

2004

# Optimal Sensor Placement for Cooperative Distributed Vision

Luis E. Navarro-Serment  
*Carnegie Mellon University*

John M. Dolan  
*Carnegie Mellon University*

Pradeep Khosla  
*Carnegie Mellon University*

Follow this and additional works at: <http://repository.cmu.edu/isr>

---

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Institute for Software Research by an authorized administrator of Research Showcase @ CMU. For more information, please contact [research-showcase@andrew.cmu.edu](mailto:research-showcase@andrew.cmu.edu).

# Optimal Sensor Placement for Cooperative Distributed Vision

Luis E. Navarro-Serment<sup>1</sup>, John M. Dolan<sup>2</sup>, and Pradeep K. Khosla<sup>1,2</sup>

<sup>1</sup> Department of Electrical and Computer Engineering

<sup>2</sup> The Robotics Institute

Carnegie Mellon University

Pittsburgh, PA 15213, USA

{lenscmu, jmd, pkk}@cs.cmu.edu

**Abstract**— This paper describes a method for observing maneuvering targets using a group of mobile robots equipped with video cameras. These robots are part of a team of small-size (7x7x7 cm) robots configured from modular components that collaborate to accomplish a given task. The cameras seek to observe the target while facing it as much as possible from their respective viewpoints. This work considers the problem of scheduling and maneuvering the cameras based on the evaluation of their current positions in terms of how well can they maintain a frontal view of the target. We describe our approach, which distributes the task among several robots and avoids extensive energy consumption on a single robot. We explore the concept in simulation and present results.

**Keywords**— sensor placement; cooperative sensors; distributed vision; automatic surveillance.

## I. INTRODUCTION

Surveillance, reconnaissance, and security are good examples of difficult or tedious undertakings that can benefit from the use of teams of robotic sensors capable of modifying their vantage points. A team of robots has distinct advantages over single robots with respect to sensing. Team members exchange sensor information, collaborate to track and identify targets, or even assist each other to overcome obstacles. By coordinating its members, a team can exploit information derived from multiple disparate viewpoints. A single robot, though potentially equipped with a large array of different sensing modalities, is limited at any one time to a single viewpoint. Moreover, a team of robots can simultaneously collect information from multiple locations.

One important aspect of surveillance and vigilance tasks that is frequently addressed is that of observing and tracking the movements of people or vehicles navigating within the confines of a particular area. In the case of observing people, it is often important not only to detect and track a person, but also to identify him/her, or at least to produce images that could be used in the future to determine whether a person was present in a particular area at any given time/date.

The correct placement of a video camera with respect to the subject or target is crucial to produce a useful image. A good photographer always tries to capture a scene from the best possible angle. There are several factors to be considered when evaluating how good a particular camera position is in terms of the characteristics of the image that will be produced. For

example, consider a conventional mug shot made for police files. If the suspect is located either too close to or too far away from the camera, distinctive characteristics of the suspect's face can be lost in the picture. In the same manner, it is important that the suspect faces the camera at certain angles for both the frontal and profile shots.

Research in this area typically considers static placement of sensors, and deals with issues related to polygon visibility [7]. Some other researchers have considered the use of sensors that shift their positions over time to ensure that targets remain under surveillance at all times [1][2]. However, most of these approaches do not take into account that some applications require the sensor to be positioned with a particular orientation, and offer solutions in which just the detection of a target is considered satisfactory.

In this paper we describe some preliminary work on an automatic target observation system to be used primarily by our team of centimeter-scale robots, called Millibots [4]. The Millibots are capable of carrying different types of sensors while maintaining position estimates, and currently are able to collaborate to explore and map areas inaccessible to larger robots. We are currently developing a system in which video cameras seek to observe the target while frontally facing it as much as possible from their respective viewpoints for surveillance tasks. The same system can be used for other applications that require the observation of a moving target for different facing angles.

