Indoor active surveillance

Nelson Gonçalves, João Sequeira, M. Isabel Ribeiro
Instituto Superior Técnico
Institute for Systems and Robotics, Portugal
{ngoncalves,jseq,mir}@isr.ist.utl.pt

Madhavan Shanmugavel, Antonios Tsourdos, Brian White
Department of Aerospace, Power and Sensors
Cranfield University, UK
{m.shanmugavel,a.tsourdos,b.a.white}@cranfield.ac.uk

Abstract

This paper describes a robotics based surveillance system for indoors operation. The system is capable of performing common tasks autonomously such as detecting an intruder and having a robot approaching it while the rest of the team spreads to maximize the coverage of the area.

The system developed is a small size networked system with fixed cameras connected over Ethernet and three mobile platforms used as active agents.

The paper presents details of the prototype system, experimental results, and discusses the scalability of this setup when the area under surveillance is expanded.

1. Introduction

This paper describes a robot-based active surveillance system for indoor operations. The system is composed of three mobile platforms, with onboard processors and ultrasound sensors, a set of fixed cameras and a network of personal computers. For the purpose of the paper, the surveillance task is simplified to the detection and interception of intruders in a non structured laboratory environment.

Surveillance tasks are a typical example of tasks where the type of motion adopted by the robots conveys information to the environment, including intruders being chased. For instance, the robot might adjust its motion depending on its own interpretations of the intentions of the intruder. If the intruder is moving slowly, a low threat degree can be assumed and only a single robot in the team moves to intercept the intruder. If the intruder is moving fast, multiple robots might be given the interception task, eventually in a coordinated form.

The work in the literature on robot control architectures is immense. The specific robotic surveillance application has a number of aspects common to architectures used in other applications, e.g., path planning and following, obstacle avoidance, world mapping, and localization. Examples of single and multiple robot architectures using concepts from artificial intelligence, biology, semiotics and economic trade markets are widely available (see for instance (Parker, 1998; Sequeira and M.I. Ribeiro, 2006; Dias *et al.*, 2005)). Current robotics research extends these generic architectures to cope with distributed decision making, networked robotics (NR), and advanced human-robot interaction techniques and applies them to specific problem domains such as surveillance. The work described in this paper emphasizes the integration of most of these key aspects that need to be present in a surveillance system.

Among the NR systems communications is often the emphasized aspect. For example, (Woo *et al.*, 2003) proposed a three tier architecture with application, infrastructure and middleware layers. The application layer contains the functional blocks related to single robot activities, e.g., path planning. The infrastructure layer provides the network services. The middleware layer handles the communication between services. The Miro framework, (Enderle *et al.*, n.d.) is a CORBA based, distributed object oriented, middleware layer providing network transparency, event based publishing, logging facilities, and sensor and actuator services. The MRHA architecture for indoors security patrolling tasks is described in (Laird *et al.*, 1995). It is based on the seven layer OSI/ISO reference model with the robots seen as resources controlled by a set of networked computers. The network includes supervisor, distributed databases, planning/dispatching and communications computers. The supervisor manages the overall system. The database computers estimate the positions of the robots over time using the raw data acquired from the robot. The planner/dispatcher computers are in charge of navigation and collision avoidance. The communications computer handles the wireless communications.

Robots have been employed in commercial surveillance systems mainly as mobile sensor platforms. The PatrolBot (www.mobilerobots.com) is used in the surveillance of buildings like the Victoria Secret's headquarters at Columbus, USA, and in the United Nations building at Geneva, Switzerland. Mostitech (www.mostitech.com), Personal Robots (www.personalrobots.com), and Fujitsu (www.fujitsu.com) currently sell robots for domestic intruder detection. These robots inform the home owner when an intruder is detected. In military and police scenarios the robots are, in general, teleoperated to gather information on the enemy positions and in explosives ordinance disposal, (Nguyen and Bott, 2000; Everett, 2003). The Robowatch (www.robowatch.de) robot is equipped with ultrasound, radar, video and infrared sensors. It is supervised by a human through a graphic interface and allowed limited autonomy. These robots do not aim cooperative operation with other robots or humans. Upon detecting unexpected events they just signal a human supervisor.

