there are total 640 images in the database. Also, the number of images in each class varies from 16 to 80. The images in this database are represented by 16 dimensional feature vectors. We use 16 Gabor filters [7] (2 scales and 4 orientations) for feature extraction. The mean and the standard deviation of the magnitude of the transform coefficients are used as feature components after being normalized by the standard deviations of the respective features over the entire images in the database.

5.2 Results

For both problems, each image in the database is selected as a query and top 20 nearest neighbors are returned that provide necessary relevance feedback. The average retrieval precision is summarized in Table 2. There are two rows under each problem in the table. The first row indicates the performance of the method under the condition that the features are normalized so as to have zero mean and unit variance. The second row shows the results obtained by the method, conditioned on the features being scaled to lie between 0 and 1.

Table 2: Average retrieval precision for real data.

UCI Database			
Method	0 (rf)	1 (rf)	Improvement
PFRL (01)	92.10	95.64	6.82
PFRL (*scale*)	92.08	96.05	7.66
MIT Database			
Method	0 (rf)	1 (rf)	Improvement
PFRL (01)	77.05	84.02	14.37
PFRL (*scale*)	78.27	84.44	12.70

The second column in Table 2 shows the average retrieval precision obtained by the method without any relevance feedback (rf). The third column shows the average retrieval precision computed after learning has taken place. That is, relevance feedback obtained from the previous retrieval is used to estimate local feature relevance, hence a new weighting. The last column shows relative performance improvement by the method. It can be seen from Table 2 that our method demonstrate significant performance improvement across the tasks. Furthermore, various normalization techniques do not seem to have a significant impact on the overall ability of each method. Note that the procedural parameters (T (9) and C (13)) input to our method were determined empirically, and they were set to 15 and 16, respectively, in all the experiments reported in this section.

6 Conclusions

This paper presents a novel probabilistic feature relevance learning technique for efficient content-based image retrieval. The experimental results using both simulated and real data show convincingly that learning feature relevance based on user's feedback can indeed improve retrieval performance of an image database system. Furthermore, since the relevance estimate is local in nature the resulting retrieval, in terms of the shape of the neighborhood, is highly adaptive and customized to the query location.

Our retrieval technique learns local feature relevance for each given query. However, it is possible that the knowledge acquired during one retrieval can be gradually collected and it can become part of the database itself through continuous learning. This knowledge can be used in conjunction with case-based learning to achieve generalization in future retrievals in order to further optimize the performance of the system.

References

[1] R.H. Creecy, B.M. Masand, S.J. Smith, and D.L. Waltz, "Trading Mips and Memory for Knowledge Engineering," *CACM*, 35:48-64, 1992.

[2] M. Flickner et al., "Query by Image and Video Content: The QBIC system" *IEEE Computer*, pp. 23-31, September 1995.

[3] J.H. Friedman "Flexible Metric Nearest Neighbor Classification," Tech. Report, Dept. of Statistics, Stanford University, 1994.

[4] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18 No. 6, pp. 607-616, June 1996.

[5] T.P. Minka and R.W. Picard, "Interactive Learning with a "Society of Models"", *Pattern Recognition*, vol.30, (no.4):565-81, April 1997.

[6] P.M. Murphy and D.W. Aha, UCI repository of machine learning databases. http://www.cs.uci.edu/ mlearn/MLRepository.html, 1995.

[7] X. Wu and B. Bhanu, "Gabor Wavelet Representation for 3-D Object Recognition," *IEEE Trans. on Image Processing*, 6(1), pp. 47-64, Jan. 1997.

Adaptive Target Recognition Using Reinforcement Learning

Bir Bhanu, Yingqiang Lin, Grinnell Jones, Jing Peng
Center for Research in Intelligent Systems
University of California, Riverside, California 92521
Email: bhanu@cris.ucr.edu
URL: http://www.cris.ucr.edu

Abstract

This paper focuses on using reinforcement learning to improve the performance of a SAR recognition engine. We develop a learning algorithm which can direct the SAR recognition engine to perform at or close to a particular user prespecified performance point (PCI_u,Pf_u) on the ROC curve of the engine. Experimental results with the current learning algorithm are presented using MSTAR public data.