## II. PROBLEM DESCRIPTION

The *Optimal Sensor Placement* problem studied in this paper is defined as follows: given

**S**: A finite two-dimensional spatial region, with entrances and exits.

**R**: A subset of  $M$  robots equipped with video cameras, deployed as part of a larger set of  $N$  robots  $\mathbf{R}_T$ .

$$\mathbf{R} = \{r_1, r_2, \dots, r_M\}$$

**T**: A set of  $K$  targets navigating within **S**.

$$\mathbf{S} = \{t_1, t_2, \dots, t_K\}$$

Define the evaluation function  $E(\mathbf{x}_{r_i}(t), \mathbf{x}_{t_i}(t))$  with range  $[0,1]$ , where  $\mathbf{x}_{r_i}(t)$  and  $\mathbf{x}_{t_i}(t)$  are state vectors that represent the position and orientation of robot (or camera)  $r_i$  and target  $t_i$  respectively. This evaluation function takes into account the camera's optical configuration (i.e., focal length, field of view, etc.), the desired image size of the target, the distance from the camera to the target, and the angle between the target's facing direction and the view direction of the camera.

A robot  $r_i \in \mathbf{R}$  is optimally placed with respect to the target  $t_i \in \mathbf{T}$  at time  $t$  when the evaluation function  $E(\mathbf{x}_{r_i}(t), \mathbf{x}_{t_i}(t)) = 1$ . The goal is to maximize the mean value of the evaluation function  $E(\mathbf{x}_{r_i}(t), \mathbf{x}_{t_i}(t))$  over time through the adequate selection and maneuvering of the set of robots  $r_i \in \mathbf{R}, i = 1, \dots, M$  over time steps  $\Delta T$ .

### III. RELATED WORK

The work most closely related to ours involves the concept of active vision, which implies computer vision implemented with a movable camera. Several researchers have explored the use of multiple cameras for cooperative multisensor surveillance; Collins et al. [8] describe a multicamera surveillance system that allows a single human operator to monitor activities in a cluttered environment using a distributed network of active video sensors. Matsuyama et al. [7] investigate the use of evaluation functions for planning layout of multiple cameras in 2D static scenes. Their work assumes that the cameras remain in fixed positions; the sensor placement problem is solved in the off-line camera work planning. This work differs from ours in that our robots must dynamically shift their positions over time to ensure a frontal view of the target. Parker et al [2] describe the use of multiple robots for cooperative observation of targets. However, this work assumes robots with  $360^\circ$  field of view sensors, and does not enforce any particular sensor-target arrangement.

### IV. APPROACH

The rationale behind our strategy is simple: the robots will remain operational for as long as their energy sources can supply the power required. It is well known that one of the most energy-consuming tasks that any mobile robot can perform is to move from one place to another. Consequently, in order to conserve energy, we want the robots to shift their positions as little as possible. For this reason, it is crucial to know exactly where the cameras should be placed, in order to use the available energy efficiently.

It is reasonable to assume that the closest robot to the target should be the one assigned to observe it. The shorter the distance a robot has to move, the better in terms of energy. However, once a robot has moved to the optimal observation position, it will very likely become the closest robot to the optimal position most of the time, thus creating a situation in which one robot is permanently assigned to observe the target. Although there are a number of reasons why this situation may be disadvantageous, we are mostly concerned about the

extensive use of a single robot's energy reserve. This is particularly important in the case of the Millibots, since their batteries tend to die faster when they shift their positions continuously. Consequently, we seek to distribute the observation task among multiple robots.

There are two basic problems involved in our approach: 1) finding the optimal sensor placement, and 2) selection of the combination of sensors to be used at any given time. These topics are described in the following sections.

#### A. Sensor Evaluation Function

We have modified the evaluation functions used in [7] and extended the concept from statically to dynamically positioned cameras. All points with an unobstructed view of a target are evaluated with respect to two variables: distance to target, and the angle between the target's facing direction and the view direction of the camera. The distance to target affects the apparent size of the target in the image. The optimal distance to target  $d_{opt}$  is determined beforehand, and it takes into account the optical configuration of the camera, and the size of the target. The desired angle between the target's facing direction and the view direction of the camera is our primary concern, and it is determined by the nature of the application. In this paper we consider the full frontal facing angle to be optimal (i.e., the target is facing straight at the camera), although in some cases the user might wish to observe the target at a different angle.

Let us consider a target  $t_i \in \mathbf{T}$  at time  $t$ . The current target state is described by  $\mathbf{x}_{t_i} = [x_{t_i} \ y_{t_i} \ \phi_{t_i}]^T$ , which includes its current position and facing angle. Similarly, assume that the state vector describing a video camera carried by a robot is given by  $\mathbf{x}_{r_i} = [x_{r_i} \ y_{r_i} \ \theta_{r_i}]^T$ .

