Cooperative Multi-Robot Observation of Multiple Moving Targets

Lynne E. Parker[1] and Brad A. Emmons[2]
[1] Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6364
[2] Department of Mathematics, Albion College, Albion, Michigan 49224

Abstract

An important issue that arises in the automation of many security, surveillance, and reconnaissance tasks is that of monitoring, or observing, the movements of targets navigating in a bounded area of interest. A key research issue in these problems is that of sensor placement — determining where sensors should be located to maintain the targets in view. In complex applications of this type, the use of multiple sensors dynamically moving over time is required. In this paper, we investigate the use of a cooperative team of autonomous sensor-based robots for multi-robot observation of multiple moving targets. We focus primarily on developing the distributed control strategies that allow the robot team to attempt to maximize the collective time during which each object is being observed by at least one robot in the area of interest. Our initial efforts in this problem address the aspects of distributed control in homogeneous robot teams with equivalent sensing and movement capabilities working in an uncluttered, bounded area. This paper first formalizes the problem, discusses related work, and then shows that this problem is NP-hard. We then present a distributed approximate approach to solving this problem that combines low-level multi-robot control with higher-level control. The low-level control is described in terms of force fields emanating from the targets and the robots. The higher level control is presented in the ALLIANCE formalism, which provides mechanisms for fault tolerant cooperative control, and allows robot team members to adjust their low-level actions based upon the actions of their teammates. We then present the results of the implementation of portions of our approach, both in simulation and on physical robots.

1 Introduction

An important issue that arises in the automation of many security, surveillance, and reconnaissance tasks is that of monitoring, or observing, the movements of targets navigating in a bounded area of interest. A key research issue in these problems is that of sensor placement — determining where sensors should be located to maintain the targets in view. In the simplest version of this problem, the number of sensors and sensor placement can be fixed in advance to ensure adequate sensory coverage of the area of interest. However, in more complex applications, a number of factors may prevent fixed sensory placement in advance. For example, there may be little prior information on the location of the area to be monitored, the area may be sufficiently large that economics prohibit the placement of a large number of sensors, or the area may not be physically accessible in advance of the mission. In the general case, the combined coverage capabilities of the available robot sensors will be insufficient to cover the entire terrain of interest. Thus, the above constraints in an application require the use of multiple sensors dynamically moving over time.

In this paper, we investigate the use of a cooperative team of autonomous sensor-based robots for applications in this domain. We focus primarily on developing the distributed control strategies that allow the team to attempt to minimize the total time in which targets escape observation by some robot team member in the area of interest. Of course, many variations of this dynamic, distributed sensory coverage problem are possible. For example, the relative numbers and speeds of the robots and the targets to be tracked can vary, the availability of inter-robot communication can vary, the robots can differ in their sensing and movement capabilities, the terrain may be either enclosed or have entrances that allow objects to enter and exit the area of interest, and so forth. Many other subproblems must also be addressed, including the physical tracking of targets (e.g. using vision, sonar, IR, or laserrange), prediction of target movements, multi-sensor fusion, and so forth. Thus, while our ultimate goal is to develop distributed algorithms that address all of these problem variations, we first focus on the aspects of distributed control in homogeneous robot teams with equivalent sensing and movement capabilities working in an uncluttered, bounded area.

Section 2 defines the multitarget observation problem of interest in this paper, followed by a discussion of related work in section 3. We then note the difficulty of the defined problem in section 4. Section 5 describes our approach, discussing each of the subcomponents of the system. Section 6 describes the implementation of portions of our approach on both a simulated and a physical robot team of four Nomad robots. Finally, we offer concluding remarks in section 7, as well as directions of continuing and future research.

2 Problem Description: *CMOMMT*

The cooperative multi-robot observation of multiple moving targets (or *CMOMMT* for short) problem is defined as follows. Given:

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

S : a two-dimensional, simple polyhedral spatial region, with entrances/exits

$\mathcal{R}$: a team of M robots with 360^0 field of view observation sensors, noisy and of limited range

$\mathcal{O}(t)$: a set of N targets, $o_j(t)$ such that $In(o_j(t), S)$ is true (where $In(o_j(t), S)$ means that target $o_j(t)$ is located within region S at time t)

Define an $M \times N$ matrix $A_{ij}(t)$, where

$$a_{ij}(t) = \begin{cases} 1 & \text{if robot } r_i \text{ is monitoring target } o_j(t) \\ & \text{in } S \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

We further define the *logical OR* operator over a vector H of k elements as:

$$\bigvee_{i=1}^{k} h_i = \begin{cases} 1 & \text{if there exists an } i \text{ such that } h_i = 1 \\ 0 & \text{otherwise} \end{cases}$$

We say that a robot is *monitoring* a target when the target is within that robot's observation sensory field of view. Then, the goal is to maximize:

$$\sum_{t=0}^{T} \sum_{j=1}^{N} \bigvee_{i=1}^{M} a_{ij}(t)$$

over time steps Δt under the assumptions listed below. In other words, the goal of the robots is to maximize the collective time during which each object in S is being monitored by at least one robot during the mission from $t = 0$ to $t = T$. Note that we do not assume that $\mathcal{O}(t)$'s membership is known in advance.