A team of miniature robots with onboard cameras is considered in (Rybski *et al.*, 2000) for reconnaissance and surveillance tasks. The limited computational capabilities of their robots require that image processing, and the decision processes are done off board and the low level commands sent to the robots through a RF link. A CORBA based architecture coordinates all the resources in the system.

In the specific case considered in this paper, the network is composed by of-the-shelf Pionner robots and fixed video cameras installed in a realistic, medium structured, indoors environment. This type of environment is close to what can be found in scenarios monitored by commercial systems. The paper is organized as follows. Section 2 describes the global architecture. Section 3 details the experimental setup and presents a set of experiments. Section 4 presents the conclusions and future work.

2. The active surveillance robot control architecture

In applications dealing with a large number of robots and fixed sensors, the use of distributed decision schemes supported in service oriented information systems is a common approach to control architectures.

For small size setups, such as monitoring a small warehouse or a small number of floors in an office building, the control architecture can be predominantly centralized, simplifying the communications issues.

This is the case of the system described in this paper. Figure 1 illustrates the architecture being used in the system described in this paper.

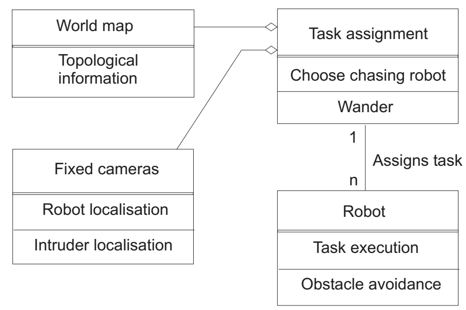

Figure 1. The overall control architecture

This architecture is defined on the following operational assumptions,

- There are no dynamic obstacles other than the robots,
- Only one intruder can enter the area under surveillance,
- Only one robot used to intercept the intruder,
- The cameras field-of-view cover completely the space under surveillance.

The robots have two concurrent primitive behaviors, running locally, namely (i) executing the assigned task and (ii) avoiding obstacles.

2.1. Task assignment

When an intruder is detected the task assignment automaton defines the goal positions to be reached by each robot as follows:

- The robot that is closest to the intruder is assigned the intercepting task;
- The rest of the team must cover the environment such that the distance between the robots is maximal (in a sense, this maximizes the area covered).

Under the wander group behavior the robot will simply move forward until the local obstacle avoidance is active.

The world map is a topological map based on a Voronoi diagram of the environment. The vertexes of the a priori known obstacles and the boundary on the environment are used to compute a 2D Voronoi diagram. The segments that intersect an obstacle or the boundary are eliminated from the diagram, yielding a topological map that contains only segments entirely contained in free space. The resulting diagram yields a visibility graph such that any task related to motion can be defined over this diagram.

Figure 2 shows an example of a topological map, constructed as above, for the indoor environment used for the experiments in this paper. The indoor environment corresponds to the lower area in the map. Artificial obstacles of polygonal shape, were introduced sparsely, close to the boundary of the area under surveillance.

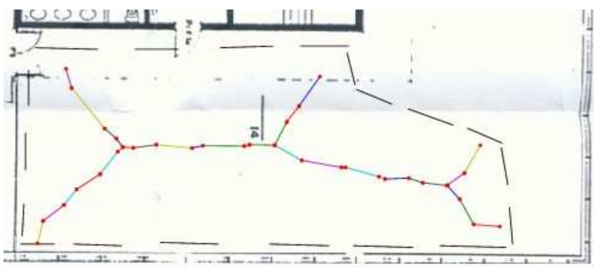

Figure 2. A topological diagram for an indoor environment

The topological map resembles the 2D Voronoi diagram when obstacles are described by polygons (instead of points). Note that, if needed, segments with endpoints close to obstacles can be easily transformed (or simply removed) using a distance criteria.

Once the visibility graph is computed, the Floyd-Warshall algorithm is used to determine the adjacency matrix that represents the shortest distance between any pair of nodes. Let this matrix be denoted by L, with L_{ij} the distance measured over the topological map between nodes n_i and n_j, the position of the robots by $r_i, i = 1..3$, the position of an intruder by q, and $g_i, i = 1..3$ the goal position for each of the robots. The tasks for each robot, that is the goal positions, when no intruder is detected, are defined as,

$$g_i = n_i,\ \ g_j = n_j,\ \ g_k = n_k \tag{1}$$

s.t. maximizing the distance between nodes

$$\max_{n_i,n_j,n_k} \left(d(n_i,n_j) + d(n_j,n_k) + d(n_i,n_k)\right)$$

The maximizing nodes, n_i, n_j, n_k can be found using exhaustive search. A sub-optimal solution, easily found, consists in selecting two nodes by (i) searching the entries with the highest value in the L matrix, and (ii) using rows/columns that correspond to these nodes to search for the third node.

