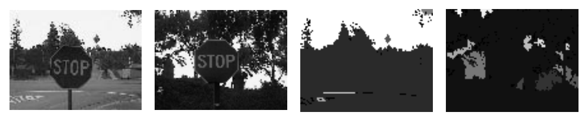
Abstract

Current computer vision systems whose basic methodology is open-loop or filter *type typically use image segmentation followed by object recognition algorithms. These systems are not robust for most real-world applications. In contrast, the system presented here achieves robust performance by using local reinforcement learning to induce a highly adaptive mapping from input images to segmentation strategies. This is accomplished by using the confidence level of model matching as reinforcement to drive learning. The system is verified through experiments on a large set of real images.*

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. 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. 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, Figures 1(a) and (b) show two outdoor color images with varying environmental conditions, and their corresponding segmentations obtained using *Phoenix* [4] with default parameters are shown in Figures 1(c) and (d). From these segmentation results, no algorithm would be able to perform model matching with sufficient confidence to recognize the stop sign.



(a) (b) (c) (d)

Figure 1. (a),(b): Two outdoor color images. (c),(d): Corresponding segmentations using *Phoenix* with default parameters

This paper presents a computationally efficient local reinforcement learning technique that is capable of inducing a highly adaptive mapping from input images to segmentation strategies. 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.

1.1. 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 integration of the two paradigms at the algorithmic level makes it possible to take advantages of some of the best features of both worlds. 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 matching confidence is used as feedback to drive learning in a local reinforcement learning framework. The goal is, therefore, to maximize the matching confidence by finding a set of segmentation algorithm parameters for the given recognition task.

This work is most closely related to earlier work by the authors [5], 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 show that local reinforcement learning described here can outperform the global learning method using empirical results based on a large set of real images.

2. Local Reinforcement Learning

Local learning first partitions the input space into a set of local regions (clusters). These methods then learn a separate mapping individually in each local region. The partitioning procedure used in this paper is the K-means method. The number of regions, K, is determined experimentally using the Calinski-Harabasz Index [3] as an indicator. The Index is defined as $(TraceB/(K-1))/(TraceW/(n-K))$, where n is the number of sample data. B and W are the between and within cluster sum of squares and cross product matrices from multivariate statistics, respectively. 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.

2.1. 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 [6], where units in such a network are *Bernoulli quasilinear units*, in that the output of such a unit is either 0 or 1, determined stochastically using the Bernoulli distribution with parameter $p = f(s)$, where f is the logistic function, $f(s) = 1/(1 + \exp(-s))$ and $s = \sum_i w_i x_i$ is the usual weighted summation of input values to that unit. For such a unit, p represents its probability of choosing 1 as its output value.

For the Bernoulli quasilinear units used in this research, the REINFORCE algorithm prescribes weight increments equal to $\Delta w_{ij} = \alpha(r - b)(y_i - p_i)x_j$, 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 [6] that the algorithm statistically climbs the gradient of expected reinforcement in weight space.

3. Empirical Evaluation

This section describes empirical results evaluating the performance of our system on a large set of outdoor color images. For this experiment, the *Phoenix* algorithm [4] was chosen as the image segmentation component in our system. *Phoenix* works by splitting regions using a histogram for color features. Note that any segmentation algorithm with adjustable parameters can be used in our approach. The *Phoenix* algorithm has a total of fourteen adjustable parameters. The four most critical ones are used in learning. These parameters are *Hsmooth*, *Maxmin*, *Splitmin*, and *Height*. The ranges for each of these parameters are the same as those used in [1]. 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. And 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. Its output is used as reinforcement to drive learning.

The experiment consists of 500 images, some of which are shown in Fig. 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 the 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. Red component of each image is projected onto the subspace spanned by the first four eigenvectors corresponding to four largest eigen values of the red feature plane of the images. These inputs are normalized to lie between 0 and 1. The training data are first clustered using the K-means algorithm based on the eigen inputs. The K value that attained the largest Calinski-Harabasz Index was selected as the final cluster number (4 in this experiment). 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. 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 13.

(a) (b) (c) (d)

Figure 2. (a): Confidence level during learning. (b),(c): Segmentation of the images shown in Fig. 1. (d): Segmentation of a testing image with stop sign partially occluded.

Comparison with Global Learning: In the first experiment, a global network having 8 hidden units (experimentally determined) was trained on the entire training data to construct a single mapping. 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. Fig. 2(a) 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 unseen 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.

Comparison with Case-Based Learning: In the second experiment, case-based learning (CBL) was applied to the same task. The 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. It 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.

Comparison with Default Parameters: 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. Figures 2(b) and (c) show the segmentation results of the images shown in Fig. 1 using the local learning method, from which successful model matching was achieved (above 0.9). Fig. 2(d) shows the segmentation of a testing image in which the stop sign is partially occluded.

4. Conclusion

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

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

[5] 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.

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