Distributed Visual Servoing of a Mobile Robot for Surveillance Applications

David Rawlinson
david.rawlinson@eng.monash.edu.au
Punarjay Chakravarty
pcha25@student.monash.edu.au
Ray Jarvis
ray.jarvis@eng.monash.edu.au
ARC Centre for Perceptive & Intelligent Machines in Complex Environments: Intelligent Robotics, Monash University

Abstract

A mobile robot intercepts a target identified in the image plane of a fixed camera without prior knowledge of camera viewpoint or geometry. The target may be identified manually or autonomously. When the robot moves within the camera image plane, the system develops a table of mappings between the image plane and the ground plane in which the robot moves. The robot is identified in the image plane via background subtraction and localized in the ground plane using a particle filter. This adds the potential for robotic intervention to surveillance using fixed camera systems.

1 Introduction

Fixed camera video surveillance systems featuring automatic target identification and tracking have been actively researched for more than a decade [Collins, et al., 2000, Haritaoglu, et al., 2000, McKenna, et al., 2000, Stauffer and Grimson, 1999]. Some successful commercialisations have resulted, including ObjectVideo's VEW 2.0 [Lipton, et al., 2004] which detects, classifies and tracks objects in real-time. Many of the clever features of these systems are dependent on fixed gaze: For example, VEW invites users to segment the camera image plane; thus is the intelligence of the observer incorporated in the system.

Simultaneously, research into mobile robotics has developed to the extent that devices may expected to perform robustly during prolonged deployments in controlled indoor environments e.g. [Thrun, et al., 2000] report on a robotic museum guide, and ActivMedia, LLC. have produced the "PatrolBot" [Patrol bot, 2003] capable of effective navigation within large building complexes, tolerance to unexpected obstacles and long-range goal-seeking & path planning.

We propose a system incorporating both fixed cameras and mobile robots, able to exploit the benefits of both without introducing limitations on either. Such a union is inherently useful: Mobile devices add a critical element to security systems, namely the possibility of intervention.

A key problem for mobile robot intervention in locations identified by fixed cameras is how the target location known only as coordinates within the image plane of a camera can be expressed in a format useful to the robot's navigation system. The simplest solution would be to assume that merely moving to the approximate location of the camera is sufficient, and that the robot could thereafter continue using on-board sensors already compatible with navigation.

In practice, the complexity and clutter of many realistic environments and the fact that surveillance cameras may view large areas makes accurate target acquisition difficult. Therefore, it is highly desirable to be able to express the location of the target accurately, i.e. to possess a transformation between the image plane(s) of the fixed camera(s) and the ground plane used for robot navigation.

Ideally, transformations between planes would be produced without restricting or complicating the design, layout or implementation of either mobile or fixed elements of the composite system.

Here we describe our progress towards this goal. In our experiments a mobile robot is allowed to wander within an arena, a map of which is provided. A fixed camera observes most of the arena, & hence sometimes the robot. The robot is localized using 8 sonar, odometry & a particle filter.

The robot is distinguished in the camera image via. background subtraction. When the robot wanders within view of the camera, its positions i,j within the image plane and x,y (drawn from particle filter localization) within a prior map are continuously logged. After sufficient wandering, the resultant database of point transformations can be used to navigate the robot to specific locations within the image plane.

The system proposed does not limit the arrangement or number of cameras, nor does it constrain the type, locomotion or sensory capabilities of the mobile robot. The system can cope with visual obstructions to parts of the map, and very cluttered scenes; transformations can be produced for any location in which the robot is visible to the fixed camera(s). We regard automated surveillance as the primary application of this research.

The rest of the paper is organised as follows: Section 2: review of related work. Section 3: robot localization using a particle filter. Section 4: background subtraction for robot identification. Section 5: image to ground transformation. Section 6: robot navigation. Section 7: experimental results. Section 8: conclusions. Section 9: future work.