The evaluation of  $(x_{r_i}, y_{r_i})$  with respect to its distance to the target is given by

$$E_{dist}(\mathbf{x}_{r_i}(t), \mathbf{x}_{t_i}(t)) = \exp\left(\frac{-1}{2} \left( \frac{\sqrt{(x_{r_i} - x_{t_i})^2 + (y_{r_i} - y_{t_i})^2} - d_{opt}}{\sigma} \right)^2\right) \quad (1)$$

Equation (1) resembles a Gaussian distribution, which peaks when the distance between  $(x_{r_i}, y_{r_i})$  and  $(x_{t_i}, y_{t_i})$  is equal to  $d_{opt}$ . The parameter  $\sigma$  adjusts how severely the deviation from the optimal distance,  $d_{opt}$ , is penalized.

In order to evaluate the angle between the target's facing direction and the view direction of the camera, the view direction that centers the target inside the image frame,  $\theta_{r_i}^{aim}$ , is computed first. This is simply the angle of the unit vector that radiates from the point  $(x_{r_i}, y_{r_i})$  toward the point  $(x_{t_i}, y_{t_i})$ , as

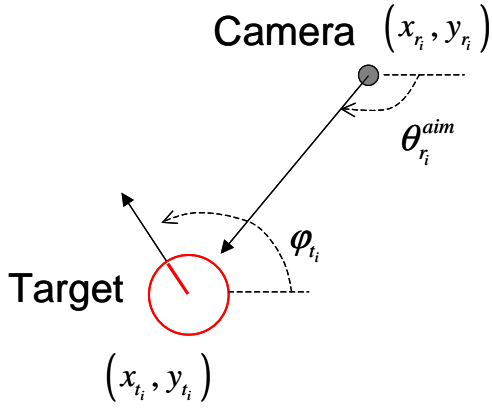


Figure 1. Geometry of the Camera-Target arrangement.

shown in Fig.1. The computation of  $\theta_{r_i}^{aim}$  simplifies the calculation of the optimal view direction, since it reduces the search space for the optimal solution. We make the assumption that the cameras are driven by a holonomic drive mechanism, so they can be oriented equally well along any bearing, notwithstanding the orientation of the robot.

The optimal view direction of the camera occurs when the target is facing straight at the camera. As shown in Fig. 1, when  $\theta_{r_i} = \theta_{r_i}^{aim}$  the angular distance  $\theta_{r_i}^{aim} - \phi_{t_i}$  provides an indication of how well the camera maintains a frontal view of the target. The evaluation of  $(x_{r_i}, y_{r_i})$  with respect to the facing angle is given by

$$E_{facing}(\mathbf{x}_{r_i}(t), \mathbf{x}_{t_i}(t)) = \frac{1}{2} (1 - \cos(\theta_{r_i}^{aim} - \phi_{t_i})) \quad (2)$$

Equation (2) has a minimum when  $\theta_{r_i}^{aim} = \phi_{t_i}$ , i.e., when both the camera and the target are pointing in the same direction. On the other hand, (2) peaks when the angular distance equals  $\pi$  (or  $-\pi$ ), when the camera and target point at one another.

The total evaluation is then computed by multiplying the effects of both evaluation functions (1) and (2):

$$E_{total}(\mathbf{x}_{r_i}(t), \mathbf{x}_{t_i}(t)) = E_{dist} \cdot E_{facing} \quad (3)$$

The optimal camera position for a target can then be determined by finding the position in  $\mathbf{S}$  that maximizes the evaluation function:

$$\mathbf{x}_{opt} = \arg \max_{(x_r, y_r)} E_{total} \Big|_{\theta_{r_i} = \theta_{r_i}^{aim}} \quad (4)$$

Fig. 2 illustrates the use of (3) for evaluating all points surrounding a target. As expected, the optimal camera position is indicated by a global maximum located in front of the target, at a distance  $d_{opt}$ .

