CyberScout: Distributed Agents for Autonomous Reconnaissance and Surveillance

Mahesh Saptharishi[2], Kiran S. Bhat[1], Christopher P. Diehl[2], John M. Dolan[1], and Pradeep K. Khosla[1,2]
[1] Robotics Institute,
[2] Electrical and Computer Engineering
Carnegie Mellon University

Abstract
The objective of the CyberScout project is to develop an autonomous surveillance and reconnaissance system. In this paper, we focus on advances in vision-based surveillance agents for detection, scene mosaicing, classification and correspondence. An agent-based software framework is used to promote synergy between the various surveillance algorithms and provide a distributed computing infrastructure for the system.

Keywords:
Visual surveillance, motion detection, image mosaicing, object classification and moving object correspondence.

1 Introduction

Camera-based surveillance has long been used for security and observation purposes. Surveillance cameras are typically fixed at known positions and have coverage of a circumscribed area defined by the fields of view of the cameras. Although some recent vision work has addressed autonomous surveillance, in most cases humans perform the sensory processing, either in real time, or by reviewing footage. Likewise, humans have performed reconnaissance, or scouting, for centuries in military and other applications in order to determine the "lay of the land" and identify and classify activities in the environment. We combine the sensory capabilities of surveillance with the mobility of reconnaissance by mounting cameras on mobile robotic platforms. The resulting groups of collaborating reconnaissance and surveillance robots pose interesting challenges in vision-based surveillance algorithms, multi-agent software architectures, and mission management [1].

2 The CyberScout distributed surveillance and reconnaissance system

We have created a group of two mobile and four stationary sentries capable of cooperating for reconnaissance and surveillance. In addition, we have developed a distributed agent-based software framework, called CyberARIES, capable of efficiently running various vision, planning and control algorithms.

2.1 The CyberScout robotic sentries

The mobile sentries are retrofitted Polaris All-Terrain Vehicles (ATVs) (Figure 1) with automated throttle, steering, gearing, and braking and computation for control, navigation, perception, and communication [2]. The ATV computing architecture is two-tiered: a PC/104 controls (low-level) vehicle locomotion, while a group of three PCs perform (high-level) perception, planning, and communications. Navigational sensing is performed by a 20-cm resolution NovAtel differential GPS unit. Each vehicle is equipped with five cameras, a panning stereo pair in front for obstacle avoidance and mapping, and three pan/tilt cameras for surveillance, one each located at the front left, front right, and rear. Each stationary sentry is a PC with a camera on a tripod. All sentries are able to communicate with one another via WaveLAN wireless Ethernet, and all run the same perception algorithms for performing surveillance

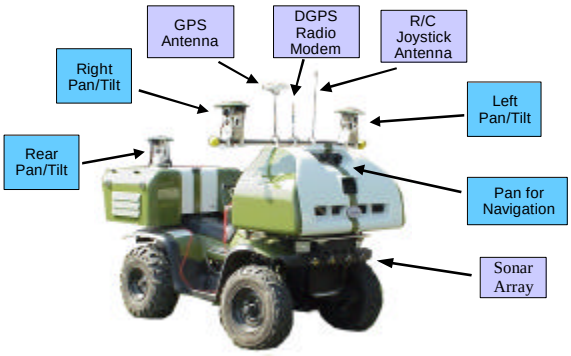

Fig 1. One of two All-Terrain Vehicles (Lewis and Clark) retrofitted for perception and navigation

2.2 A distributed agent-based software infrastructure

CyberARIES (Autonomous Reconnaissance and Intelligence Exploration System) abstracts the low-level system resources and promotes a modular architecture for the system. The fundamental building block within CyberARIES is an “agent”. The agent is an algorithm contained within a shell that provides access to necessary resources through a simple abstract interface. The agent can be viewed as an entity that requests and/or provides services to other agents in the system. Figure 2 shows the connectivity of the agent shell. The *agent runloop* holds the

algorithm. For example, in the case of the classifier, the classification algorithms are contained within the *agent runloop*. The input to the classification algorithm is accepted from other agents via the *sink*. The output of the algorithms is then distributed to other agents via the *source*. The *resource management* block manages the computational, hardware and software resources for the agent. The *distribution agent* abstracts network communication between agents from the algorithms contained within the *runloop*. The *distribution agent* is also a CyberARIES agent and one instance of this agent exists on each computation platform in the network.