2 Related Work

Both target identification & tracking through fixed camera vision, and target interception via. mobile robot have been

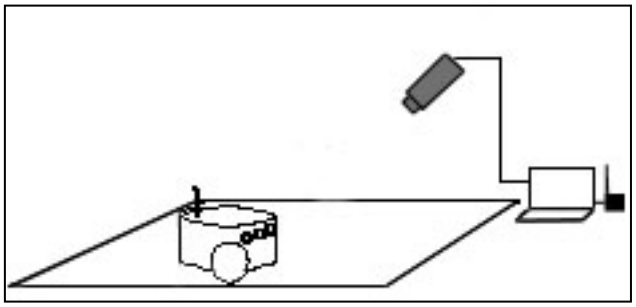


Figure 1. System setup (mobile robot and stationary camera share resources of a laptop computer)

investigated by a number of research groups in the past. However, cooperative systems that combine both are relatively unexplored, the exceptions being [Uppala, et al., 2002] and [Piaggio, et al., 2001].

[Uppala, et al., 2002] use 2 omni-directional cameras (one stationary, the other mounted on a mobile robot) to track a moving human target. A colour histogram model of the person is transmitted to the robot from the stationary camera, then used by the robot to approach said person.

[Piaggio, et al., 2001] use active beacons distributed throughout a hospital building to localize a robot. An Extended Kalman Filter combines information from the beacons and robot odometry data. In addition, the robot is provided with proximity sensors to avoid obstacles.

Target tracking in video sequences has been explored by the following researchers:

[Haritaoglu, *et al.*, 2000]'s W^4 is a real-time video surveillance system for tracking people and monitoring their activities in an outdoor environment. It utilises a single monochromatic stationary video source, enabling it to be used for outdoor surveillance tasks at night or in low light situations.

[McKenna, *et al.*, 2000]'s *TGP* is a computer vision system for tracking multiple people using a single RGB video source. It aims at providing a general and robust method for tracking people forming groups and interacting with each other while coping with occlusions and changes in illumination.

[Loy, et al., 2002] have demonstrated a system that adaptively allocates computational resources over multiple cues to robustly track a target in 3D.

The VSAM project [Collins, *et al.*, 2000] (sponsored by DARPA) was jointly undertaken by Carnegie Mellon University and the Sarnoff Corporation. It ran round-the-clock from 1997 to 1999 during which time it achieved an impressive array of results including: real-time moving object detection and tracking from stationary and moving camera platforms, recognition of generic object classes (e.g. human, sedan, truck) and specific object types (e.g. campus police car, FedEx van), object pose estimation with respect to a geospatial site model, active camera control and multi-camera cooperative tracking, human gait analysis, recognition of simple multi-agent activities, real-time data dissemination, data logging and dynamic scene visualization.

The museum tour-guide robots RHINO and MINERVA [Fox, et al., 2001, Thrun, *et al.*, 2000] demonstrate the suitability of the Particle Filter [Rekleitis, 2002] to indoor mobile robot localization. These robots, equipped with arrays of sonar sensors and 1 or 2 laser range finders successfully guided thousands of people through crowded museums. On this evidence we selected the particle filter for our experiment, although any succesful localization strategy would have sufficed.

3 Robot Localization using the Particle Filter

The Particle Filter has been used successfully in global localization problems (e.g. [Fox, et al., 2001]), including the "kidnap problem", in which a well-localized robot is teleported to some arbitrary unknown location. In this experiment, we have used a particle filter to localize our mobile robot in an indoor arena (Figure 1) in the Mobile Robotics Lab at the Intelligent Robotics Research Centre (IRRC) without knowledge of its initial position but given a prior map.

Figure 2. Pioneer mobile robot

3.1 Bayesian Reasoning

The Pioneer mobile robot (Figure 2) used in the system is equipped with 8 sonar sensors arranged across a 180 degree arc. The problem of estimating the current state *x* of the robot's pose given only current sonar measurements and previous pose estimates is solved by using Bayesian reasoning.

