CYBER-ATVs: DYNAMIC AND DISTRIBUTED RECONNAISSANCE AND SURVEILLLANCE USING ALL TERRAIN UGVs

Alvaro Soto[1]**, *Mahesh Saptharishi,*[3] *Ashitey Trebi Ollennu*[2]*, John Dolan*[1,2]*, and Pradeep Khosla*[1,2,3]

(1) Robotics Institute, Carnegie Mellon University
(2) Institute for Complex Engineered Systems, Carnegie Mellon University
(3) Electrical and Computer Engineering, Carnegie Mellon University

Abstract: This paper describes our current effort to develop robotic vehicles for tactical distributed surveillance. Our research is focused on multi-agent collaboration, reconfigurable systems, efficient perception and sensor fusion, distributed command and control, and task decomposition. In particular, this paper describes the main features and capabilities of our All Terrain Vehicles (ATVs), concentrating on their autonomous navigation capabilities.

Key Words: mobile robotics, distributed surveillance, visual perception.

1. INTRODUCTION

Reconnaissance and surveillance are tasks that generally require the simultaneous coverage of large areas. A system based on mobile robotic technologies, in conjunction with a distributed and collaborative multi-agent architecture, seems to be a suitable candidate to perform this task. In the Cyberscout project at Carnegie Mellon University we are creating such a system. The focus of the project is to develop mobile robotic technologies and unattended ground sensors that will extend the sphere of awareness and mobility to perform security operations.

In order to increase mobility and sensory coverage, we have developed several robotic platforms with different sizes and capabilities ranging from Microrobots (<5x5x5 cm^3) to All Terrain Vehicles (CyberATVs). The central idea is that a user can task this heterogeneous group of platforms as a single logical entity. Users task, control, and monitor the different platforms through CyberRAVE (Real and Virtual Environment), a software framework and graphical user interface which enables rapid configuration and prototyping of cooperating groups of real and virtual robots. This paper describes our work on robotics All Terrain Vehicles or CyberATVs. Further details about CyberRAVE and the overall architecture can be found in [1].

An ATV is a particularly suitable platform for augmenting mobility. It can ride over small obstacles, and also carry payloads such as powerful computing equipment, power units, and even a person. In the Cyberscout project, we have retrofitted two Polaris ATVs, automating their throttle, steering, braking, and gearing functions; and giving them computation for control, navigation, sensing and communications.

In this paper we describe the hardware and software architecture of the ATVs, their current capabilities, and future research. To avoid overlap with previous publications of our research group, we concentrate in this paper on the current autonomous navigation capabilities of the ATVs platform. Section 2 describes the hardware configuration of the system, while section 3 describes the software architecture. Section 4 describes the different operating modes of the ATVs. Section 5 describes the current autonomous navigation capabilities. Finally, section 6 presents conclusions and future lines of research.

2. HARDWARE

Figure 1 shows a picture of one of the ATVs and its main hardware components.

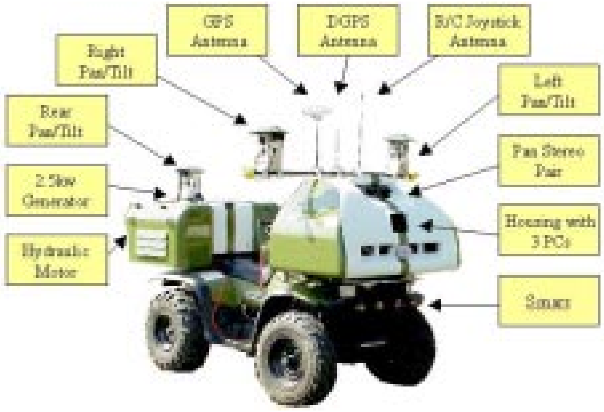

Fig. 1. Hardware components of the ATVs

2.1. Actuation

In order to provide each ATV with autonomous computer control, mechanical actuators for throttle, steering, gearing, and braking were added.