When an intruder is detected the tasks are defined as,

$$g_i = q\ :\ \min_i \|q - r_i\| \tag{2}$$

$$g_j = n_j,\ \ g_k = n_k$$

s.t.

$$n_i\ :\ \min_i \|q - n_i\|$$

$$\max_{n_j,n_k} \left(d(n_i,n_j) + d(n_j,n_k) + d(n_i,n_k)\right)$$

The node n_i is the closest to the intruder position. The other two nodes are found by searching for the highest values on the rows/columns that correspond to n_i in the L matrix.

2.2. Path generation

Using the topological map, each robot executes its task by moving through the sequence of nodes forming the shortest path between its current and goal positions. These points are used to generate the reference trajectories to be followed by the robots when executing their tasks.

The locations are prescribed with their position coordinates (x, y) and orientation θ, together called a pose. Given a set of initial pose P_{si} and a final pose P_{fi} for each i^{th} robot, the path planner produces a path $r_i(t)$, satisfying maximum curvature bound of the robot, κ_{max}. The path $r(t)$ can be a single or a composite path.

$$P_{si}(x_{si}, y_{si}, \theta_{si}) \xrightarrow{r_i(t)} P_{fi}(x_{fi}, y_{fi}, \theta_{fi}),\ \kappa_i(t) < \kappa_{i,max} \tag{3}$$

The path planner uses Dubins path (L.E.Dubins, 1957) for the path planning and uses the shortest path from the Dubins set (Shanmugavel *et al.*, 2005). Dubins path is the shortest path connecting two poses in a plane under the constraint of maximum curvature (L.E.Dubins, 1957). It is a composite path formed either by connecting two circular arcs by their common tangents or by three consecutive tangential circular arcs. The former is CLC path and the latter is CCC path, where 'C' stands for Circular arc and 'L' stands for Line segment. In simple terms, the straight line is shortest path for rectilinear motion and the circular arc is the shortest path for turning. Combining these two paths provides the shortest path for a general motion of a robot in a plane. Though extensive work on Dubins path can be found in the literature, some of the recent work can be found in (Shanmugavel *et al.*, 2005), (Enright and Frazzoli, 2005), (Shanmugavel.M *et al.*, 2006) and (Savla *et al.*, 2006) on path planning and in (Robb *et al.*, 2005) for missile applications. In this paper, we are using the CLC paths for path planning and 'Dubins path' connotes 'CLC path'.

Figure 3 shows an example of the trajectories for a typical mission. Robot 1 moves towards the Intruder position as it is the closest at detection time. Robot 2 moves to Goal 2 and Robot 3 to Goal 3 in order to approach the sub-optimal solution for problem (2).

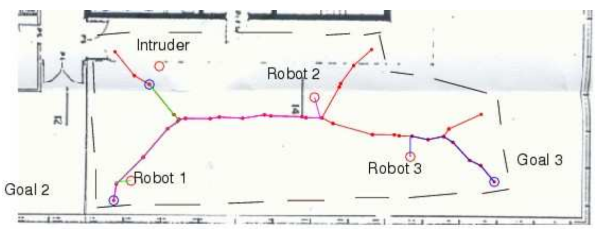

Figure 3. Generated Dubins paths

Figure 4 shows a detail of the trajectory generated for Robot 3 where the smoothing of the Dubins path is clearly visible.

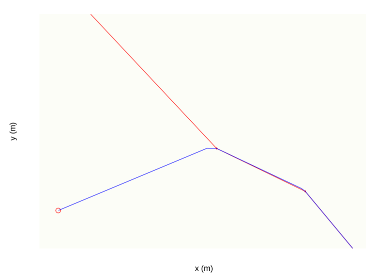

Figure 4. Detail of the Dubins path generated

2.3. Low level control

The low level control is responsible for generating the primitive robot behaviors. Because the robot must move without collisions through the environment, the obstacle avoidance primitive subsumes the task execution when necessary.