In addressing this problem, we assume the following: Define *sensor_coverage*(r_i) as the area visible to robot r_i's observation sensors, for $r_i \in \mathcal{R}$. Then we assume that, in general,

$$\bigcup_{r_i \in \mathcal{R}} sensor\_coverage(r_i) \ll S.$$

That is, the maximum area covered by the observation sensors of the robot team is much less than the total area to be monitored. This implies that fixed robot sensing locations or sensing paths will not be adequate in general, and that, instead, the robots must move dynamically as targets appear in order to maintain observational contact with them and to maximize the coverage of the area S.

We further assume the following:

- The robots have a broadcast communication mechanism with bandwidth of order $O(mn)$ (for m robots and n targets) that allows them to talk with each other within the area S.

- For all $r_i \in \mathcal{R}$ and for all $o_j(t) \in \mathcal{O}(t)$,

$$max\_v(r_i) > max\_v(o_j(t)),$$

where $max\_v(a)$ returns the maximum possible velocity of entity a, for $a \in \mathcal{R} \cup \mathcal{O}(t)$.

- Objects in $\mathcal{O}$ can enter and exit region S through distinct entrances on the boundary of S.

- The robot team members share a known global coordinate system.

In the general case, the observation sensor on each robot is of limited range and is directional (e.g., a camera), and can only be used to observe targets within that sensor's field of view. However, to simplify the problem initially, we report here the results of the case of an omni-directional 2D sensory system (such as a ring of cameras or sonars), in which the robot sensory system is of limited range, but is available for the entire $360°$ around the robot, as depicted in figure 1.

3 Related Work

Research related to the multiple target observation problem can be found in a number of domains, including art gallery and related problems, multitarget tracking, and multi-robot surveillance tasks. While a complete review of these fields is impossible in a short paper, we will briefly outline the previous work that is most closely related to the topic of this paper.

The work most closely related to the *CMOMMT* problem falls into the category of the *art gallery* and related problems [11], which deal with issues related to polygon visibility. The basic art gallery problem is to determine the minimum number of guards required to ensure the visibility of an interior polygonal area. Variations on the problem include fixed point guards or mobile guards that can patrol a line segment within the polygon. Most research in this area typically utilizes centralized approaches to the placement of sensors, uses ideal sensors (noise-free and infinite range), and assumes the availability of sufficient numbers of sensors to cover the entire area of interest. Several authors have looked at the static placement of sensors for target tracking in known polygonal environments (e.g., [4]). These works differ from the *CMOMMT* problem, in that our robots must dynamically shift their positions over time to ensure that as many targets as possible remain under surveillance, and their sensors are noisy and of limited range.

Sugihara *et al.* [15] address the *searchlight scheduling problem*, which involves searching for a mobile "robber" (which we call *target*) in a simple polygon by a number of fixed searchlights, regardless of the movement of the target. Their objective is to determine whether a search schedule exists, given a polygon and the locations of the searchlights. They develop certain necessary and sufficient conditions for the existence of a search schedule in certain situations, under the assumption of a single target, no entrances/exits, and fixed searcher positions.

Suzuki and Yamashita [16] address the *polygon search* problem, which deals with searching for a mobile target in a simple polygon by a single mobile searcher. They examine two cases: one in which the searcher's visibility is restricted to k rays emanating from its position, and one in which the searcher can see in all directions simultaneously. Their work assumes that the searcher has an infinite sensory range, that the target cannot enter or exit the polygon after
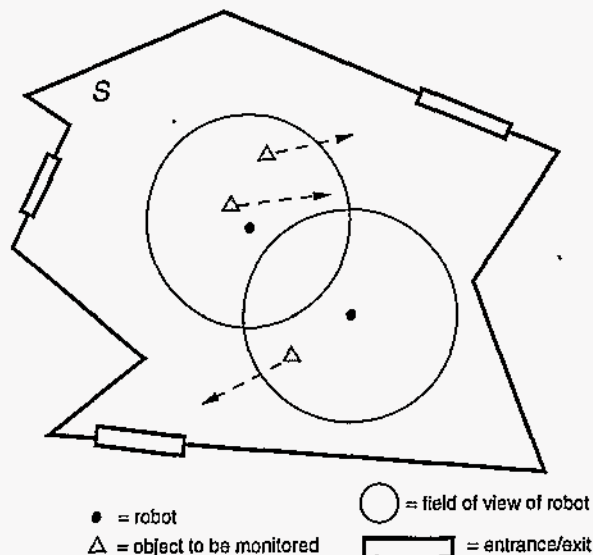
Figure 1: The problem depicted in terms of omni-directional 2D robot sensors.