---

* *Send correspondence to Alvaro Soto at amsoto@ri.cmu.edu*

The throttle function is actuated by a R/C servo motor connected via a cable to the throttle plate. Steering, gear and brakes are actuated by linear hydraulic cylinders driven by a common hydraulic pump connected to an electric motor. The use of an hydraulic system was motivated by the large torque required to turn the steering at low speeds, and the space needed to install an electric motor with the necessary power. Actuation feedback is given by a linear potentiometer connected to the steering system, a tachometer mounted in the gearbox, and a gear selection indicator. Further information about the actuation system can be found in [2][3].

2.2. Computer System

A PC/104 unit based on a 586-133MHz processor performs low-level control of all the locomotion capabilities. Through optoisolated analog/digital I/O channels, this processor is able to receive each of the feedback signals and to control all the actuators.

The high-level planing and perception is performed by 3 motherboards based on PentiumII-350MHz processors. These motherboards are mounted on a custom-built housing, and connected in a start configuration via Fast Ethernet (100Mbps). General-purpose motherboards were preferred instead of a parallel computer or DSP due to the higher performance/cost ratio and easier programming.

2.3 Sensors

At this moment the ATVs are equipped with 3 types of sensors: GPS, video cameras and sonars. A magnetic compass will be included in the near future.

The GPS unit uses differential corrections to achieve a resolution of 20 cm. This differential signal is transmitted to the ATVs from a base station located on campus.

The visual perception system consists of 5 color cameras: a panning stereo pair located at the front center, and three pan/tilt color cameras, one each located at the front left, front right, and rear. Although the stereo pair is mainly used for obstacle avoidance and mapping, and the pan/tilt cameras for surveillance, the architecture is totally flexible. A task located on any of the processors can request access to any of the cameras or pan/tilt systems. This is achieved using a custom-built connection box.

The sonar system consists of 4 sonars located on the front of the vehicle. These sonars are mainly used as a safety system to detect any close obstacle that could be overlooked by the vision system.

2.4. Communications

Communications between the low-level and high-level processors are performed via a serial connection. Communications with other platforms and with remote users are performed via wireless Ethernet using 915MHz Wavelan technology. A 4-port hub allows internal communications among the on-board high-level processors.

2.5. Power System

The actual maximum power requirement of each ATV is approximately 2kw. Each ATV is accordingly equipped with a 2.5 kW electric generator. The generator provides power for the hydraulics, high-level processors, and sensing equipment. 5 volts are supplied separately to the PC/104 by the ATV's battery via a 12-to-5V DC converter. Using this configuration each ATV is able to run for a period of 3 to 4 hours, which can be easily extended adding an extra tank of gas to the generator.

3. SOFTWARE

The software of the ATVs is based on a distributed multi-agent architecture. Agents are independent processes that run concurrently and can be started on an as-needed basis. The architecture is totally decentralized, so that the interaction between agents emerges from the environment or mutual constraints. This scheme allows for an easy scalability of the system through the incorporation of new agents to provide new services. Figure 2 shows the main components of the software architecture of the ATVs. In the figure, each of the modules corresponds to a collection of agents.

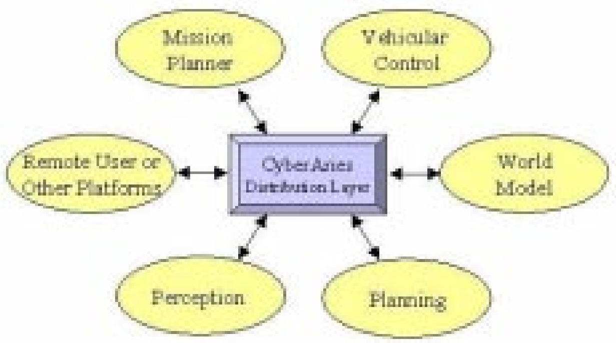

Fig. 2. Software components of the ATVs

The core of the software architecture is called CyberAries (Autonomous Reconnaissance and Intelligent Exploration System) [4]. CyberAries provides powerful inter-agent communication capabilities that greatly simplify the job of developing ensembles of cooperating agents. It mainly consists of an agent-framework environment and a distribution layer. The agent-framework provides all the operating system provisions such as concurrent processing or automatic scheduling, and the application abstraction provisions such as memory management and resource categorization. The distribution layer is responsible for providing and balancing the communications, processing, and sensing resources among the active agents. If an agent needs to send a message or to access any resource, the distribution layer handles all the details of the connection, checking for availability and resource allocation.