This approach assumes that the environment is Markov, that is, the past and future data are conditionally independent, if the current state is known. The state of the robot is estimated using a posterior probability density function, called the *belief*, conditioned on the data:

$$Bel(x_t) = p(x_t \mid d_{0...t})$$

where x_t is the state at time t, and $d_{0...t}$ denotes the data starting at time 0 upto time t. In our case, we will be dealing with 2 types of data : sonar range measurements $(y_0 ... y_t)$ and odometry data $(u_0 ... u_t)$.

$$Bel(x_t) = p(x_t \mid y_t, u_{t-1}, y_{t-1}, u_{t-2}, ... u_0, y_0) \qquad (1)$$

Equation (1) can be transformed by Bayes' rule to give the following recursive update equation:

$$Bel(x_t) = \frac{p(y_t \mid x_t, u_{t-1}, \ldots y_0)\, p(x_t \mid u_{t-1}, \ldots, y_0)}{p(y_t \mid u_{t-1}, \ldots y_0)}$$
$$= \frac{p(y_t \mid x_t, u_{t-1}, \ldots y_0)\, p(x_t \mid u_{t-1}, \ldots, y_0)}{p(y_t \mid u_{t-1}, d_{0\ldots t-1})}$$

Using the Markov assumption that measurements y_t are conditionally independent of past measurements and controls, given the knowledge of state x_t, we get:

$$Bel(x_t) = \frac{p(y_t \mid x_t)\, p(x_t \mid u_{t-1}, \ldots, y_0)}{p(y_t \mid u_{t-1}, d_{0\ldots t-1})}$$
$$= \eta_t\, p(y_t \mid x_t) \sum_{x_{t-1}} p(x_t \mid x_{t-1}, u_{t-1}, \ldots y_0)\, p(x_{t-1} \mid u_{t-1}, \ldots y_0)$$
$$= \eta_t\, p(y_t \mid x_t) \sum_{x_{t-1}} p(x_t \mid x_{t-1}, u_{t-1})\, p(x_{t-1} \mid u_{t-1}, \ldots y_0)$$
$$= \eta_t\, p(y_t \mid x_t) \sum_{x_{t-1}} p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1}) \qquad (2)$$

where $\eta_t = \frac{1}{p(y_t \mid u_{t-1}, d_{0\ldots t-1})}$ is a normalizing constant that ensures that the probabilities sum to one.

Equation (2) is the basis of the particle filter algorithm described in the next subsection.

3.2 Particle Filter Algorithm

The particle filter uses a set of N = 1000 probability-weighted particles to keep track of a multi-modal pdf over time. A series of control actions are taken, each one modifying the state vector of the robot according to some motion model. At certain times, a sonar reading arrives that constrains the state of the variable of interest at that time according to a perception model. To implement the Bayesian update equation (2), 3 distributions are required: the intial belief $Bel(x_0)$ (which is taken to be a uniform distribution over the entire state space), the next state probability $p(x_t \mid x_{t-1}, u_{t-1})$ (motion model), and the perceptual likelihood $p(y_t \mid x_t)$ (perceptual model). The algorithm is iterative, each iteration comprising the following steps:

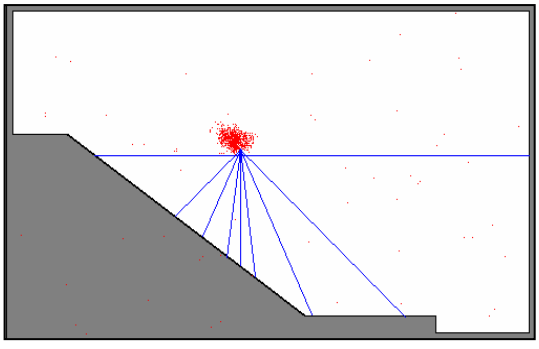


Figure 3. Ray tracing to produce theoretical sonar readings at state x_t

Drift

Relative odometry data is collated between each Drift step and applied to each particle.

Update

Each particle's weight is evaluated based on the perception model $p(y_t \mid x_t)$. This is computed in 3 steps. First, ray-tracing is used to produce theoretical sonar readings corresponding to x_t (the particle's proposed robot pose) in an ideal noise-free environment (Figure 3).

Second, the difference between each theoretical and actual sonar reading is calculated and stored.

Finally a one-dimensional scaled Gaussian kernel centred on 0 as shown in Figure 4 is used as a probability look-up table for each of the 8 differences. The pdf $p(y_t \mid x_t)$ is calculated as the sum of the 8 extracted probabilites; summation being used over multiplication to prevent the resultant values becoming very small.