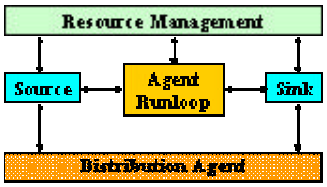

Fig 2. An agent consists of an algorithm contained within the *agent runloop* and wrapped with a *source, sink, distribution agent* and a *resource management* shell.

3 Visual Surveillance

Moving objects in the scene are the subjects of interest in most visual surveillance applications. Specifically, the actions of the moving objects need to be monitored and appropriate flags need to be raised when events of interest occur in the scene. Additionally, the mobile mechatronic platform needs to be controlled so as to aid the surveillance task. The objectives of the surveillance task and the associated real-time computation constraints involve difficult challenges in building highly robust yet computationally cheap algorithms for detection, classification and correspondence.

3.1 Motion Detection

Most surveillance systems described in the literature [3,4,5,6] have used background subtraction as an efficient means of motion detection with a stationary camera. Unfortunately, background subtraction techniques are not always robust under camera jitter, varying lighting conditions, and moving foliage. Problems of a periodic nature such as jitter and moving foliage induce multi-modal distributions of pixel intensity values. We propose a simple technique that generates a multi-modal background model of the scene. The multi-modal background can then be used for motion detection and segmentation via background subtraction.

One popular method for background model generation is the use of AR (auto-regressive) or IIR (infinite impulse response) filters [4,7]. A single AR or IIR filter is used for each pixel to estimate the dominant mode of the background. Our

technique extends this method by allowing for the estimation of all the modes of the background. This is accomplished by appropriately adding an AR filter for each mode. The system is initialized with a single AR filter for each pixel. The AR filter estimates the center and width of the dominant mode of the background. Associated with each filter per pixel is a value that approximates the probability that the mode represented by the filter is seen by the pixel. When an intensity value seen by the pixel falls within the mode of one of the pixel's filters and the associated probability is greater than a preset threshold, then the pixel is declared as background. When a particular intensity value is not represented by any of the filters for that pixel, a new filter is added with an associated low probability. When a filter represents an intensity value, its probability is increased and the probabilities of the rest of the filters associated with that pixel are decreased. At each detection cycle the filters corresponding to a pixel adapt to better fit the modes of the background. In practice, we have found that no more than four filters are required for a robust background model.

This motion detection technique is enhanced by the use of feedback from higher-level agents such as the classifier and correspondence agents. The classifier, described in section 3.3, has the capability to reject spurious detections. This information can be used either to create a new filter for the pixel or to increase or decrease the probability threshold for detection. Information from the correspondence agent can be used to predict future locations of detections. This considerably improves the quality of segmentation.

3.2 Image Mosaicing

In order to perform motion detection while the camera is panning, we have developed an efficient, pseudo-real-time algorithm for constructing an image mosaic from a sequence of background images. The detection algorithm described above continuously updates the background represented by the viewable subset of the image mosaic. Several techniques have been proposed to create an image mosaic from sequences of images [8,9,10]. They obtain the registration between images by minimizing the sum-squared error of image intensities at each pixel. Although these techniques produce very accurate registration results, they tend to be slow, and typically require user interaction to initialize the registration. We create an image mosaic in pseudo-real time by locating and tracking feature points [11] in the image sequence. This technique is much faster than previous techniques, and does not require any user intervention. We also propose a method to accurately index the viewable portion of the image mosaic corresponding to a particular camera rotation [7]. An example of the motion detection and segmentation with an image mosaic is shown in Figure 3.

In addition, we have developed a technique to register images obtained from multiple centers of projection. The algorithm uses color and range information (from stereo), obtained from various camera locations, and registers them onto a

single mosaic plane. Figure 4 shows an example of a 3D Mosaic created from two images obtained at different centers of projections.