1 Introduction

In this paper, we are concerned with using learning to improve the performance of a SAR recognition engine. The goal of our research is to integrate learning with the SAR engine and make the engine more reliable and robust to handle images acquired under dynamically changing conditions.The features used by the SAR engine for recognizing an object are the locations and magnitudes of the scattering centers. Our SAR engine is model based with the models built off-line. The most important SAR engine parameters are the number of scattering centers (N) and vote threshold (T). Given an input SAR image, the SAR engine can recognize the identity of the object or return "unknown" if the number of votes is less than T. After the SAR engine runs over the iuput images, we can use an evaluation program to calculate its performance (PCI,Pf) if we know the true object (Ground Truth) in each image. (For details, see [1], [2].) In order to get acceptable (PCI,Pf), the N and T must be chosen appropriately. If the input images change or the conditions under which the images are acquired change, N and T may also be changed in order to maintain acceptable (PCI,Pf).

We use a reinforcement learning algorithm to select good values of N and T. We embed the SAR engine and evaluation program into the learning system, as shown in Figure 1 and 2. The learning system selects the parameters N and T for the SAR engine and the evaluation program reports the performance (PCI,Pf) of the SAR engine to the learning system. The learning system calculates the reinforcement of the selected pair of parameters by comparing the (PCI, Pf) with the (PCI_u,Pf_u) and uses the reinforcement to change the Q-Value of the selected pair of parameters. After the learning stops, the Q-Value represents the goodness of the parameters. Good parameters can be chosen according to Q-Values.

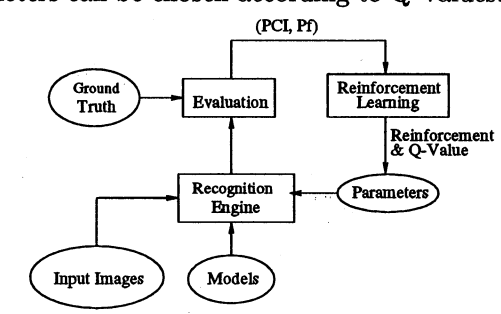


Figure 1: Learning integrated recognition

At the current stage, our learning algorithm works, but needs further improvement. Section 2 gives the learning algorithm. Experimental results are shown in section 3. Section 4 provides the conclusions and some suggestions about future work.

2 Learning Algorithm

Figure 2 shows the flowchart of the learning algorithm. Further details are given in [3].

Initial Conditions: When the algorithm starts to run, it asks the user to input the performance point (PCI_u, Pf_u). Each pair of parameters (N, T) of SAR engine is associated with a value Q(N,T), which measures the goodness of this pair of parameters. The larger the Q Value, the better the pair of

This work is supported in part by grants F49620-97-1-0184. The contents and information do not necessarily reflect the position or the policy of the U.S. Government.

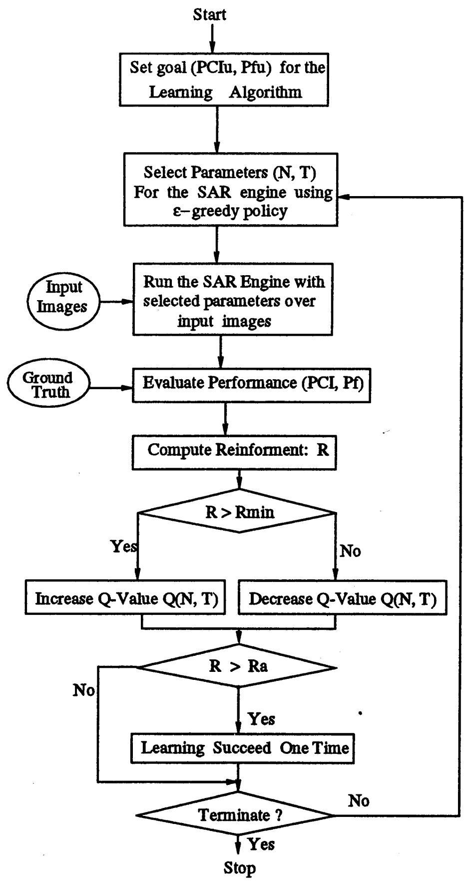


Figure 2: Flowchart of the Learning Algorithm

parameters. All the initial Q values are 0. All pairs of parameters (N, T) are marked SELECTABLE.