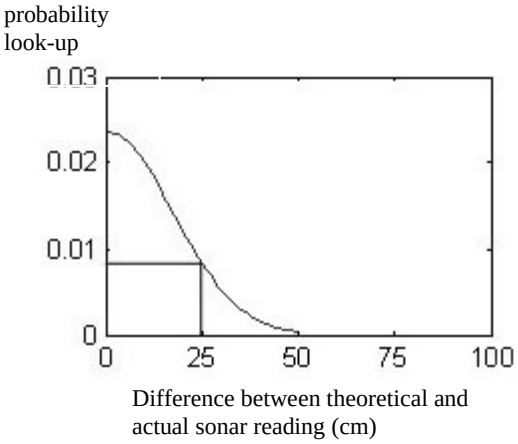


Figure 4. One dimensional scaled Gaussian kernel, centred on zero gives a look-up probability of 0.008 for a difference between theoretical and actual sonar reading of 25.

Resample

Roulette-wheel selection is used to resample 96% of the particles; particles with larger pdf having greater chance of reselection. The remaining 4% of the particles are redistributed randomly throughout state space. Random redistribution helps to prevent a total coalesce on an inferior solution and allows the robot to recover from a kidnap situation as can occur when robot communications are temporarily suspended.

Diffuse

Particles are perturbed from their positions by a small noise factor drawn from a normal distribution. Noise was applied to each particle property x, y and θ independently.

Extraction of Best Estimate

The best localization estimate was taken to be the probability-weighted arithmetic mean of the cluster of particles within radius *R* of the most probable particle.

4 Background Subtraction

The robot is distinguished in the images received from the fixed camera via background subtraction. Initially, the robot is absent from the scene, during which time a pixel-based model is constructed. Pixels in subsequent frames that do not match the model are labeled interesting; large connected regions of interesting pixels (blobs) more so. In the experiments reported here we took the largest blob in terms of pixel count to be the robot.

4.1 Background Model

We have used a bimodal distribution to model the background, similar to the background model used by the W^4 system of [Haritaoglu, *et al.*, 2000]. The background model is created as follows:

1. Successive frames of the background are taken from the camera for 1 minute, during which time the scene is stationary and the robot is absent.
2. The least intensity value (N(x)), the maximum intensity value (M(x)) and largest mean inter-frame difference value (D(x)) for each pixel over all frames collected in the previous step are stored as the background model.

4.2 Foreground Detection

In each iteration of the background subtraction, a pixel *x* from image I^t is segmented as a foreground pixel if:

$$\left|I^t(x) - M(x)\right| \geq kD(x) \vee \left|I^t(x) - N(x)\right| \geq kD(x) \quad (3)$$

where the modeled variance *k* was empirically determined to be 4. All other pixels are classified as background. Salt & pepper noise is filtered using a median filter leaving only large regions of connected pixels.

For the purposes of our experiment, it was assumed that the robot was the largest single foreground feature in the scene, identified by binary connected component analysis. The centroid of this region was taken to be the position of the robot in the image plane.

5 Transformation from Image to Ground Plane

Whenever the robot is confidently localized and visible in the camera image, the coordinates i,j indicate the robot's current location within the image plane. Simultaneously, the coordinates x,y represent the particle filter's best estimate of the robot's location within the ground plane.

When available, the coordinates x,y are stored within a data structure indexed by i,j. The structure has i*j buckets, each of which contains a running average of x,y values entered. Therefore, the structure is essentially a database of i,j to x,y point transformations.

The user selects a target within the image plane by clicking on the live video feed. The i,j coordinates of the click are used to index the database and all bucket content within 20 pixels radius of this point is averaged to produce an estimate x,y of the equivalent location in the ground plane. This estimate becomes the robot's goal. No goal is set if no point transfornations are available within the specified radius of the user's selected target.

Although this transformation technique is simple, it is quite effective and makes no assumptions about the scene in view beyond the theoretical existence of a single true mapping between the two planes.