- = robot
△ = object to be monitored
◯ = field of view of robot
▭ = entrance/exit

the start of the problem, and that only one searcher is available.

Another large area of related research has addressed the problem of multitarget tracking (e.g. [1, 3, 2, 7]). This problem is concerned with computing the trajectories of multiple targets by associating observations of current target locations with previously detected target locations. In the general case, the sensory input can come from multiple sensory platforms. Our task in this paper differs from this work in that our goal is not to calculate the trajectories of the targets, but rather to find dynamic sensor placements that minimize the collective time that any target is not being monitored (or observed) by at least one of the mobile sensors.

Finally, a number of authors have developed systems for multi-robot surveillance (e.g., [5, 6, 17]). However, this related research in multi-robot surveillance does not deal with the issue of interest in this paper — the dynamic placement of mobile sensors in areas in which targets may enter and exit.

4   Problem Difficulty

The CMOMMT problem outlined above is a difficult problem; we now show that it is NP-hard. Let us consider the simpler, static, problem of determining the optimal placement of robots at one instant in time, given that we know the location of all targets. The goal is to determine the optimal placement of the robots to ensure that the maximum number of targets lie within the sensory range of some robot. We assume that the robots have homogeneous sensing capabilities, and that the targets are stationary. We can then show NP-hardness by reduction to the well-known NP-complete problem VERTEX COVER [8]. We sketch the idea of the proof below.

The VERTEX COVER problem is as follows: given a graph $G = (V, E)$ and a positive integer $K \leq |V|$, determine whether there is a vertex cover of size K or less for G; that is, a subset $V' \subseteq V$ such that $|V'| \leq K$ and, for each edge $\{u, v\} \in E$, at least one of u and v belongs to V. We then have the following:

Theorem 1 *The CMOMMT (cooperative multi-robot observation of multiple moving targets) problem is NP-hard in the number of targets and the number of robots.*

Proof Sketch:
By reduction to VERTEX COVER: Let P_i and P_j be the Cartesian (x, y) point locations of targets $o_i(t)$ and $o_j(t)$, and d be the sensor range of the robots. Define:

$$\begin{aligned} V &= \{P_k \mid o_k(t) \in O(t)\} \\ E &= \{\{P_i, P_j\} \mid \text{ the distance between } o_i(t) \\ &\quad \text{ and } o_j(t) \text{ at time } t \text{ is less than } 2d \\ &\quad \text{ for } o_i(t) \in O(t)\} \end{aligned}$$

Then it can be shown that the CMOMMT problem is reducible to the VERTEX COVER problem, where $G = (V, E)$ as defined above, and $K = M$, where M is the number of robots. The solution to VERTEX COVER, V', would give the optimal positions for the M robots in CMOMMT.   □

Thus, since finding the optimal solution is computationally prohibitive, we instead investigate an approximate solution.

5   Approach

Figure 2 shows the overall design of the control system within each robot team member. This design is based upon the ALLIANCE architecture [12, 13], which facilitates the fault tolerant cooperative control of multiple robot teams. We now provide a brief overview of ALLIANCE, and then describe how we use this approach to develop the overall control system for robots performing the CMOMMT application.