4. OPERATION MODES

Figure 3 shows the current modes of operation of the CyberATVs. Dark and light squares differentiate between modes that we have completed, and modes that are still in progress. Most of these modes have been already described in previous publications [1][2][3][4]. Here we concentrate on our current progress in autonomous navigation, giving a brief description of the rest of the modes.

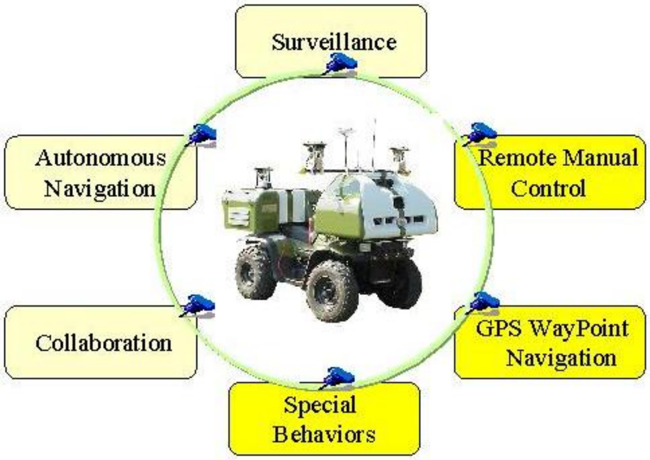

Fig. 3. Main modes of operation of the ATVs

In the remote control operation mode, a user can manually drive the ATV via a wired or R/C joystick, or using a laptop or wearable computer via wireless Ethernet.

In the GPS-based waypoint navigation mode, the ATV stores GPS coordinates while it is driven manually through a desired trajectory. Using these coordinates the ATV is able to repeat the path or to communicate the trajectory to the second vehicle.

In the surveillance operation mode, we have developed robust algorithms for personnel and vehicle detection, classification, and tracking. Although the current algorithms only work when the camera is stationary, we are currently working on versions that will work when the vehicle is moving. For the case of building surveillance, we have developed algorithms for window detection and classification.

To facilitate navigation tasks, two special behaviors have been added to the ATVs: autonomous convoying based on visual servoing, and sidewalk following based on Hue-Saturation and Intensity (HSI) chromatic analysis.

The development of the collaboration mode is still in an early stage. So far, we have implemented two cooperative behaviors: communication of waypoints and autonomous convoying, both mentioned above. In the area of cooperative surveillance, our current effort is the development of target hand-off algorithms for target tracking. In the area of cooperative navigation we are developing algorithms for the autonomous creation of world maps.

5. AUTONOMOUS NAVIGATION

At present we have implemented two modes of operation with autonomous navigation: the GPS waypoint navigation described above, and a Safe Wandering mode. In Safe Wandering mode the ATV periodically choses a random close GPS position as a new goal, and it tries to reach it. This section describes the methods used to achieve these autonomous navigation modes of operation.

5.1. Obstacle Detection

We are currently using vision as our main sensor modality for obstacle detection. Its passive and unobtrusive nature, notable progress in the area, plus the evidence of powerful biological systems indicate that this is a viable alternative. Our approach to visual perception is an adaptive integration of multiple visual cues.

In general a dynamic unconstrained environment allows for many interpretations. Patterns, tendencies and models lay in a complex high dimensional space of shapes, colors, sounds, past experiences, knowledge, and so on. In general, the mapping from single sensor data to perception is many to one. In order to deal with this ambiguity we need to integrate information from different sources or sensing dimensions.