Select Parameters for the SAR Engine: Only those pairs of parameters marked SELECTABLE and with nonnegative Q Value can be selected by the learning algorithm. If no pair of parameters meets this requirement, the learning algorithm stops automatically. The parameter is selected using an ϵ-greedy policy.

1. Select the pair of parameters (N,T) whose corresponding Q Value is the maximum among all the Q-Values with probability 1 - ϵ. If several pairs of parameters have the maximum Q-Value, choose one of them randomly.
2. Select parameters (N, T) randomly with probability ϵ.

The value of ϵ determines the probability the learning algorithm searches the parameter space of the SAR engine. Parameter ϵ has an initial value ϵ_0 (we used 0.9), which means the algorithm searches the space with higher probability. Each time when the learning succeeds, the ϵ decreases by $\Delta\epsilon$ (we used 0.1). So, as the learning algorithm runs and the learning succeeds again and again, the probability of searching decreases and the learning algorithm gradually focuses on good parameters. The value of ϵ must be at least $\epsilon_{min} = 0.1$.

Compute Reinforcement: The learning algorithm compares the performance point (PCI, Pf) returned by the SAR engine with the performance point (PCI_u, Pf_u) pre-specified by the user and calculates the distance D between them. Then it computes the reinforcement R using

$$R = (1 - \frac{D}{\sqrt{2}}) \times S \qquad (1)$$

where S is a distance scale factor (we used S = 1.0).

Update Q-Value: If the reinforcement R $\geq R_{min}$ (we used $R_{min} = 0.9$), the algorithm increases Q(N, T) using the following formula:

$$Q(N,T) = Q(N,T) + \alpha \times [R - Q(N,T)] \qquad (2)$$

where α was set to 0.1. Then the algorithm compares the new Q(N,T) with the old Q(N,T). If the difference between them is smaller than Q_Δ (we used $Q_\Delta = 0.001$), the parameter (N,T) is marked UNSELECTABLE, and the ϵ is set to its initial value ϵ_0, which means that the learning algorithm should explore the parameter space more actively.

By mathematical induction, it is easy to prove that if the same parameter (N,T) is selected infinite times, its Q-Value Q(N,T) converges to R in equation (2).

If reinforcement R $\leq R_{min}$, the algorithm decreases the Q(N,T) using

$$\begin{aligned} Q(N,T) &= Q(N,T) + \alpha \times \\ &\quad [R - R_{min} - Q(N,T)] \end{aligned} \qquad (3)$$

Then the pair of parameters (N,T) is marked UNSELECTABLE, which means that the pair of parameters (N,T) is not good and will not be selected any more.

Learning Succeed One Time: If the learning succeeds, that is, the reinforcement received is greater than R_a (we used $R_a = 0.95$), the value of ϵ is decreased by $\Delta\epsilon$, which means the probability of searching is decreased.

Termination Condition: If the learning succeeds more than L times, the learning algorithm stops. Otherwise the learning algorithm tries another pair