The ALLIANCE software architecture is a behavior-based, fully distributed architecture that utilizes adaptive action selection to achieve fault tolerant cooperative control. Robots under this architecture possess a variety of high-level functions (modeled as *behavior sets*) that they can perform during a mission, and must at all times select an appropriate action based on the requirements of the mission, the activities of other robots, the current environmental conditions, and their own internal states. Since cooperative robotic teams often work in dynamic and unpredictable environments, this software architecture allows the team members to respond robustly and reliably to unexpected environmental changes and modifications in the robot team that may occur due to mechanical failure, the learning of new skills, or the addition or removal of robots from the team by human intervention. This is achieved through the interaction of mathematically modeled motivations of behavior, such as impatience and acquiescence, within each individual robot. These motivations allow robots to take over tasks from other team members if those team members do not demonstrate their ability — through their effect on the world — to accomplish those tasks. Similarly, it allows a robot to give up its own current task if
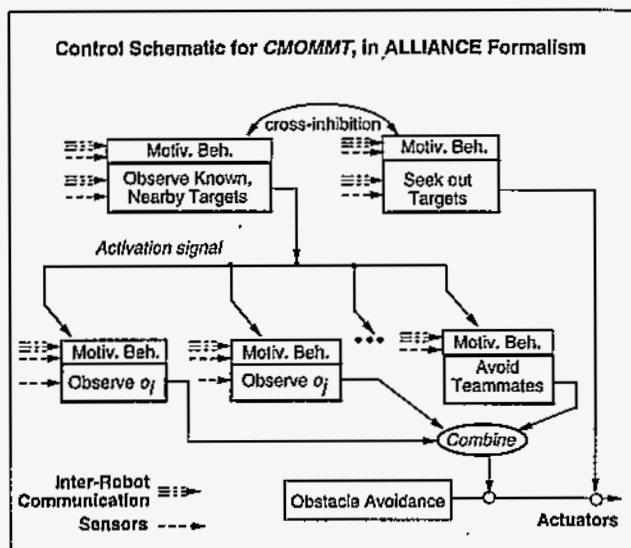
Figure 2: Control within an individual robot for the *CMOMMT* mission, in the ALLIANCE formalism.

its sensory feedback indicates that adequate progress is not being made to accomplish that task. The primary mechanism for achieving adaptive action selection in this architecture is the *motivational behavior*. The output of a motivational behavior is typically the activation level or importance weighting of its corresponding behavior set, represented as a non-negative number. The current level of activation controls the output of its corresponding behavior set.

In the *CMOMMT* problem, each robot has two highest-level behavior sets: "Observe Known, Nearby Targets" and "Seek Out Targets". The "Observe Known, Nearby Targets" behavior set controls the robot's movement in relationship to other nearby robots and nearby targets. It is responsible for ensuring that the current robot maintains a viewpoint on the proper targets, dependent upon other robots' movements and the positions of the targets. The "Seek Out Targets" behavior set allows robots to intelligently search the area S when no objects are nearby. In figure 2, the motivational behaviors are indicated by the small rectangle attached at the top of the behavior sets.

5.1 Observe Known, Nearby Targets

The "observe known, nearby targets" behavior set is responsible for controlling the current robot's movements to maintain observation of certain targets nearby. These movements within S are based upon the movements of nearby targets and the actions of other robot team members. This part of the control scheme is modeled by a collection of lower-level motivational behaviors, each of which is spawned automatically when a robot has become aware of a target nearby. The motivational behaviors in this subsystem are responsible for determining the weight, or importance, of the current robot's continued monitoring of target o_i. If any target o_j leaves the current robot's predictive tracking range, the corresponding motiva-

tional behavior is terminated. The generated weights are then factored into the output of the "Observe ..." behavior set (described below) to calculate the desired direction of motion of the current robot. This combination of information is modeled in figure 2 as the *combine* module.

Subsections 5.1.1 and 5.1.2 describe how the local control information based upon robot and target locations is derived. Subsection 5.1.3 describes how the motivational behaviors derive the weights corresponding to each target. Subsection 5.1.4 discusses the combination of the lower-level and higher-level information.

5.1.1 Target and Robot Detection

Ideally, we would like robots to be able to passively observe the actions of nearby robots and targets, to ascertain their current positions and velocities. Research fields such as machine vision have dealt extensively with this topic, and have developed algorithms for this type of passive position calculation. However, since the physical tracking and 2D positioning of visual targets is not the focus of this research, we instead assume that robots use a global positioning system (such as GPS for outdoors, or the laser-based MTI indoor positioning system [9], which is in use at our CESAR laboratory) to determine their own position, and communicate this information to other robot team members. In our approach, robots do not store position information for robots that are not relatively close (made explicit below).

In addition to robot position information, team members need to determine the positions and velocities of the targets within their own field of view. Since previous work [10, 14], has shown that communication and awareness of robot team member actions can significantly improve the quality of a distributed solution for certain task domains, we supplement a robot's knowledge of target movements gained from direct sensing (e.g. from its cameras or sonar) with position and derived velocity information on target sightings that is communicated by other robot team members within a given communication range. Thus, targets can be one of two types: directly sensed or "virtually" sensed through predictive tracking. In our approach, each robot communicates to its teammates the positions and velocities of all objects within its own view (i.e. the virtual targets). However, a team member does not store position information for targets that are not within its own vicinity. Note that this approach requires the available communication bandwidth to be $O(mn)$, for m robots and n targets.