In contrast to other sensor modalities, vision can allow the perception of a large number of different features of the environment such as color, shape, depth, motion, and so on. Depending on the task and the environment, the quantity of information or entropy in each visual cue can fluctuate. For instance, while stereo vision is usually a strong depth cue, in the case of images from a homogeneous grass field, the stereo decoding of depth is highly noisy. Even worse, it usually gives wrong information due to bad matches. Instead, in this case a visual cue such as texture or color conveys higher entropy.

An efficient adaptive integration of visual cues should consider the task and environment constraints to reduce ambiguity. Using these ideas, we have implemented obstacle detection algorithms based on two visual cues: color and stereo disparities.

5.1.1. Color Based Obstacle Detection

We have been using color segmentation for the detection of areas free of obstacles in structured environments. The basic idea is that ground-planes of structured environments, such as grass field or roads, usually present a homogeneous intensity or color, which can be easily identified through color segmentation techniques [5].

Our color segmentation technique is based on a HSI model. The main advantage of HSI over RGB-based techniques is that HSI allows a nice decoupling of the color and intensity information. While the Hue component conveys the color information, the intensity component contains information of brightness, and the saturation indicates how pure the color is.

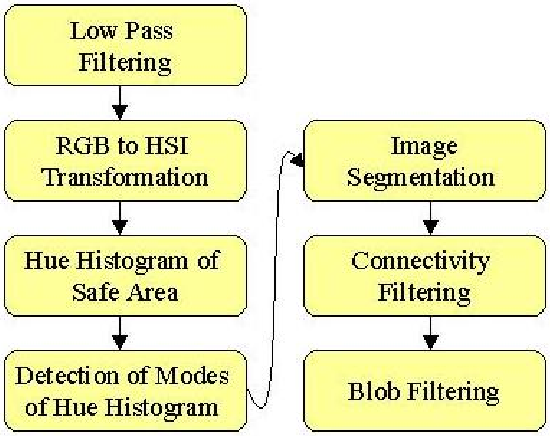

Fig. 4. Diagram of the obstacle detection using HSI color segmentation.

Figure 4 shows a diagram of our obstacle detection algorithm based on HSI color segmentation. The first step is to reduce the image noise using a 3x3 neighborhood averaging low pass filter. Then, the RGB input image is converted to HSI color components using a heuristic expression. Intensity is calculated as the average of the RGB components; saturation is given by the difference between the maximum and minimum value of the RGB components; and Hue is calculated according to the relative influence of the dominant RGB component with respect to the other 2 components. This is achieved using a circular transformation, where the RGB components are equally spaced on the circle. Red at 0°, Green at 120°, and Blue at 240°. The hue value for an input point with dominant color Green is given by:

$$Hue(R,G,B) = 120 + \frac{B-R}{\max(R,G,B)-\min(R,G,B)}$$

We prefer this heuristic transformation over the traditional RGB-HSI transformation based on a color pyramid [6], because it produces a smoother mapping.

In the next step, we assume that the floor area directly in front of the robot, corresponding to the bottom of the input image, is ground-plane free of obstacles. We use the image points in this safe ground area to build a histogram of the distribution of points with valid hue information. We consider as points with valid hue information, all the points with a saturation level over a threshold. If the safe ground area possesses a homogenous or texture pattern with enough points with valid hue information, the resulting histogram is multimodal. A simple analysis of the peaks and valley of this histogram allows setting adequate thresholds to distinguish between obstacles and floor points.

The rest of the algorithm uses these thresholds to segment the image points between obstacle and floor candidates, and then a connectivity and blob filtering analysis produces the final obstacle detection.

Figure 5 shows the result of the obstacle detection using the HSI-based segmentation on a grass field. The trapezoid in the left figure corresponds to the safe area used to build the hue histogram. The trapezoid in the right figure corresponds to the area used to detect obstacles. In general the segmentation for a grass field is very accurate. Using a flat-plane assumption and information about the tilt angle of the camera, it is possible to estimate the distance from the obstacle to the vehicle.