Unicycle robots are used. The control inputs are the desired forward velocity, u_1, and the desired angular velocity, u_2. The controllers used in each behavior have an identical structure:

$$\begin{aligned} u_1 &= V \tanh(\|\rho\|) \\ u_2 &= W \tanh(\alpha) \end{aligned} \tag{4}$$

where ρ is the vector from the current robot position to the desired position, α is the robot orientation error, V and W are the maximum forward and angular velocities of the robot. The computation of ρ and α is particular to each behavior. For the experiments presented, V and W where set to $0.3\,m/s$ and $0.8\,rad/s$ respectively.

The obstacle collision avoidance is activated when the robot sonars indicate the presence of an obstacle closer than a pre-defined safety distance. In this situation the robot behavior is to rotate until an obstacle free direction is found and moving in that direction, away from the obstacle. For this behavior, the sonars are used to determine an obstacle free direction. Then α is the error between the current robot orientation and the obstacle free direction determined by the sonars. To avoid collisions due sonar measurement errors, the robot rotates in place, and $\|\rho\|$ is set to zero.

The wandering task is defined as moving forward in a straight line, at a constant velocity. This task is similar to the obstacle avoidance primitive behavior, except the initial motion direction is selected by the decision automaton. In this behavior the desired orientation, α, is set to zero and $\|\rho\|$ is set to a constant, grater than zero value. Thus, the robot will move as if chasing a point always in front of it.

The path following behavior input are the Dubins paths, described in Section 2.2. These paths are sets of straight line segments and circular arcs. The circular arcs are approximated by a finite set of small line segments. Each line segment is represented by a pair of points. As a result, the overall path is approximated by a finite set of points which the robot follows in sequence. Each point is transformed to the robot frame, resulting in an error vector, ρ. The orientation error is computed from error vector, $\alpha = \angle\rho$.

Figure 5 shows an experimental run of the path following behavior. The path planner was given the way-points represented by the open circles, in the order indicated. The computed path, the solid line, was then approximated by line segments. The endpoints of these line segments are the points represented by full circles. This set of points was the input given to the path following behavior. The robot started at position $(0,0)$, and the path executed is represented by the dashed line. The error between the executed and desired paths is mainly due to the approximation of the path by line segments.

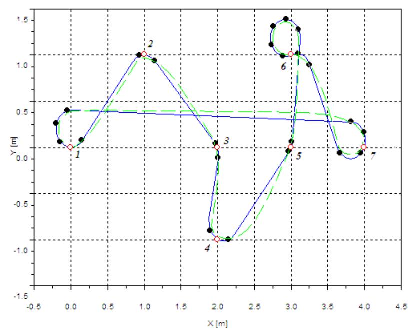

Figure 5. Example of the path following controller

3. Experimental setup and results

The experimental setup consists of four main components: i) centralized decision scheme, ii) a communication framework, iii) fixed surveillance cameras and iv) mobile robots.

A set of three standard of-the-shelf webcams where mounted on the ceiling of the laboratory (see Figure 6). Their purpose is to detect intruders present in the environment and also to estimate the pose of robots visible in the image. The cameras where placed so that, despite their limited resolution, an intruder can be detected anywhere in the environment. The robot pose estimation, on the other hand, required an higher image resolution or more cameras to be available. As a consequence, in some locations of the environment the robots have available only odometry information.

Figure 6. A view of the indoor lab environment

The cameras where all calibrated using colored markers placed on the ground, at known positions. After the calibration procedure, on average, each camera identified points on the floor close to the markers with an error of less than 5 *cm* in each coordinate. For distant points, the error was found to be roughly 10 *cm*.

Three Pioneer P3AT mobile robots were used. Communication between the robots, the central decision automaton and the cameras was available through an wireless link. No information was exchanged between the robots.

The centralized decision was implemented in Matlab using the StateFlow tools on a dedicated desktop computer. All of the other components where coded in C++ language.

3.1. Communications framework

The communication framework was implemented using the YARP library, (G. Metta, 2006). In the design philosophy of this library, each process owns input and output ports, used to exchange data with other processes. The ports are identified by their textual names. A central name server is used to maintain the correspondence between a symbolic port name and the port network address. The library also provides wrappers for data communication independent of the supporting operating system, the network configuration and the low-level protocols.