6 Robot Navigation

A Finite State Machine (FSM) [Jones and Roth, 2003] controls movement of the robot, incorporating 3 behaviours:

1. Avoid Obstacles
2. Wander
3. Approach Goal

Behaviour 1 is always active; the robot will turn away from obstructions before hitting them. Behaviour 2 is by default active but subsumed by behaviour 3 when a target is selected (Switch state of FSM is triggered). The design of the FSM is detailed in Figure 5.

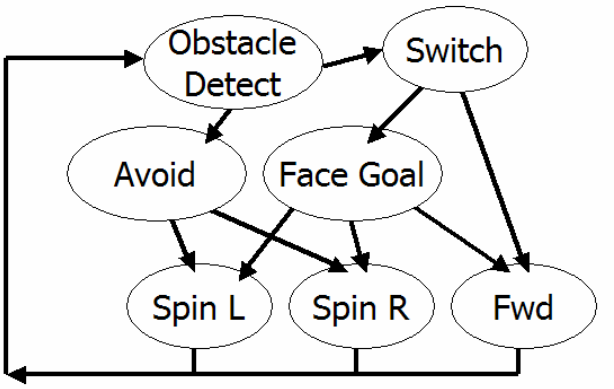


Figure 5. FSM controlling movement of the robot

7 Experiments

The robot communicates with a host computer using a radio modem. The computer is connected to a fixed camera by cable. In repeated trials, the robot is initally allowed to wander whilst the point-transformation database is constructed. Subsequently, the mouse is used to select target points in the video feed from the fixed camera. In all trials the robot moved directly to the location selected so that the body of the robot occluded the selected pixel. A video of one trial is available (see video 1).

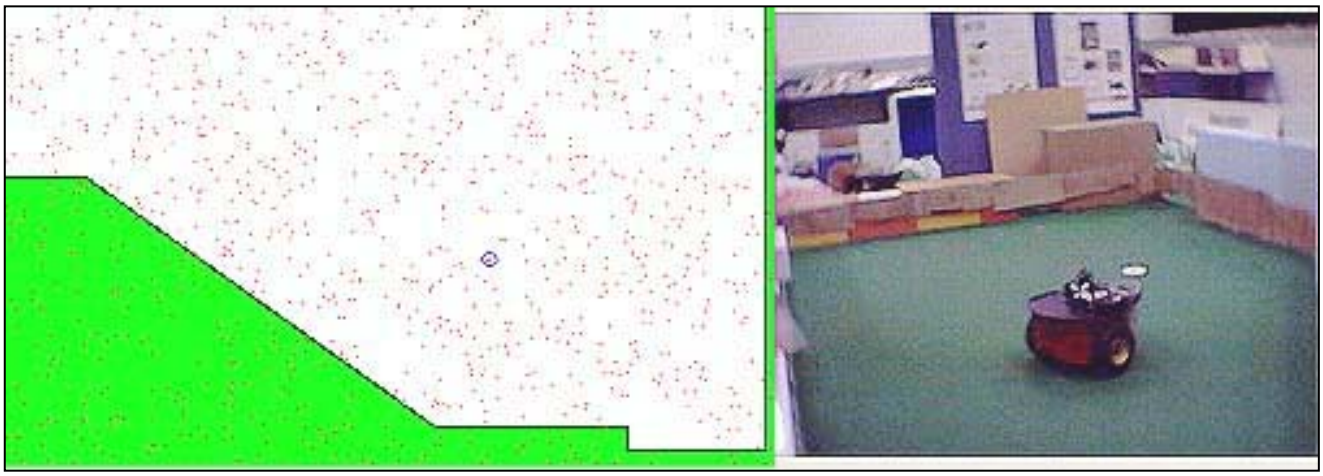
Included overleaf are screenshots from the host computer showing:

a. Before localization: The robot has recently been activated, and the particles are scattered throughout the arena.

b. Successful localization: A cluster has formed around the most probable location of the robot, which correctly corresponds with the location of the robot in the arena.