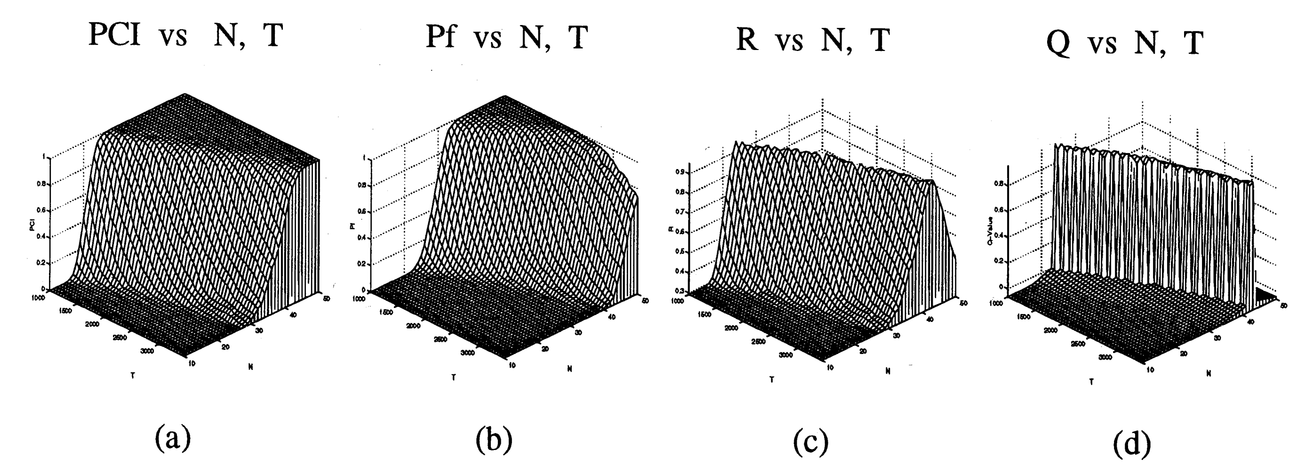


Figure 3 Adapting parameters for articulated objects

of parameters (N,T) and the loop runs again. It is worth noting that the algorithm will stop eventually (In the worst case after an exhaustive search). The searching strategy used in learning is critical to the efficiency of the learning algorithm. If the searching strategy is good, the learning algorithm can find the good parameters and converge quickly.

3 Experimental Results

We conducted three sets of experiments based on articulated targets, depression angle changes and target configuration variants. In all these three experiment, the test data (input images) and model are real SAR images from the MSTAR public data.

3.1 Articulated Object Results:

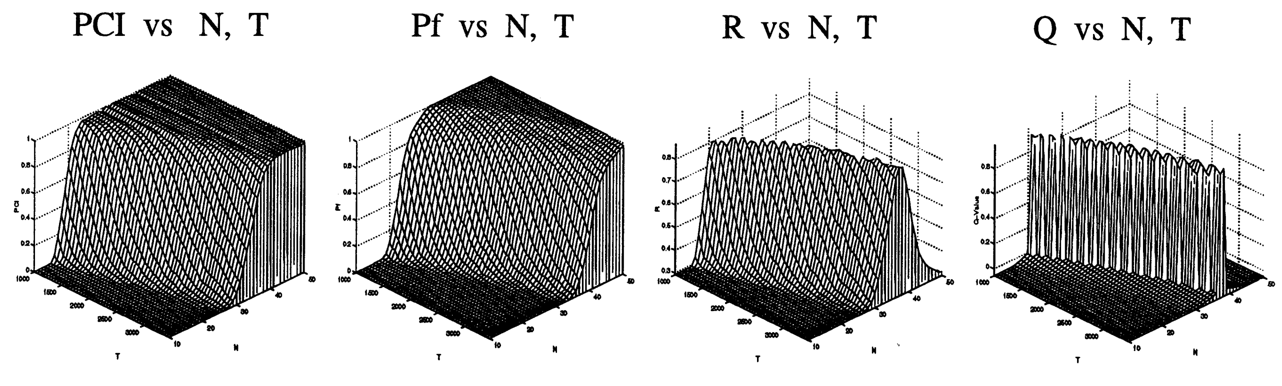
In this experiment, the input images (test data) are articulated versions of T72(#a64) and ZSU23/4(#d08) at 30° depression angle, using the non-articulated versions of these same serial number objects as the model. BRDM2(#e71) is also included in test data, used as "unknown" confuser vehicle. Figure 3(a) shows the affect of N and T on PCI, and Figure 3(b) shows the affect of N and T on Pf, where N ranges from 10 to 50 and T from 1000 to 3500. We can see that as the N increases, the PCI and Pf increase, as T increases, the PCI and Pf decrease. The best results are high PCI and low Pf value. Figure 3(c) shows the goodness of the parameters expressed as the reinforcement R obtained from each pair of parameters when user specified performance point is the best point (1.0, 0.0). The larger the reinforcement, the better the pair of parameters. The pair of parameters (34, 2050) are the best parameters. Figure 3(d) shows the Q-value of each pair of parameters after the learning algorithm stops. In this experiment, L is 1000 and the user specified performance point is (1.0, 0.0). The Q-Value measures the goodness of the pair of parameters. They are consistent with the reinforcement shown in Figure 3(c). Parameters (34,2050) has the maximum Q-Value.