In the experimental setup, each camera owns two output ports. These two ports broadcast to the network the position of the detected intruders and the estimated pose of the robots in the image frame.

Each of the robots also owns two ports in charge of, respectively, reading the the task assigned to the robot, and writing the robot self estimated pose.

The intruder port of each camera and the pose port of each robot are read by the central decision scheme.

3.2. Intruder detection

The intruder detection by the cameras was performed using an image differencing algorithm for motion detection. Despite the algorithm simplicity, experiments showed it was able to detect motion anywhere in the image frame. The disadvantage of this approach is that if an intruder suddenly halts it will cease to be detected. But if the intruder does not move, then it cannot cause damage to the environment.

The movement of the robots is also perceived by the motion detection algorithm. Thus the estimation of each robot pose is used to eliminate false positive intruder detections.

It is assumed that only one intruder can be present. This is not a severe limitation on the active surveillance system. If the presence of more intruders is expected, only the intruder detection and tracking algorithm would require modification. The other elements of the architecture would remain unchanged.

The cameras observe the intruder position, which is assumed to be a gaussian distributed variable. To estimate the variable mean value, μ, and covariance matrix,Σ^{-1}, the following likelihood function is maximized:

$$L(x_1,\ldots,x_n|\mu,\Sigma^{-1}) = \prod_{i=1}^{3} N(x_i|\mu,\Sigma^{-1}) \quad (5)$$

where x_i is the observation of the i-th camera and $N(.)$ is the gaussian distribution. The mean value, μ, is assumed to be the intruder position. To reduce the uncertainty in the position estimation, more cameras should be used.

Figure 7 shows the result of the intruder detection algorithm (for a single intruder).

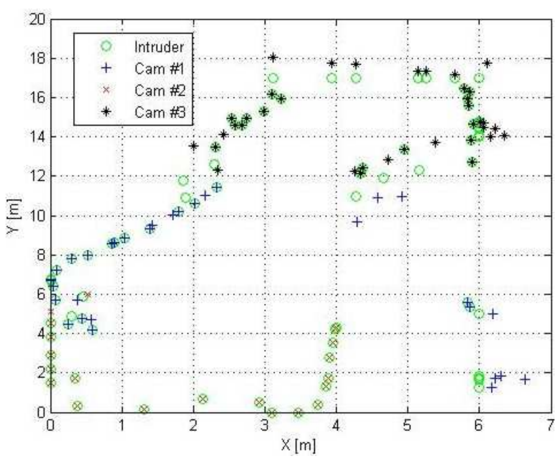

Figure 7. Intruder detection example

The circles represent the intruder estimated position. The remaining symbols mark the intruder detections by the different cameras.

3.3. Robot pose estimation

To ease the detection of the robots pose by the cameras, each robot was equipped with a unique pair of color markers. With the application of the appropriate color filters to the captured images, the pose of each robot could be determined.

The estimation of the robot pose from the cameras is combined with the robot odometry pose estimation using an Extended Kalman Filter (EFK). The process noise is assumed to have a zero mean gaussian distribution. This noise is additive to the robot forward and rotation velocities.

The observation model consists of a value for the pose estimation received from one of the cameras, with zero mean gaussian noise added. Because the cameras calibration errors are small, the observation noise variance is assumed equal and constant on all cameras. Although using this noise model, the EKF filter implementation is simplified in practice it will not be very accurate. This is because the gaussian noise has a non null mean and the noise variance is not equal for all image pixels, neither for all cameras. As a consequence the EKF filter performance will be poorer, e.g. exhibit slow convergence with an offset error.

Figure 8 shows the self-localization estimation for one of the robots, while wandering at a constant velocity of $0.2m/s$ through the environment. The robot position estimation is represented by the solid line, and the robot initial true ground position is marked with the $*$ sign. The other symbols represent the different cameras position estimations for the robot. The robot initial position belief is $(3.32, 1.38)$.

From Figure 8 it can be seen that the different cameras produce different estimates for the same robot position. Although the EKF filter convergence is slow, as expected, the robot and the cameras estimative eventually converge. But due to the zero mean gaussian assumption, the position estimation from the cameras has an offset error.