To clarify this idea, figure 3 depicts three ranges that are defined with respect to each robot r_i. The innermost range is the *sensing range* of r_i, within which the robot can use a sensor-based tracking algorithm to maintain passive contact with targets in its field of view. The middle range is the *predictive tracking range* of the robot r_i, which defines the range in which targets localized by other robots $r_k \neq r_i$ can affect r_i's movements. The outermost range is the *communication range* of the robot, which defines the extent of the robot's communicated messages. Here, it is assumed
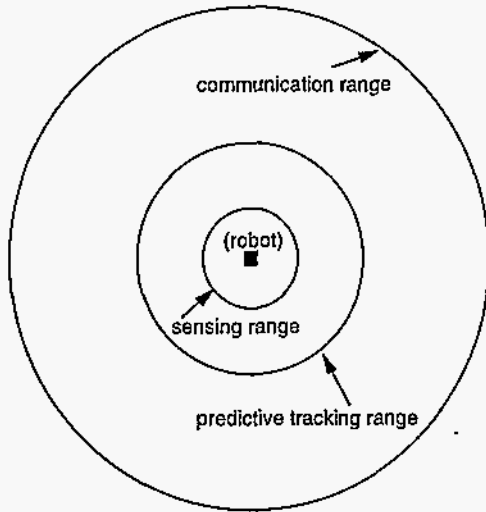
Figure 3: Definition of the *sensing range*, *predictive tracking range*, and *communication range* of a robot. Although the exact range values may change, we assume that the relative ordering of range distances remains the same.
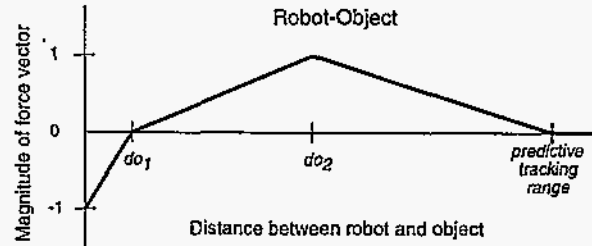

Figure 4: Function defining the magnitude of the force vector of nearby targets.

Figure 5: Function defining the magnitude of the force vector of nearby robots.

that the communication range covers the entire area S. For simplicity, we assume that the sensing, predictive tracking, and communications ranges are the same, respectively, across all robot team members.

When a robot receives a communicated message regarding the location and velocity of a sighted target that is within its predictive tracking range, it begins a predictive tracking of that target's location, assuming that the target will continue linearly from its current state. If the communicated information indicates that a target is within robot r_i's predictive tracking range, that information is held in memory and used as described in the following subsections.

We assume that if the targets are dense enough that their position estimations do not supply enough information to disambiguate distinct targets, then existing tracking approaches (e.g. [2]) should be used to uniquely identify each target based upon likely trajectories.

5.1.2 Local Force Vector Calculation

The local control of a robot team member is based upon a summation of force vectors which are attractive for nearby targets, and repulsive for nearby robots. Figure 4 defines the magnitude of the attractive forces of a target within the predictive tracking range of a given robot. Note that the robot is repelled from a target if it is too close (*distance* $< do_1$) to that target, and it is strongly attracted to the target if the robot is nearby (*distance* $< do_2$). Beyond a given distance do_2, the attraction falls off linearly until the predicted tracking range is met, at which point the attraction goes to 0.

Figure 5 defines the magnitude of the repulsive forces between robots. If the robots are too close together (*distance* $< dr_1$), they repel strongly. If the
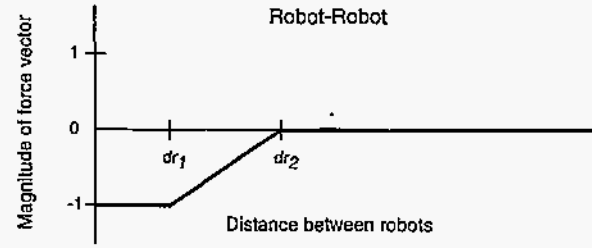
robots are far enough apart (*distance* $> dr_2$), they have no effect upon each other in terms of the force vector calculations. The magnitude scales linearly between these values.

As defined so far, the force vectors are equivalent for all objects, and for all robots. However, we need to inject additional high-level control into the system to take into account more global information. This is modeled as predictive weights that are factored into the force vector calculation, as described in the next subsection.

5.1.3 High-Level Control via ALLIANCE

One problem with using only force vectors is that of local minima. To help resolve some of these problems, we use higher-level control via motivational behaviors to differentially weight the contributions of each target's force field on the total computed field. This higher-level knowledge is expressed in the form of two types of probabilities: the probability that a given target actually exists, and the probability that no other robot is already monitoring a given target. Combining these two probabilities helps reduce the overlap of robot sensory areas toward the goal of minimizing the likelihood of a target escaping detection.