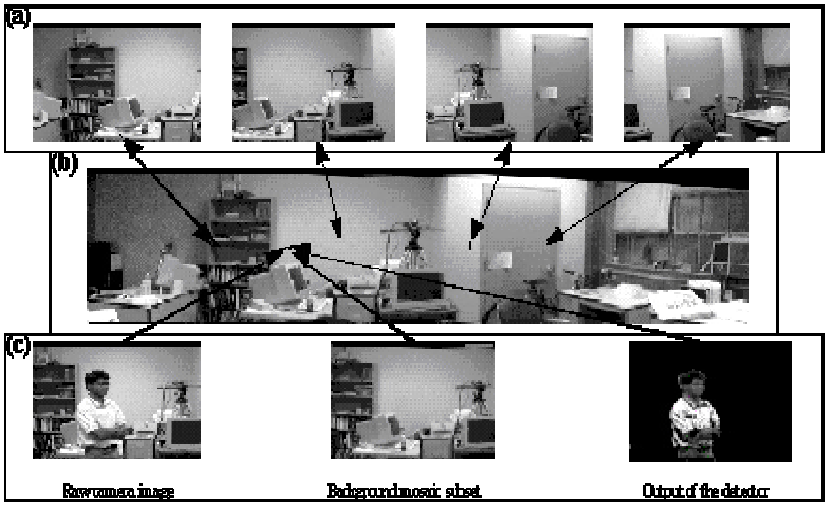

Fig 3. Results of detection with image mosaics. (b) shows the mosaic constructed using a spatial sequence of 70 images. 4 images from this sequence are shown in (a). (c) shows a moving person at a particular camera rotation, the corresponding subset indexed from the background mosaic, and the extracted foreground object. The indexing and detection algorithms execute at 10 Hz on a Pentium 266MHz laptop.

Fig 4. Results of 3D Mosaicing. The bottom row shows a 3D Mosaic created by registering two images shown in the top row. Note that there is significant translation and rotation between these two input images. The center of projection of the resulting mosaic coincides with the center of projection of the first image. The algorithm uses color & range data obtained from a stereo head to perform registration.

3.3 Classification

The classification procedure maps a given image sequence of a moving object to the most likely class label by classifying each image independently and then classifying the resulting set of class labels. The image classifier is a single-layer perceptron trained with differential learning [12]. The novel training method used enhances the translation invariance and rejection capability of the classifier [3]. The current classifier operating within the CyberScout system classifies 20x20 pixel binary image sequences into one of three classes in real-time: person, people or vehicle. In addition, the classifier can also reject classifying the object. The classifier has demonstrated class-conditional sequence error rates of less than 5% and a false alarm rejection rate of 80% in disjoint tests. Unknown objects and novel views of known objects are detected by considering the class label history over the image sequence. Image sequences that yield a significant fraction of rejections or cause atypical classifier confusion are saved for user interpretation. We have also implemented finer-grain classification, such as deciding if a person is wearing a backpack, as part of a hierarchical classification scheme [1]. Figures 5 and 6 show the system in action.

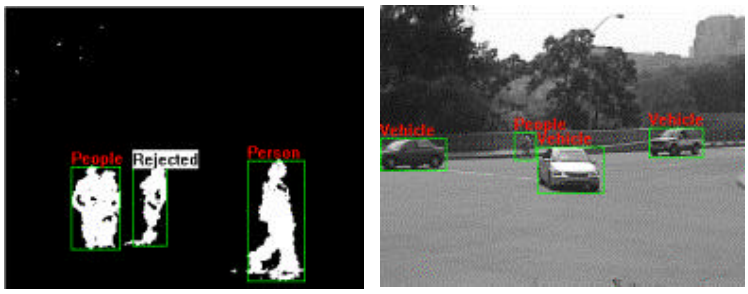

Fig 5: (left) Binary motion images generated by the motion detector with the classified motion regions delineated, (right) Original image with the classified motion regions

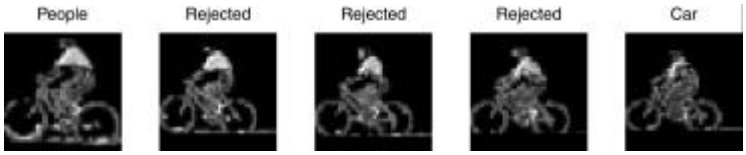

Fig 6: Example of a novel image sequence producing significant classifier confusion

3.4 Correspondence

Temporal correspondence of a moving object plays a very important role in robustly classifying, tracking and interpreting an object's actions. Large numbers of targets of varying sizes preclude the use of simple positional correspondence, i.e., correspondence based purely on the positions of moving objects. In such situations, other features of the moving objects, such as different appearance traits, need to be considered for robust correspondence. How can we select appearance features so as to facilitate good correspondence? The measure of goodness of the features we choose not only depends on the object in question, but also on other objects in the scene. A globally "good" set of features can be estimated *a priori*, but only a subset of these features might be relevant to the correspondence of a particular object. We pose the estimation of the relevance of globally good features for corresponding a particular object as an on-line learning task. We have developed a technique called *differential discriminative diagnosis* [13] to provide a systematic method for estimating the relevance of features and checking the temporal consistency of these features for a particular object.