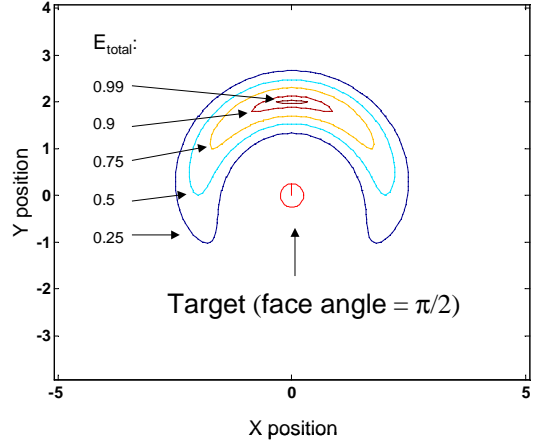


Figure 2. This plot illustrates the performance of the evaluation function. In this plot,  $d_{opt} = 2$  and  $\sigma = 0.4$ . The evaluation function peaks at the point (0,2), which indicates the optimal camera position.

By multiplying the total evaluation function (3) computed individually for each target, it is possible to determine the optimal placement of a camera with respect to multiple targets:

$$E_{combined} = \prod_{K \in List} E_{total_K}, \quad (5)$$

where *List* denotes a list of the objects to be observed by the camera. For example, consider the case in which one camera is required to observe two targets, as shown in Fig. 3. The optimal camera position is computed for each target separately, and then the product of the individual evaluation functions is calculated. As expected, there is a point in the (x, y) plane that maximizes the evaluation function for both targets. However, it is interesting to note that this point, though it produces the best possible view of two targets at the same time, might not produce a useful picture of both targets (for instance, in Fig. 3 the optimal camera position was evaluated at only 87%).

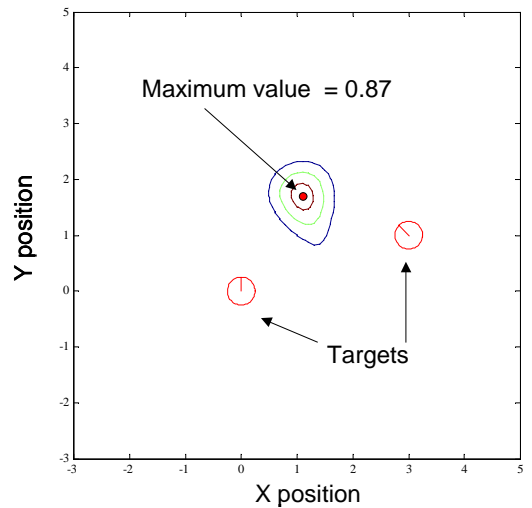


Figure 3. One camera observing two targets. The best image produced is evaluated as 87% of the maximum value.

Nonetheless, the value of this combined evaluation function is still a useful indicator that could be used to decide whether one or two cameras should be scheduled to observe both targets.

### B. Strategy for Sensor Selection

In this section, an algorithm for sensor selection and scheduling is presented. As mentioned before, the goal is to maximize the mean value of the evaluation function  $E(\mathbf{x}_{r_i}(t), \mathbf{x}_{t_i}(t))$  over time through the adequate selection and maneuvering of robots  $r_i \in \mathbf{R}, i = 1, \dots, M$  over time steps  $\Delta T$ , while reducing the collective use of battery power.

The evaluation function (3) is used to find the optimal camera position, and plays a central role in our algorithm.

The algorithm is the following: Given  $\mathbf{R}$ ,  $\mathbf{S}$ , and  $\mathbf{T}$ , and the operation parameters  $d_{opt}, \sigma, \Delta T$ , execute the following steps every  $\Delta T$  seconds:

1. Obtain an estimate of  $\mathbf{x}_{t_i}$  for all  $t_i \in \mathbf{T}$  navigating in  $\mathbf{S}$  at time  $\Delta T \cdot t$ . (This is usually done using other robots in  $\mathbf{R}_T$ ).
2. Compute optimal camera positions for all  $t_i \in \mathbf{T}$ , using (4). If an optimal camera position cannot be found (i.e. no single global maximum is present), or the maximum evaluation number is below a certain threshold (i.e., the view is not good enough), increase the number of cameras to observe the targets by one until a set of optimal positions is found.
3. Find the sensor  $r_i \in \mathbf{R}$  that is closest (Euclidean distance) to each optimal position found in step 2, and assign it. If there is a conflict (i.e. one robot is closest to two or more targets that cannot be observed simultaneously), the next closest robot is assigned.
4. Maneuver all assigned sensor into their corresponding targets.