3.2 Depression Angle Change Results:

The input images (test data) are versions of T72(#132) and BMP2(#C21) at 17° depression angle, using these same serial number objects at 15° as the model. BTR70(#C71) at 15° is also included in test data as "unknown" confuser. The results are shown in Figure 4. In this experiment, L is 2500 and the user specified performance point is (0.89, 0.16), which is the performance point when the best pair of parameters (36,3100) in Figure 4(c) is selected. Parameter (36,3100) has the maximum Q-Value in Figure 4(d).

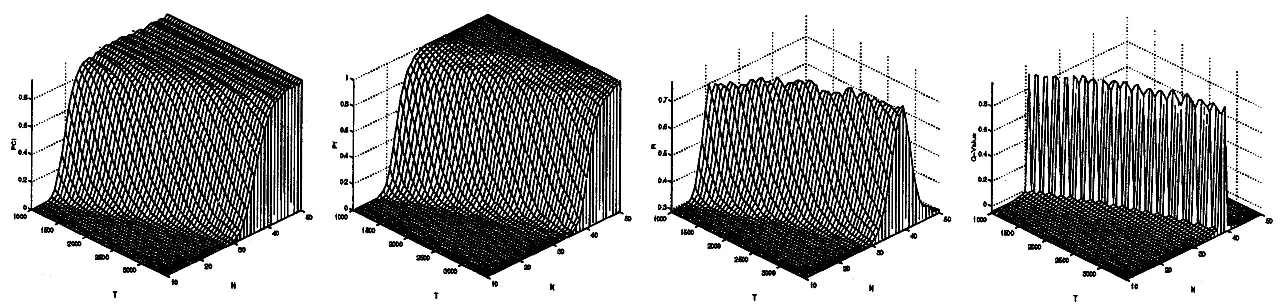
3.3 Configuration Variant Results:

In this experiment, a single configuration of the vehicle (BMP2#C21 and T72#132) is used as model and test data are two other variants of each vehicle type. BTR70(#C71) is also included in test data, used as "unknown" confuser. Both the model and data are acquired at 15° depression angle. The results are shown in Figure 5. In this experiment, L is 1500 and the user specified performance point is (0.72, 0.15), which is the performance point when the best pair of parameters (35,2750) in Figure 5(c) is selected. Parameter (35,2750) has the maximum Q-Value in Figure 5(d).



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

Figure 4 Adapting parameters for depression angle changes



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

Figure 5 Adapting parameters for configuration changes

4 Conclusion and Future Work

In this paper, we integrate the reinforcement learning and a SAR recognition engine, and the learning algorithm directs the SAR engine to perform at or around the performance point prespecified by the user. The learning algorithm improves the robustness and adaptability of the SAR engine. Our experiments show that the learning algorithm works, but needs further improvement so that it is time efficient. We need to develop a search strategy which searches only a small part of the parameter space to find the best parameters. Also, in order to compute the reinforcement, the learning algorithm requires that an evaluation program report the SAR engine performance, which means the ground truth of the input images must be known. If the ground truth is unavailable, how the learning algorithm can estimate the performance of the SAR engine and its integration with image segmentation [5] based on prior experience will be another area for future research.