The probability that a target exists is modeled as a decay function based upon when the target was most recently seen, and by whom. In general, the probability decreases inversely with distance from the current robot. Beyond the predictive tracking range of the robot, the probability becomes zero.

The probability that no other robot is already monitoring a nearby target is based upon the target's position and the location of nearby robots. If the target is in range of another robot, then this probability is

generally high. However, we also incorporate a feature of "impatience", as modeled in the ALLIANCE framework, if a nearby robot does not appear to be satisfactorily observing its local targets. This impatience effectively reduces the probability that the other robot is already monitoring nearby targets. In more complex versions of the *CMOMMT* problem, robots could also learn about the viewing capabilities of their teammates, and discount their teammates' observations if that teammate has been unreliable in the past.

The output of the motivational behavior corresponding to a given target is the product of the probability that the target exists and the probability that no other robot is currently monitoring that target. These probabilities have the effect of causing a robot to prefer the observation of certain targets over others.

5.1.4 Combination of Local and Higher-Level Information

The local force vectors are combined with the higher-level information, resulting in the commanded direction of robot movement. This direction of movement for robot r_l is given by:

$$\sum_{j=0}^{N}(FVO_j \times Pr(exists_j) \times Pr(NT_j)) + \sum_{i \neq l} FVR_i$$

where FVO_k is the force vector attributed to object o_k, $Pr(exists_k)$ is the probability that object o_k exists, NT_k is the probability that object o_k is not already being tracked, and FVR_r is the force vector attributed to robot r_r. This movement command is then sent to the robot actuators to cause the appropriate robot movements.

5.2 Seek Out Targets

When a robot does not detect any target nearby, the sum of the force vectors as described above will cause the robot to move away from its robot neighbors and then idle in one location. While this may be acceptable in some applications, in general, we would like to have the robots actively and intelligently seek out potential targets in the area. In [16], Suzuki and Yamashita discuss search schedules for "∞-searchers", and define the derivation of these search schedules for certain special cases. An "∞-searcher" is a mobile searcher that has a 360° infinite field of view. A search schedule for an ∞-searcher is a path through a simple polygonal area that allows the searcher (or robot) to detect a mobile "intruder" (or target), regardless of the movements of the target. While clearly related to the *CMOMMT* problem, this earlier work makes a number of assumptions that do not hold in the *CMOMMT* problem: infinite range of searcher visibility, only a single searcher, only a single target, and an enclosed polygonal area which does not allow any targets to enter or exit the area.

In our future work, we intend to develop an automated process that allows the robots to generate the appropriate search schedule for a given area based upon this earlier work of Suzuki and Yamashita. Our current approach, however, simplifies the task by supplying the robot team members with a human-derived

search path through the area S. In practice, the derivation of the ∞-search schedule by hand through the polygonal areas that define the interiors of most buildings appears to be fairly straightforward under the assumptions of [16]. More challenging is dealing with multiple targets, multiple robots, and entrances/exits in the polygonal area. We leave this task to future work.

Thus, when no targets are detected by a given robot, that robot moves along the search path looking for targets, paying special attention to entrances for the appearance of new targets. To prevent the robot's path from being predictable to a knowledgeable target, the robot randomly selects a direction to traverse at each intersection in the search path. If two robots encounter each other moving in the opposite direction along the search path, they reverse directions. As soon as targets are detected along the search route, the highest level motivational behaviors will switch the robot from seek mode to observe mode.

6 Experiments

Our approach to the cooperative multi-robot monitoring problem has been implemented both in simulation and on a team of four physical robots. The multi-robot simulator in our laboratory allows us to test and debug our algorithms in simulation prior to executing them on the actual robots. The code generated during the simulation can then be ported directly to the robots for experimentation in the "real world" with relatively minor changes.

In the initial phase of research in this problem, which concentrates on the cooperative control issues of distributed tracking, we utilize an indoor global positioning system as a substitute for vision- or range-sensor-based tracking. Under this approach, each target to be tracked is equipped with an indoor global position sensor, and broadcasts its current x,y position via radio to the robots within communication range. Each robot team member is also equipped with a positioning sensor, and can use the targets' broadcast information to determine the relative location of nearby targets.

Figures 6 and 7 illustrate two examples of the implementation of portions of our approach — namely, the local force-field control — on the simulated robots. In these figures, the black points represent targets, and the gray points represent robots.

Figure 6 shows a case where two targets are being tracked by two robots. The first frame begins with the two targets heading towards each other, and each of the robots "following" one of the targets. In the second frame, the objects have passed each other, changing directions in order to avoid each other, and the robots meet in the middle. At this point the repulsive force between the two robots takes precedence and pushes them away from each other, causing them to swap targets. In the final two frames, the robots continue to follow the new targets.