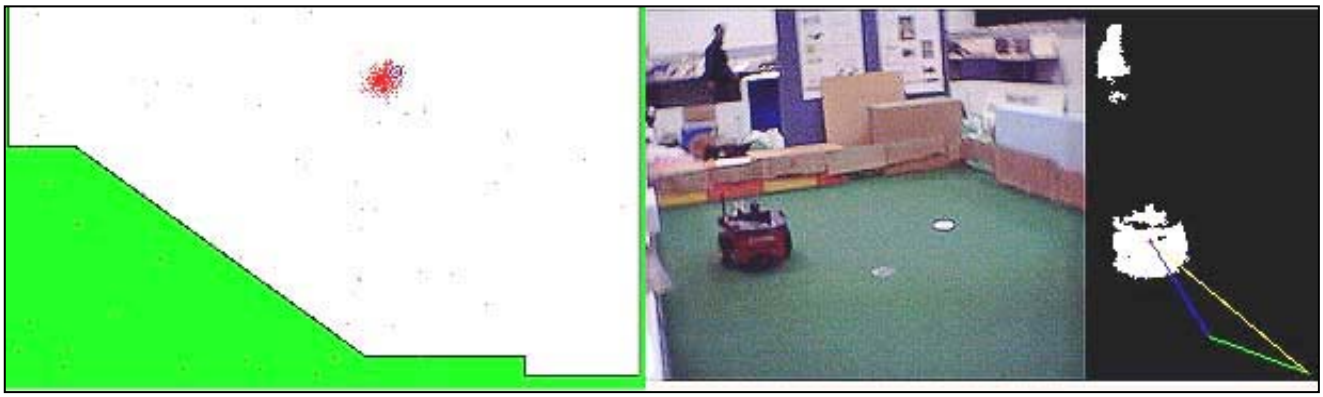
c. The particle filter has successfully localized the robot, and the robot has been identified in the video feed using background subtraction. The robot centroid is indicated by a red cross. Accidentally, someone has

walked across the camera's view and also been selected as foreground. Since the person is far from the camera, the robot is still correctly identified. This type of situation is not well handled in the current program, but will be addressed in future work.



(a)

(b)



(c)

Figure 6. (a) Particles scattered evenly throughout state space on start-up, (b) Successful localization: particles clustered around most probable robot location, (c) Robot localized on map of ground plane of arena, and in the image plane of the stationary camera

8 Conclusion

After ~10 minutes of wandering the robot can be directed to targets across most of the arena; given more wander time, the accuracy and coverage of the interception behaviour increases. In applications such as security or surveillance, hours, days or even weeks of wander time would be quite acceptable. Of course, our experiment employed a small arena and a single fixed camera so very little wander time was "wasted" out of view.

With many cameras and more ground to cover, days of wandering might be needed to log a useful number of point transformations across all cameras. However, wander time could be reduced through use of an algorithm designed to cover ground thoroughly and not revisit.

The FSM with which goal-seeking behaviour was achieved causes the robot to repeatedly cross the target point in the robot's map from different directions. This helps to mitigate error in the transformation from the image plane.

The simple local averaging system we used to generate robot map (ground plane) coordinates from an image plane target was actually quite adequate. The robot appeared to move directly to the location selected in the image, with error less than the dimensions of the robot itself. For intervention purposes as outlined in the introduction, it seems reasonable that once the robot has moved to within a metre of a target on-board sensors could be used to accurately perform the intervention task.

The largest component of the error in our transformation from image to ground plane coordinates originated from our use of the robot blob centroid as the image plane location, when the camera was mounted slightly above and far away from the robot. The centroid approach is really only valid when the camera is mounted directly above the robot. Had we been willing to impose some restrictions on camera placement we might have reduced this error dramatically by taking the blob baseline as the robot's location in the image plane.

Since the robot is localized independently via sonar it need not remain in view of the camera (our arena deliberately included an obscured corner); it is then also possible that the robot could wander between the visual fields of multiple cameras. Navigation of the robot to **any** point selected from **any** camera image would be trivial using the existing system of point transformations to the ground plane keyed by image plane coordinates. In the case where multiple cameras' visual fields overlap the cameras could still be treated independently, but it is likely that even crude amalgamation of transformations generated from additional perspectives would improve accuracy.