References

[1] B. Bhanu and G. Jones. "Performance characterization of a model-based SAR target recognition system using invariants," In *Proc. SPIE: Algorithms for Synthetic Aperture Radar Imagery IV,* Vol, 3070, pp. 305 - 321, April 1997.

[2] G. Jones, B. Bhanu and J. Guo. "Goemetrical and magnitude invariants for recognition of articulated and non-standard objects in MSTAR images. In *Proc. DARPA Image Understanding Workshop*, Nov. 21-23 1998.

[3] J. Peng and B. Bhanu. "Delayed reinforcement learning for adaptive image segmentation and feature extraction." *IEEE Trans. Sys, Man, and Cyber,* 28, no. 3:482-488, Aug 1998.

[4] J. Peng and B. Bhanu. "Closed-loop object recognition using reinforcement learning." *IEEE Trans. Pattern Analysis and Machine Intelligence,* 20, no. 2:139-154, February 1998.

[5] B. Bhanu and J. Peng. "Adaptive integrated image segmentation and object recognition." In *Proc. Eleventh Vision Interface Conference,* pages 471-478, June 1998.

View Planning for Site Modeling

Peter K. Allen Michael K. Reed Ioannis Stamos
Department of Computer Science, Columbia University
{allen,m-reed,istamos}@cs.columbia.edu *

Abstract

3-D models of complex environments, known as site models, are used in many different applications ranging from city planning, urban design, fire and police planning, military applications, virtual reality modeling and others. Site models are typically created by hand in a painstaking and error prone process. This paper focuses on two important problems in site modeling. The first is how to create a geometric and topologically correct 3-D solid from noisy data. The second problem is how to plan the next view to alleviate occlusions, reduce data set sizes, and provide full coverage of the scene. To acquire accurate CAD models of the scene we are using an incremental volumetric method based on set intersection that can recover multiple objects in a scene and merge models from different views of the scene. These models can serve as input to a planner that can reduce the number of views needed to fully acquire a scene. The planner can incorporate different constraints including visibility, field-of-view and sensor placement constraints to find correct view points that will reduce the model's uncertainty. Results are presented for acquiring a geometric model of a simulated city scene and planning viewpoints for targets in a cluttered urban scene.

1 Introduction

Realistic 3-D computer models are fast becoming a staple of our everyday life. These models are found on TV, in the movies, video games, architectural and design programs and a host of other areas. One of the more challenging applications is in building geometrically accurate and photometrically correct 3-D models of complex outdoor urban environments. These environments are typified by large structures (i.e. buildings) that encompass a wide range of geometric shapes and a very large scope of photometric properties. 3-D models of such environments, known as site models, are used in many different applications ranging from city planning, urban design, fire and police planning, military applications, virtual reality modeling and others. This modeling is done primarily by hand, and owing to the complexity of these environments, is extremely painstaking. Researchers wanting to use these models have to either build their own limited, inaccurate models, or rely on expensive commercial databases that are themselves inaccurate and lacking in full feature functionality that high resolution modeling demands. For example, many of the urban models currently available are a mix of graphics and CAD primitives that visually may look correct, but upon further inspection are found to be geometrically and topologically lacking. Buildings may have unsupported structures, holes, dangling edges and faces, and other common problems associated with graphics vs. topologically correct CAD modeling. Further, photometric properties of the buildings are either missing entirely or are overlaid from a few aerial views that fail to see many surfaces and hence cannot add the appropriate texture and visual properties of the environment. Our goal is to have a mobile system that will autonomously move around a site and create an accurate and complete model of that environment with limited human interaction.