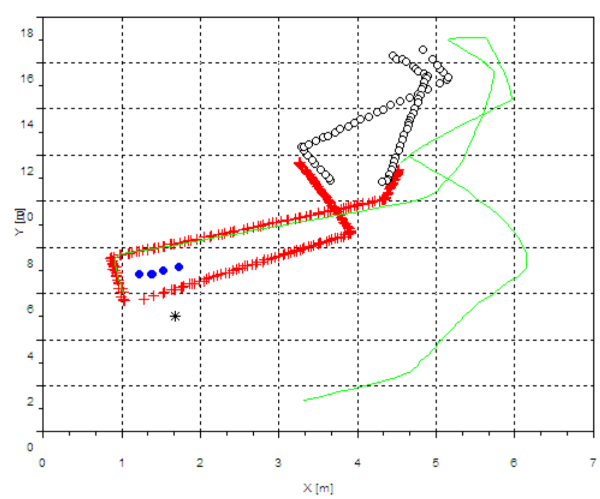

Figure 8. The self-localization estimation of one robot

3.4. Experimental results

Two experiments where conducted to assess the different components of the architecture. The first experiment assesses the low level control system at each robot integrated with the intruder detection system (in essence a visual servoing experiment). In Figure 9 a single robot is at rest, at position $(0,0)$ when an intruder is detected. While the robot starts moving, the position of the intruder changes according to the three clusters of $\star$ marks, in the right part of the figure. When the robot reaches a neighborhood of the intruder, he suddenly starts moving left, leading the robot to a sharp left turn, following the intruder motion.

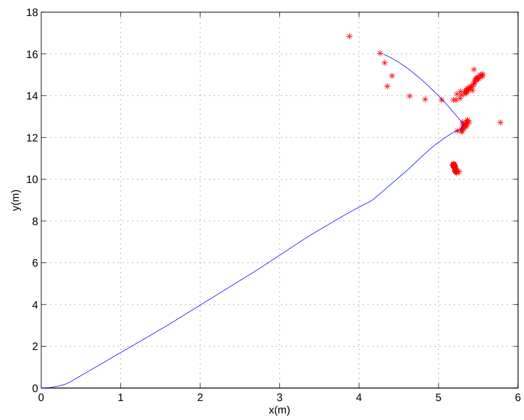

Figure 9. A single robot chasing an intruder

In the second experiment, Figure 10, the robots wandered through the environment until an intruder was detected after which one of them chases the intruder while the others spread around to maximize the area covered. The position of each robot was obtained from the robots self-pose estimation. In this experiment, the robots first wandered through the environment until the EKF pose estimation converged. The intruder was detected at $(4.7,6.5)$ and it was assigned to Robot 2. Robot 1 was assigned the goal position at $(5.8,14)$ and Robot 3 at $(6.5,1)$.

Despite all robots initially moved to their assigned goals, only Robot 2 was able to reach the intruder. In the case of Robot 1 and Robot 3, their goals are located at extreme points of the Voronoi diagram, close to obstacles. When moving towards their goals the robots obstacle avoidance behavior became active and the robots where unable to reach their goals.

4. Conclusions

This paper described the deployment of a prototype system to test active robotics surveillance in indoor environment. The system relies on a small number of mobile robots and fixed cameras with the underlying communication framework being implemented over open source technology.

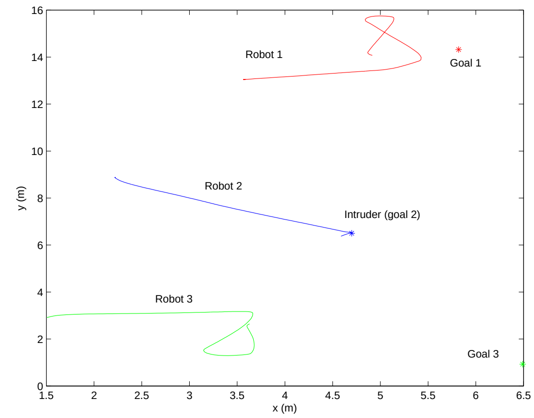

Figure 10. Multiple robots chasing an intruder and watching for the rest of the environment

The active surveillance system was installed in a realistic, medium structured, indoors environment. This type of environment is close to what can be found in scenarios monitored by commercial systems.

The experimental results clearly show that a good performance in this kind of application can be achieved with the integration of common subsystems. Robot control scheme, albeit simple, shows a good performance. Image processing proved too sensitive to environmental conditions, namely lighting conditions.