The correspondence task is performed by a two-step procedure. The first step relies on positional correspondence using linear prediction. This step nominates a set of likely candidate matches for a reference object. The second step uses appearance-based correspondence [14] to find the best match (if one exists) among those nominated by linear prediction. A simple linear classifier is trained to decide whether or not two images are of the same object. Differential discriminative diagnosis identifies those features that are most relevant to the correspondence task. Efficient correspondence is achieved by enforcing the temporal consistency of the relevances for a particular object. This technique corresponds moving objects with an accuracy of 96%.

4 Conclusions

Autonomous reconnaissance and surveillance performed by mobile robots presents challenges in the areas of vehicle control, vision algorithms, and mission management. The surveillance algorithms described here represent advances in the areas of autonomous vision-based detection, classification, and correspondence. Multi-modal background model-based detection in conjunction with mosaicing increases the robustness and speed of target acquisition. Classification with reliable rejection allows the recognition of novel classes, possibly leading to autonomous extension of the set of identifiable classes. The combination of motion-based prediction with appearance-based classification of auto-selected key features leads to highly accurate target correspondence. Current and future work will concentrate on using surveillance information to dynamically adjust mission tasking.

Acknowledgment
We gratefully acknowledge the support of DARPA contract F04701-97-C-0022, "An Autonomous, Distributed Tactical Surveillance System", in performing this work.

References
1. M. Saptharishi, K. Bhat, C. P. Diehl, C. S. Oliver, M. Savvides, A. M. Soto, J. M. Dolan, P. Khosla: Recent Advances in Distributed Collaborative Surveillance, Aerosense 2000, Orlando, April 2000, pub. SPIE. (To be published)
2. J.M. Dolan, A. Trebi-Ollennu, A. Soto, P. Khosla: Distributed Tactical Surveillance with ATVs, Proceedings on Unmanned Ground Vehicle Technology, Orlando, April 1999, pub. SPIE, vol. 3693, pp. 192-199.
3. C. P. Diehl, M. Saptharishi, J. B. Hampshire II, P. K. Khosla: Collaborative Surveillance Using Both Fixed and Mobile Unattended Ground Sensors, Aerosense ’99, Orlando, April 1999, pub. SPIE, vol. 3713, pp. 178-185.
4. A. J. Lipton, H. Fujiyoshi, R. S. Patil: Moving target classification and tracking from real time video, Workshop on Application of Computer Vision, 1998, pub. IEEE, pp. 8-14.
5. W. E. L. Grimson, L. Lee, R. Romano, C. Stauffer: Using adaptive tracking to classify and monitor activities in a site, CVPR ’98, 1998, pub. IEEE, pp. 22-31.
6. I. Haritaoglu, D. Harwood, L. Davis: W4: Who? When? Where? What? A real time system for detecting and tracking people, International Conference on Automatic Face and Gesture Recognition, 1999, pub. IEEE, pp. 222-227.
7. K. Bhat, M. Saptharishi, P. Khosla: Motion Detection and Segmentation Using Image Mosaics, to be presented at IEEE International Conference on Multimedia and Expo 2000 (ICME), Aug 2000.
8. R.Szeliski, H.Shum: Creating full view panoramic image mosaics and environment maps, Computer Graphics Proceedings, Annual Conference Series, 1997, pp. 251-258.
9. M. Irani, P. Anandan, J. Bergen, R. Kumar, S. Hsu: Mosaic representations of video sequences and their applications, Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, Analysis, and Application, May 1996, vol. 8, no. 4, pp. 327-251.
10. F. Dufaux, F. Moscheni: Background Mosaicing for low bit rate video coding, ICIP '96, September 1996, pub. IEEE, pp. 673-676.
11. J. Shi, C. Tomasi: Good Features to Track, CVPR ’94, June 1994, pub. IEEE, pp. 593-600.
12. J. B. Hampshire II: A Differential Theory of Learning for Efficient Statistical Pattern Recognition, Ph.D. Thesis, 1993, Carnegie Mellon University.
13. M. Saptharishi: Assessing Feature Relevance On-line Using Differential Discriminative Diagnosis, M.S. Thesis, December 1999, Carnegie Mellon University.

14. M. Saptharishi, J. B. Hampshire II, P. K. Khosla: Agent Based Moving Object Correspondence Using Differential Discriminative Diagnosis, CVPR 2000, June 2000, pub. IEEE, to be published.