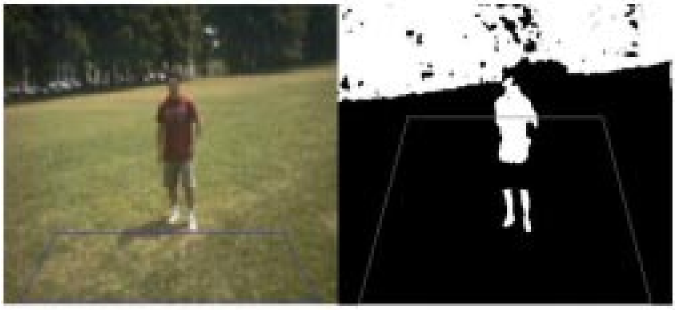

Fig. 5. Example of the results obtained with HSI-based obstacles detection

There are two important observations about the images. First, note the correct detection of the shadow of the person as not an obstacle. Although this shadow produces a large variation in intensity, the decoupling of intensity and color allows keeping in the Hue component only the color information. Also note how the green pants of the person are incorrectly detected as not obstacle. This is a limitation of the algorithm, but fortunately in our grass field the probability of finding a green person or a completed green obstacle is extremely low. Even it this happens, still the obstacle can be detected by the sonar backup system.

This last observation shows one of our design concepts. While we are trying to provide each ATV with a high degree of autonomy, we are also trying to keep the complexity of the system low, especially in terms of processing power, even if that makes the occasional intervention by a remote user necessary.

5.1.2. Stereo Vision

The principle used by stereo vision to provide range information is based on the comparison of the projection of world points on two or more images. The key points to obtain an accurate stereo map are the knowledge of the relative position of the cameras (calibration problem) and the position of the projections of world points to different images (correspondence problem). In our system the calibration problem is solved using a 3-D cube of known dimension and position with respect to the stereo pair. Then the calibration method proposed by Robert [7] is used to obtain the intrinsic and extrinsic camera parameters.

In order to facilitate the stereo correlation, two pre-processing steps are applied to the input images. First, the images are rectified in order to align the epipolar lines with the horizontal scan lines of the video cameras. Second, the images are convolved with a Laplacian of a Gaussian (LoG) filter. This filter not only allows eliminating high frequency noise and intensity asymmetries between the images, but it also enhances the image textures.

In the case of the correspondence problem, we tried the sum of squared differences (SSD) and the normalized correlation methods (NCM). Although SSD was slightly faster, we achieved better results using NCM. In this way, for each image window *Wnd*, centered at a point *(xc,yc)* on the left image, we calculate the best disparity *d(xc,yc)* maximizing over *d* the following expression:

$$\frac{n*\sum_{x,y\in Wnd} Right\,\mathrm{Im}(x+d,y)*Left\,\mathrm{Im}(x,y)-\sum_{x,y\in Wnd} Right\,\mathrm{Im}(x+d,y)\sum_{x,y\in Wnd} Left\,\mathrm{Im}(x,y)}{n^2*\sigma_{right}\sigma_{left}}$$

Where σ_{right} and σ_{left} correspond to the standard deviation of the intensity values in the correlation window *Wnd*, and *n* is the number of pixels in the window.

Due to lack of texture, differences in foreshortening, occlusions, or repeated patterns in the images, the resulting disparity map after the correlation was usually noisy. Two types of post-filtering schemes were applied to improve the results. Here, more than to produce a clean depth map, the focus was the detection of close large obstacles.

The goal of the first set of filters was to eliminate outliers due to the problems mentioned before. A first filter eliminates areas with low texture using a threshold on the minimum value of the standard deviation in the correlation window. A second filter attempts to eliminate areas with differences in foreshortening or occlusion through the use of a threshold on the minimum correlation value. Finally, a third filtering step eliminates areas with repeated patterns using a threshold on the relative difference and separation between the two greatest peaks of the correlation function.