Figure 7 shows a case where the targets stay relatively distributed throughout the simulation. The robots tend to hover around the center of the mass of targets; they keep their distance from one another throughout the simulation, due to the repulsive forces.
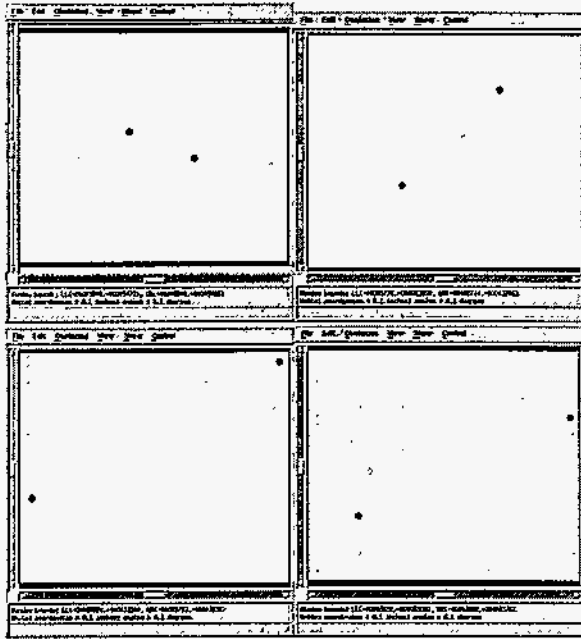
Figure 6: Two targets tracked by two robots performing a swap.

The local control subsystems have been ported to, and successfully demonstrated on, our team of 4 mobile robots. Figure 8 shows an example of the robot implementation. In these experiments, we typically designated certain robots to be targets, and other robots as observers. Since we are not dealing with the issues of visual tracking of objects in our current work, using some robots as targets allowed us to take advantage of the global positioning system on the robots to perform "virtual" tracking. Thus, the robots acting as targets were programmed to broadcast their current location to the robot team; this information could then be used by the observers to calculate their desired movements. We programmed the robots acting as targets to move in one of two ways: movements based on human joystick commands, or simple wandering through the area of interest. In figure 8, the robot targets are indicated by the triangular flags.

The first frame in figure 8 shows the arrangement of the observers and targets at the very beginning of the experiment. The second frame shows how the two observers move away from each other once the experiment is begun, due to the repulsive forces between the observers. In the third frame, a human joysticks one of the robot targets away from the other target and the observers. As the target is moved, the two observers also move in the same direction, due to the attractive forces of the target that is moving away. However, if the target exits the area of interest, S, as illustrated in the fourth frame, then the observers are no longer influenced by the moved target, and again draw nearer to the stationary target, due to its attractive forces. Note that throughout the example, the observers keep away from each other, due to the repulsive forces.

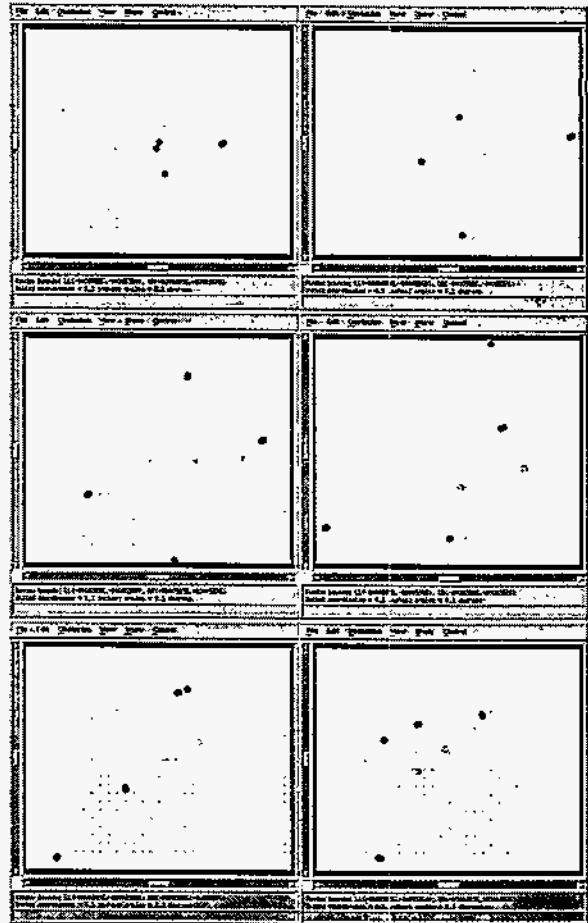In continuing work, we are implementing the entire