The next section presents simulations that illustrate the performance of this algorithm.

## V. SIMULATIONS

We have conducted a series of simulation experiments to validate our ideas. The objective is to illustrate the computation of optimal camera position using evaluation functions. Our experiments were performed using a Millibot simulator. This simulator has been extensively used by our team, and resembles the most relevant characteristics of the Millibots with a good degree of fidelity. It is assumed that all sensors know their positions at all times, as the Millibots do. The Appendix describes the Millibots and their localization system in detail. The tracking controller used is described in [6].

In the first simulation the goal is to maintain a frontal view of both objects as much as possible, using as few cameras as possible. This experiment involves two objects moving at constant but different speeds, as shown in Fig. 4. Two series of big circles denote the position and orientation of the two objects at every time step. The relative distance between consecutive circles on each object's path is different, since the objects move at different velocities. Two cameras are deployed in the area. Both cameras can dynamically shift their positions. The square marks indicate the initial position of the cameras. The series of small circles moving from left to upper right indicates the trajectory of the optimal camera position, if a single camera were used to observe the two objects simultaneously for the duration of the experiment. This optimal position is computed using (5). As shown in the top plot of Fig. 5, during the first 5 seconds the optimal observation position is not able to produce an adequate image of both objects. The evaluation number remains below 0.5, which was set as the threshold value. Consequently, camera 1 is initially assigned to observe Object 2, while Object 1 is observed by camera 2. The evaluation of the optimal camera placement eventually reaches the threshold after 5 seconds, indicating that a camera is able by itself to maintain an adequate frontal view of both objects simultaneously. At that time, camera 1 is the closest to the optimal camera position and consequently is assigned to observe both objects, while camera 2 is released and stopped. Both camera evaluations are shown in Fig. 5, bottom plot.

The second simulation is similar. The goal is also to maintain a frontal view of both objects as much as possible, using as few cameras as possible. This experiment involves two objects moving from left to right at constant but different speeds, as shown in Fig. 6. Two series of big circles denote the position and orientation of the two objects at every time step. The relative distance between consecutive circles on each object's path is different, since the objects move at different velocities. Eight cameras are deployed along an imaginary hallway. This time, the cameras can dynamically shift their positions. The square marks indicate the initial position and view direction of the cameras. The series of small circles moving from left to right indicate the trajectory of the optimal camera position, if a single camera were used to observe the two objects simultaneously for the duration of the experiment.

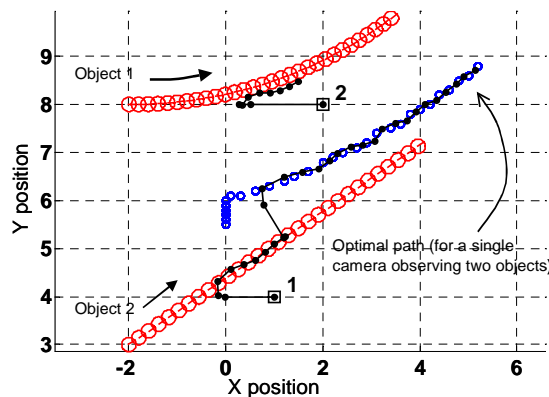


Figure 4. Observing multiple objects. Initially the optimal observation position is not able to produce an adequate image of both objects. Eventually, camera 1 is assigned to observe both objects.

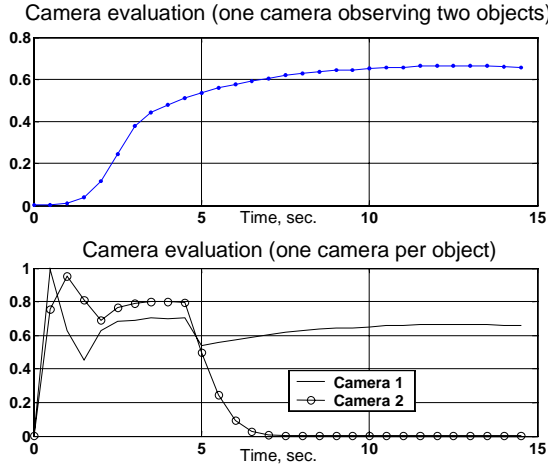


Figure 5. Camera evaluation for multiple objects. The top plot shows the evaluation of one camera observing both targets. When the optimal position is able to produce a satisfactory image, camera 1 observes both objects, and camera 2 is released.