The goal of the second set of filters was the detection of large obstacles. First the disparity map is converted to 3D world coordinates. Then, using thresholds on the maximum and minimum depth and elevation, all the points that far from the robot and at the floor level are eliminated. Finally, a region-growing algorithm, based on 4-connectivity of the disparity map, reports as obstacles all the connected smooth regions over a minimum size. Figure 6 shows a diagram of all the stereo process.

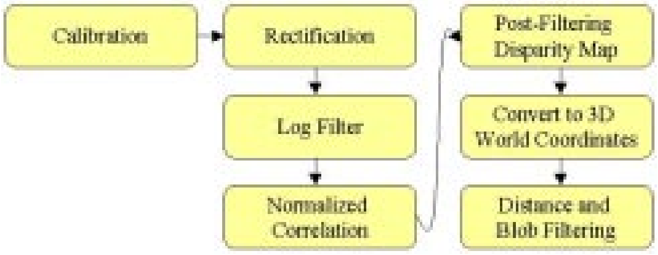

Fig. 6. Diagram of the obstacle detection using stereo vision.

Figure 7 shows the result of the obstacle detection using stereo vision. The input image is shown in the upper left. The resulting disparity map is shown in the upper right. Note how the repeated patterns in the background building and in the grass produce a very noisy disparity map. The image in the lower left shows the final obstacle detection after the post-filtering steps. Note how the cleaning and region growing filters effectively eliminate most of the noise. Here again we use environment constraints to search just for the kind of obstacles that we expect to find. Finally, the image in the lower right shows an occupancy grid map built using the range information from the stereo system. In this map the diagonal lines indicates the current field of view of the robot corresponding to the Bright World mentioned in [8].

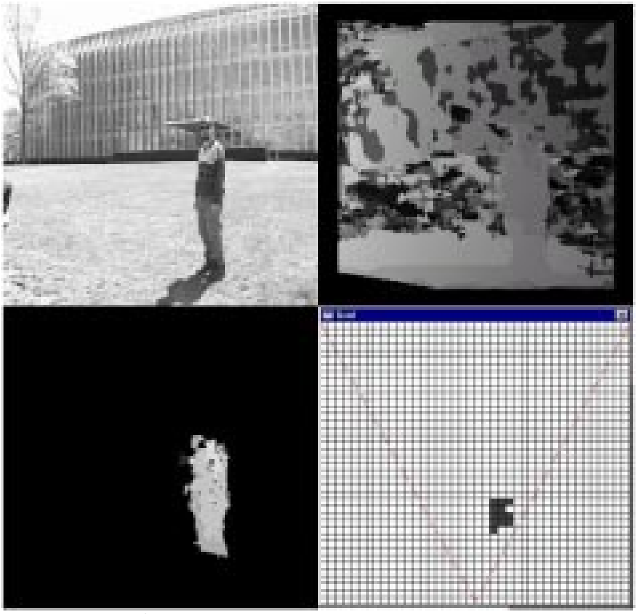

Fig.7. Example of the results obtained with stereo based obstacle detection

5.2. Obstacle Avoidance

Using the output from the obstacle detectors, we have developed a potential field-based method for obstacle avoidance. This method is applied to the part of the world that is currently in the field of view of the robot. In order to apply this technique we project all the obstacles to a flat plane parallel to the ground, and we merge all the obstacles in view using their convex hull. Also we use a configuration space (C-space) representation to reduce our robot to a point.

At each position the robot tries to reach a close goal point, which can be the next waypoint or the coordinates of a random close GPS point in the safe wondering mode. Using this goal point, the current position of the robot, and the position of potential obstacles, we calculate two vectors: a “goal vector” and a “obstacle vector”. The goal vector is giving by the heading from the current to the goal position, normalized to be a unit vector. The obstacle vector is perpendicular to the vector from the current position to the center of the convex hull of the obstacles. If an obstacle is to the left of the current heading, then the obstacle vector points counterclockwise, and vice versa, acting like a curl vector. The magnitude of this vector is inversely proportional to the distance from the obstacle, and it is null when there is no obstacle in view.