Some design aspects where found to limit the system performance. For instance, the surveillance system relied almost entirely on a single type of sensor, the web cameras. Also, the web cameras where placed at fixed locations. Although the use of diverse types of sensors could improve the system sensing capabilities, mobile sensors would also provide a better coverage of the environment. For example, cameras mounted on top of the mobile robots could monitor regions not visible by the fixed cameras.

Because of the small number of agents present in the system, the use of a centralized decision scheme was justified. But if more functionalities or agents where added to the system, then a decentralized decision scheme could greatly improve the overall system performance and robustness.

As a consequence of the limitations found, the system resources where not used in an efficient manner. For example, mobile robots could locally compute their paths. And if equipped with cameras, they would also detect intruders. For the setup presented in this paper, the resource usage efficiency is not an important aspect. But it may deem unfeasible or too costly, to build bigger active surveillance systems.

Future work includes (i) the improvement of the image processing algorithms, (ii) the improvement of the decision scheme, namely including strategies for expressive motion by the robots (in order to improve human-interaction capabilities) and (iii) extending the decentralization of the functionalities.

4.1. Acknowledgments

This work was supported by the Windsor Treaty Program Anglo-Portuguese Action B-11/06 and ISR/IST plurianual funding through the POS_Conhecimento Program that includes FEDER funds.

Nelson Gonçalves is working under grant SFRH/-BD/23804/2005, from Fundação para a Ciência e a Tecnologia.

References

Dias, B.M, R. Zlot, N. Kalra and Stentz A. (2005). Market-based multirobot coordination: A survey and analysis.

Enderle, S., S. Sablatnoeg and H. Utz (n.d.). MIRO - middleware for robots.

Enright, J. J. and E. Frazzoli (2005). On multiple UAV routing with stochastic targets: Performance bounds and algorithms. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*.

Everett, H.R. (2003). Robotic security systems. *IEEE Instrumentation & Measurement Magazine* **6**(4), 30–34.

G. Metta, P. Fitzpatrick, L. Natale (2006). YARP: yet another robot platform. *International Journal on Advanced Robotics Systems. Special Issue on Software Development and Integration in Robotics*.

Laird, R.T, H.R. Everett, G.A. Gilbreath, T.A. Heath-Pastore and R.S. Inderieden (1995). MDARS: Multiple robot host architecture. In: *Procs. Association of Unmanned Vehicle Systems, 22nd Annual Technical Symposium and Exhibition (AUVS '95)*.

L.E.Dubins (1957). On curves of minimal length with a constraint on average curvature and with perscribed initial and terminal positions and tangent. *American Journal of Mathematics*.

Nguyen, H.G. and J.P. Bott (2000). Robotics for law enforcement: Applications beyond explosive ordinance disposal. *Procs of SPIE on Enabling Technologies for Law Enforcement and Security* (4232), 433–454.

Parker, L. E. (1998). ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation. *IEEE Transactions on Robotics and Automation* **14**(2), 220–240.

Robb, M., B. White and A. Tsourdos (2005). Earliest intercept line guidance: A novel concept for improving mid-course guidance in area air defence. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*.

Rybski, P., D. Hougen, S. Stoeter, M. Gini and N. Papanikolopoulos (2000). Control of multiple small surveillance robots at AAAI 2000. In: *Procs. AAAI 2000 Mobile Robot Competition and Exhibition Workshop*.

Savla, Ketan, Emilio Frazzoli and Francesco Bullo (2006). On the dubins traveling salesperson problems: novel approximation algorithms. In: *Robotics: Science and Systems II*. MIT Press, Cambridge, MA.

Sequeira, J. and M.I M.I. Ribeiro (2006). Human-robot interaction and robot control. *Springer's Lecture Notes in Control and Information Sciences* **335**, 375–390.

Shanmugavel, Madhavan, Antonios Tsourdos, Rafał Żbikowski and Brian A. White (2005). Path planning of multiple uavs using dubins sets. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*.

Shanmugavel.M, A. Tsourdos, R. Żbikowski and B.A.White (2006). 3D dubins sets based coordinated path planning for swarm of UAVs. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado*. AIAA-2006-6211.

Woo, E., B.A. MacDonald and F. Trépanier (2003). Distributed mobile robot application infrastructure. In: *Procs. Int. Conf. on Intelligent Robots and Systems (IROS03)*. pp. 1475–80.