This optimal position is computed using (5). Initially, camera 1 is able by itself to maintain an adequate frontal view of both objects simultaneously. This is observed in the trajectory of camera 1, which initially follows the trajectory of the optimal camera position. Due to the difference in speeds, the optimal observation position is no longer able to produce a satisfactory image of both objects and the evaluation number drops below 0.5, which was set as the threshold value. Consequently, camera 6 is called into action after 5 seconds. The results are shown in Fig. 7.

It is interesting to observe in Fig.7 that the evaluation function for camera 1 reaches a steady state below the maximum of 1. This is due to the fact that the position controller used for tracking fails to keep up the pace and lags with respect to the desired position. This resembles the behavior observed in the real robots. It is also interesting to

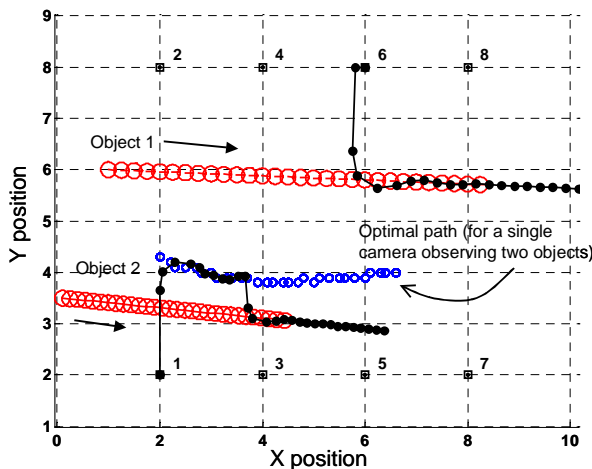


Figure 6. Observing multiple objects. Camera 1 is initially able to observe both objects simultaneously. When the evaluation of camera 1 dropped below the threshold, a second camera was assigned to observe Object1, while Camera 1 was assigned to observe Object 2.

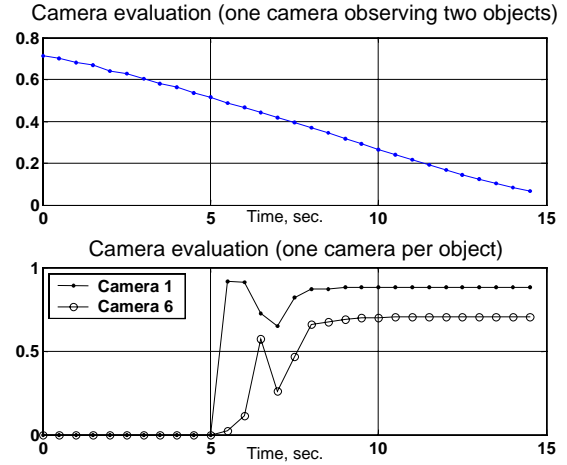


Figure 7. Camera evaluation for multiple objects. The top plot shows the evaluation of camera 1 observing both targets. When the optimal position was not able to maintain a minimum evaluation, a second camera is called in, resulting in good observation of both objects.

note that the handoff transient overshoots first, and then falls before reaching the steady state. This is again due to the behavior of the tracking controller, while the robot maneuvers into the observing position.

## VI. CONCLUSIONS AND FURTHER WORK

We have described a method for observing maneuvering targets using a group of mobile robots equipped with video cameras. We proposed an algorithm for scheduling and maneuvering the cameras based on the evaluation of their current positions in terms of how well can they take a frontal picture of the target. We presented simulation results that seem to indicate the potential of the suggested approach.

The experiments described in the preceding section are preliminary. The simulations presented do not account for other obstacles in the environment, which most likely will affect the evaluation function of points in the plane. In addition, although the algorithm is scalable, we need to further test the performance for several camera-target combinations, and for different target maneuvering indices.

There are a number of directions in which we would like to expand this research. In order to be effective, the algorithm should incorporate mechanisms for navigation (i.e. collision avoidance, etc). Furthermore, the algorithm could benefit from learning about the most transited paths in the region, so that the sensors would gravitate towards high-target-density areas.