The goal and obstacle vectors are added to one another to determine a modified goal vector. This modified goal vector is compared with the actual heading of the robot to get a heading error. Then finally the heading error is multiplied by a gain constant to form the increment to the current steering angle. Also in order to avoid deadlock conditions, when the ATV gets to close to an obstacle it stops and backs up until the obstacle is no longer too close, then goes back into forward and proceeds.

Because at the present we are not yet building a long-term map, our current algorithms are mainly reactive. In the long term we envision a navigation system similar to the one proposed in [8]. One scenario will be the part of the world that is currently in our field of view. Here we can perceive the world directly through our perception capabilities (Bright World). In this world any event can have an immediate impact on the performance of the robot, then the navigation should be guided by a set of reactive behaviors, similar to the ones described above. A second scenario will be given by the world that is out of the scope of our sensing capabilities (Dark World). In general the robot’s goal will be in this Dark World, then in order to do better than a simple random walk, the robot will need some kind of internal representation that will allow it to calculate a long-term plan. This representation will be giving by a topological graph of visual natural landmarks, augmented with GPS and visual servoing information. This map will be constructed in a distributed fashion, and it will be the basis for cooperative surveillance and navigation among the different robots.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have described the main features of our ATV reconnaissance and surveillance robots. We have successfully completed all the basic locomotion capabilities and a first version of the autonomous navigation, surveillance, and cooperation modes. We believe that the initial results we have presented here and elsewhere are promising for the development of a robust, adaptive system for distributed surveillance.

Our current and future work involves integrating the above-described capabilities in order to give the ATVs the ability to conduct a building "stakeout". In this scenario, a lead vehicle travels in autonomous mode towards a site whose general location is known, using DGPS, sidewalk tracking, and landmark-based tracking as appropriate and available. When it reaches a location from which building surveillance can begin, it communicates its position and relevant information about the path it traversed to the stay-behind vehicle, which then joins it at a location appropriate for good coverage of the building of interest. Coverage is adjusted in response to activity at the site. The ATVs will simultaneously conduct window detection and personnel/vehicle classification, as well as perform correspondence in order to track targets of interest. Using CyberRAVE, a remote user can intervene at any time in order to view information, manually adjust the action of an ATV, or provide tasking.

7. REFERENCES

[1] K. Dixon, J. Dolan, R. Grabowski, J. Hampshire II, W. Huang, C. Paredis, J. Salido, M. Saptharishi, and Pradeep Khosla. *"Rave: A Real and Virtual Environment for Multiple Robot Systems"*, IROS 1999.

[2] J. Dolan, A. Trebi-Ollennu, A. Soto and P. Khosla. *"Distributed Tactical Surveillance with ATVs"*, forthcoming in Proceedings of SPIE, Vol. 3693, AeroSense, Orlando, Fl., April, 1999.

[3] A. Trebi-Ollennu, J. Dolan, and P. Khosla, *"Adaptive Fuzzy Throttle Control for an All Terrain Vehicle"*, Institute for Complex Engineering Systems, Carnegie Mellon University, Internal Report 04-06-99, 1999.

[4] C. Diehl, M. Saptharishi, J. Hampshire II, and P. Khosla. *"Collaborative Surveillance Using Both Fixed and Mobile Unattended Ground Sensor Platforms"*, forthcoming in Proceedings of SPIE, Vol. 3693, AeroSense, Orlando, Fl., April, 1999.

[5] Liana M. Lorigo, Rodney A. Brooks, and W. E. L. Grimson. *"Visually-Guided Obstacle Avoidance in Unstructured Environments"*, IEEE Conference on Intelligent Robots and Systems, September, 1997.

[6] R. Gonzalez and R. Woods. *"Digital image processing"*, Addison-Wesley, 1992.

[7] L. Robert. *"Camera Calibration Without Feature Extraction"*, In Proceedings IEEE Intl. Conf. On Pattern Recognition, Jerusalem, 1994.

[8] A. Soto and I. Nourbakhsh. *"A Scenario for Planning Visual Navigation of a Mobile Robot"*, AAAI 1998 Fall Symposium Series, October 23-25, Florida, 1998.