Figure 7: When targets stay relatively distributed, the robots hover around the centroid.
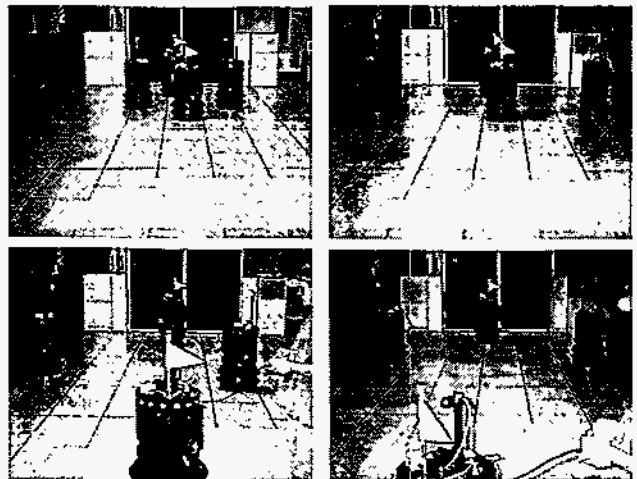

Figure 8: Results of robot team performing task using summation of force vectors. The robots with the triangular flags are acting as targets, while the robots without the flags are performing the distributed observation.

control schematic for *CMOMMT* on both simulated and physical robot teams.

7 Conclusions and Future Work

Many real-world applications in security, surveillance, and reconnaissance tasks require multiple targets to be monitored using mobile sensors. We have shown that the general version of this problem is NP-complete, and thus intractable. We then presented a distributed approach that is based upon high-level control provided through the ALLIANCE formalism, combined with lower-level attractive and repulsive force fields, and a target seeking system. Empirical investigations of the local force-field portion of our proposed cooperative control approach on both simulated and physical robot teams have been presented.

Continuing and future work includes completing the implementation on both the simulated and physical robot teams and the development of an automatic generation of ∞-search schedules. We also desire to prove the effectiveness of our approach, as compared to the optimal solution. Additional related research includes extending the work to apply to polygons with holes, to robots that differ in their sensing and movement capabilities, and addressing the subproblems of the physical tracking of targets (e.g. using vision, sonar, IR, or laserrange) and the prediction of target movements.

8 Acknowledgements

References

[1] Yaakov Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Transactions on Automatic Control*, AC-23(4):618–626, 1978.

[2] Yaakov Bar-Shalom. *Multitarget Multisensor Tracking: Advanced Applications*. Artech House, 1990.

[3] S. S. Blackman. *Multitarget Tracking with Radar Applications*. Artech House, 1986.

[4] Amy J. Briggs. *Efficient Geometric Algorithms for Robot Sensing and Control*. PhD thesis, Cornell University, 1995.

[5] E. H. Durfee, V. R. Lesser, and D. D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C-36:1275–1291, 1987.

[6] H. R. Everett, G. A. Gilbreath, T. A. Heath-Pastore, and R. T. Laird. Coordinated control of multiple security robots. In *Proceedings of SPIE Mobile Robots VIII*, pages 292–305, 1993.

[7] G.C. Fox, R.D. Williams, and P.C. Messina. *Parallel Computing Works*. Morgan Kaufmann, 1994.

[8] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[9] MTI Research Inc. Conac 3-D tracking system. Operating manual, Chelmsford, MA, 1995.

[10] Bruce MacLennan. Synthetic ethology: An approach to the study of communication. In *Proceedings of the 2nd interdisciplinary workshop on synthesis and simulation of living systems*, pages 631–658, 1991.

[11] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[12] Lynne E. Parker. ALLIANCE: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In *Proc. of the 1994 IEEE/RSJ/GI Int'l Conf. on Intelligent Robots and Systems (IROS '94)*, pages 776–783, Munich, Germany, Sept. 1994.

[13] Lynne E. Parker. *Heterogeneous Multi-Robot Cooperation*. PhD thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA, February 1994. MIT-AI-TR 1465 (1994).

[14] Lynne E. Parker. The effect of action recognition and robot awareness in cooperative robotic teams. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '95)*, Pittsburgh, PA, August 1995.

[15] Kzuo Sugihara, Ichiro Suzuki, and Masafumi Yamashita. The searchlight scheduling problem. *SIAM Journal of Computing*, 19(6):1024–1040, 1990.

[16] Ichiro Suzuki and Masafumi Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal of Computing*, 21(5):863–888, 1992.

[17] R. B. Wesson, F. A. Hayes-Roth, J. W. Burge, C. Stasz, and C. A. Sunshine. Network structures for distributed situation assessment. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):5–23, 1981.