## APPENDIX

### A. The Millibots and the localization system

The Millibots are small robots (7x7x7 cm) configured from modular components. This modular design allows the team to be easily configured to accomplish the task at hand. The current sensor set includes RF communication, computation,

mobility, video camera, sonar (short and long range), IR rangefinder, and PIR sensor modules. Because of their small size, the computational and sensing capabilities of Millibots are limited [4]. Nonetheless, by exploiting the properties of specialization and collaboration the disadvantages imposed by small robot size are overcome. By having specialized sensor modules the team as a whole can optimize on size and resources - specifically power. A sensor is activated only when the mission requires it. Collaboration is necessary to coordinate and collect information so that the individual members of the team can act as a single logical entity.

A custom-built localization system that utilizes dead reckoning and ultrasonic distance measurements between robots enables the Millibots to accomplish tasks such as mapping and exploration [3]. The distance between two robots is measured using synchronized ultrasound and RF pulses. A conical reflector mounted above a low-cost transducer allows the Millibots to detect and transmit ultrasonic pulses in any direction. Periodically, each robot that serves as a beacon emits simultaneously a radio frequency (RF) pulse and an ultrasonic pulse. Using the RF pulse for synchronization, the distance to the beacon is measured as the time-of-flight of the ultrasonic pulse multiplied by the speed of sound (343m/s at 20°C). The team leader coordinates the pinging sequence to ensure that beacon signals from multiple robots do not interfere with one another. To improve the accuracy, this procedure is repeated several times and the sample mean is utilized to estimate the distance to the beacon.

All the Millibots transmit their distance measurements to the team leader who calculates the new robot positions. A maximum likelihood algorithm determines the most likely position of the robot given the measured distances to the current beacons. Assuming that the dead reckoning and distance measurements are normally distributed random variables, the likelihood of being located at a position  $(x,y)$  is given by

$$P(x, y | x_d, y_d, \bar{r}_b(1), \bar{r}_b(2), \dots, \bar{r}_b(m)) = N(x - x_d, \sigma_x^2) N(y - y_d, \sigma_y^2) \prod_{i=1}^m N(r(i) - \bar{r}_b(i), \sigma_b^2) \quad (6)$$

where  $N(p, \sigma^2)$  is a normal distribution with zero mean and variance of  $\sigma^2$  evaluated at  $p$ ,  $(x_d, y_d)$  is the position measured

through dead reckoning,  $r(i)$  is the distance from the beacon  $i$  to the Millibot,  $m$  is the number of beacons, and  $\bar{r}_b(i)$  is the sample mean of the distance measurements from beacon  $i$  to the Millibot.

The problem of fault tolerant localization for the Millibots is addressed in [5]. This work focuses on detecting and isolating measurement faults that commonly occur in this localization system. Such failures include dead reckoning errors when the robots collide with undetected obstacles, and distance measurement errors due to destructive interference between direct and multi-path ultrasound wavefronts.

## REFERENCES

- [1] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02) Fukuoka, Japan, June 25-27, 2002.
- [2] L. E. Parker and B. Emmons, "Cooperative multi-robot observation of multiple moving targets," In Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pp. 2082-2089, Albuquerque, New Mexico, April 1997.
- [3] L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla, "A beacon system for the localization of distributed robotic teams," International Conference on Field and Service Robotics, Pittsburgh, PA pp. 232-237, 1999.
- [4] R. Grabowski, L. E. Navarro-Serment, C.J.J. Paredis, and P. K. Khosla, "Heterogeneous teams of modular robots for mapping and exploration," Autonomous Robots, vol. 8, pp. 293-308, 2000.
- [5] R. Tinos, L.E. Navarro-Serment, and C.J.J. Paredis, "Fault tolerant localization for teams of distributed robots," Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Page(s): 1061 -1066 vol.2, 2001.
- [6] S.O. Lee, Y.J. Cho, M.Hwang-Bo, B.J. You, and S.R. Oh, "A stable target-tracking control for unicycle mobile robots," Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1822-1827, 2000.
- [7] T. Matsuyama, T. Wada, and S. Tokai, "Active image capturing and dynamic scene visualization by cooperative distributed vision," Proc. 1st International Conference on Advanced Multimedia Content Processing AMCP98, pp.256-293, 1998.11
- [8] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," Proceedings of the IEEE, Vol. 89, No. 10, October, 2001, pp. 1456 - 1477.