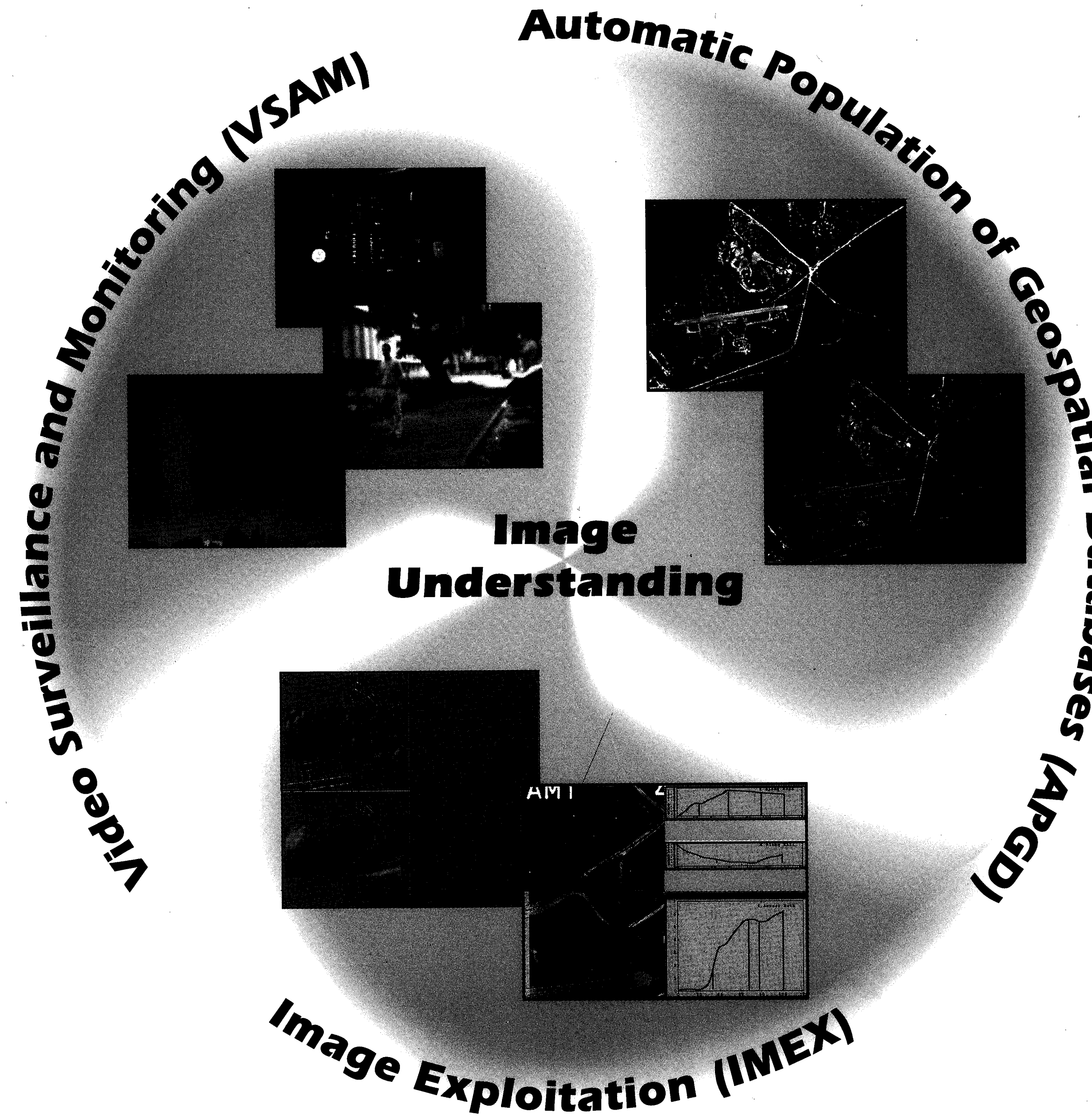
There are a number of fundamental scientific issues involved in automated site modeling. The first is

*This work was supported in part by an ONR/DARPA MURI award ONR N00014-95-1-0601, DARPA AASERT awards DAAHO4-93-G-0245 and DAAH04-95-1-0492, and NSF grants CDA-96-25374 and IRI-93-11877.

Proceedings Volume II

1998 Image Understanding Workshop

Hyatt Regency Monterey
November 20–23
Monterey, California



Hosted by SRI International

Edited by
George E. Lukes

Sponsored by
Defense Advanced Research Projects Agency
Information Systems Office