We would like to highlight here the flexibility of the system described in this paper by noting some constraints and limitations that do **not** apply:

- The ground plane does not have to be flat; since we take the average of local point transformations, we do not assume the presence of a ground *plane* at all; instead, we assume merely that there is only 1 true mapping for any point in the image plane. Homographic transformations guarantee 1:1 correspondence between individual points in both planes, and would likely exhibit greater accuracy but in practice are limited by the presumption of a single ground plane.

- The camera view can be obstructed (in the real world, scenes are often cluttered). This is not a problem since we envisage the fixed camera [Patrol bot] driving detection of targets: The result is that the robot cannot be directed to intercept targets in locations the camera[s] cannot see.

- Cameras can be located for convenience & utility, and their locations & perspectives do not have to be accurately measured and entered into the system as parameters.

- The cameras need not have a contiguous field of view; that is, the robot need not be viewed continuously.

- The system does not place restrictions on camera angles, distances, etc.

9 Future Work

In future work we would like to extend our system to demonstrate the use of multiple cameras and a larger more complex arena. It is hoped that such a system could be created by simply storing a database of point transformations for each camera.

Real applications would require that the target appear in the camera image in addition to the robot. This would require more sophisticated recognition of the robot so that it could be distinguished from targets and other irrelevant features that background subtraction or other image processing techniques might highlight. Given that the robot already makes its movements known to the camera processing system, synchronicity (correlation in time) might be used to learn the appearance of the robot in a form invariant to changes in position, configuration and orientation – perhaps using colour metrics similar to those of [Uppala, et al., 2002].

To maintain the accuracy of the point-transformation database it would also be necessary to introduce some robot recognition-confidence criterion, filtering point transformations when the robot is not clearly distinguished.

References

[Collins, et al., 2000] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt and L. Wixson. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Carnegie Mellon University, Pittsburgh.

[Fox, et al., 2001] D. Fox, S. Thrun, W. Burgard and F. Dellaert. Particle filters for mobile robot localization. in Arnaud Doucet, Nando De Freitas and Neil Gordon eds. Sequential monte carlo methods in practice, Springer-Verlag, New York, 2001.

[Haritaoglu, et al., 2000] I. Haritaoglu, D. Harwood and L. S. Davis. W4: Real-time surveillance of people and their activities. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):809-830.

[Jones and Roth, 2003] J. Jones and D. Roth. A practical

guide to behavior-based robotics. McGraw-Hill/TAB Electronics, 2003.

[Lipton, et al., 2004] A. J. Lipton, C. H. Heartwell, N. Haering and D. Madden, "Critical asset protection, perimeter monitoring, and threat detection using automated video surveillance," http://www.objectvideo.com/whitepapers.asp. (Accessed: 1 Aug 2004).

[Loy, et al., 2002] G. Loy, L. Fletcher, N. Apostoloff and A. Zelinsky. An adaptive fusion architecture for target tracking. In Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, Washington DC, 2002.

[McKenna, et al., 2000] S. J. Mckenna, S. Jabri, Z. Duric, A. Rozenfield and H. Wechsler. Tracking groups of people. Computer Vision and Image Understanding, 80:42-56.

[Patrol bot, 2003] Patrol bot, "Patrol bot: Mobile agent for automated remote monitoring & surveillance for security systems environmental monitoring in standard and automated buildings," http://www.mobilerobots.com/PatrolBotBroch3.pdf. (Accessed: 28 Aug 2004).

[Piaggio, et al., 2001] M. Piaggio, A. Sgorbissa and R. Zaccaria. Autonomous navigation and localization in service mobile robotics. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, 2001.

[Rekleitis, 2002] I. M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, McGill University, Montreal, Quebec.

[Stauffer and Grimson, 1999] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In Proceedings of the International Conference on Computer Vision and Pattern Recognition, 246-252, Ft. Collins, CO, U.S.A, 1999.

[Thrun, et al., 2000] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. International Journal of Robotics Research, 19(11).

[Uppala, et al., 2002] S. Uppala, D. R. Karuppiah, M. Brewer, S. C. Ravela and R. A. Grupen. On viewpoint control. In Proceedings of the International Conference on Robotics and Automation, Washington, DC, 2002.