CONF-9705121--2

MULTI-ROBOT MOTION CONTROL FOR COOPERATIVE OBSERVATION*

Lynne E. Parker

Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
P.O. Box 2008
Building 6010, MS-6355
Oak Ridge, TN 37831-6355

Submitted to: Fifteenth Symposium on Energy Engineering Sciences, Argonne National Laboratory

* Research sponsored by the Engineering Research Program, Office of Basic Energy Sciences, of the U.S. Department of Energy, under contract DE-AC05-96OR22464 with Lockheed Martin Energy Research, Inc.

DISCLAIMER

DISCLAIMER

MULTI-ROBOT MOTION CONTROL FOR COOPERATIVE OBSERVATION

Lynne E. Parker

Oak Ridge National Laboratory
Oak Ridge, TN 37831-6364

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 involving limited-range sensors, 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 the observation of multiple moving targets. We focus primarily on developing the distributed control strategies that allow the robot team to attempt to minimize the total time in which targets escape observation by some robot team member in the area of interest. This paper first formalizes the problem and discusses related work. We then present a distributed approximate approach to solving this problem that combines low-level multi-robot control with higher-level reasoning control based on the ALLIANCE formalism. We analyze the effectiveness of our approach by comparing it to 3 other feasible algorithms for cooperative control, showing the superiority of our approach for a large class of problems.

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, the available sensor range may be limited, 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 force 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 targets to enter and exit the area of interest, the terrain may be either indoor (and thus largely planar) or outdoor (and thus 3D), and so forth. Many other subproblems must also be addressed, including the physical tracking of targets (e.g. using vision, sonar, IR, or laser range), 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.

The following section defines the multitarget observation problem of interest in this paper, and is followed by a discussion of related work. We then describe our approach, discussing each of the subcomponents of the system. Next, we describe and analyze the results of our approach, compared to three other feasible algorithms for cooperative motion control.. Finally, we offer concluding remarks.

PROBLEM DESCRIPTION

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

$\mathcal{S}$: a two-dimensional, bounded, enclosed spatial region, with entrances/exits

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

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

Define an $m \times n$ matrix $A(t)$, where

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

We further define the *logical OR* operator over a vector H 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 targets in $\mathcal{S}$ are being monitored by at least one robot during the mission from $t = 0$ to $t = T$. Note that we do not assume that the membership of $\mathcal{O}(t)$ 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,

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

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 $\mathcal{S}$.

We further assume the following:

- The robots have a broadcast communication mechanism that allows them to send (receive) messages to (from) each other within the area $\mathcal{S}$.
- 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 $max\_v(a)$ gives the maximum possible velocity of entity a, for $a \in \mathcal{R} \cup \mathcal{O}(t)$.
- Targets in $\mathcal{O}$ can enter and exit region $\mathcal{S}$ through distinct entrances/exits.
- The robot team members share a known global coordinate system.

To somewhat 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^o around the robot.

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 not possible 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 [1], 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. [2]). 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.* [3] 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. 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 to the polygon, and fixed searcher positions

Suzuki and Yamashita [4] 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 no entrances/exits to the polygon and a single searcher.

LaValle *et al.* [5] introduces the visibility-based motion planning problem of locating an unpredictable target in a workspace with one or more robots, regardless of the movements of the target. They define a visibility region for each robot, with the goal of guaranteeing that the target will eventually lie in at least one visibility region. In LaValle *et al.* [6], they address the related question of maintaining the visibility of a moving target in a cluttered workspace by a single robot. They are also able to optimize the path along additional criteria, such as the total distance traveled. The problems they address in these papers are closely related to the problem of interest here. The primary difference is that their work does not deal with multiple robots maintaining visibility of multiple targets, nor a domain in which targets may enter and exit the area of interest.

Another large area of related research has addressed the problem of multitarget tracking (e.g. Bar-Shalom [7, 8], Blackman [9], Fox *et al.* [10]). 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.

APPROACH

Overview

Since the CMOMMT problem can be shown to be NP-complete, and thus intractable for computing optimal solutions, we propose an approximate control mechanism that is shown to work well in practice. This approximate control mechanism is based upon our previous work, described in [11, 12], which defines a fully distributed, behavior-based software architecture called ALLIANCE that enables fault tolerant, adaptive multi-robot action selection. This architecture is a hybrid approach to robotic control that incorporates a distributed, real-time reasoning system utilizing behavioral motivations above a layer of low-level, behavior-based control mechanisms. This architecture for cooperative control utilizes no centralized control; instead, it enables each individual robot to select its current actions based upon its own capabilities, the capabilities of its teammates, a previous history of interaction with particular team members, the current state of the environment, and the robot's current sensory readings. ALLIANCE does not require any use of negotiation among robots, but rather relies upon broadcast messages from robots to announce their current activities. The ALLIANCE approach to communication and action selection results in multi-robot cooperation that gracefully degrades and/or adapts to real-world problems, such as robot failures, changes in the team mission, changes in the robot team, or failures or noise in the communication system. This approach has been successfully applied to a variety of cooperative robot problems, including mock hazardous waste cleanup, bounding overwatch, janitorial service, box pushing, and cooperative manipulation, implemented on both physical and simulated robot teams.

Our proposed approach to the CMOMMT problem is based upon the same philosophy of control that was utilized in ALLIANCE. In this approach, we enable each robot team member to make its own action selections, without the need for any centralized control or negotiation. The low-level, behavior based control of each robot calculates local force vectors that attract the robot to nearby targets and repel the robot from nearby teammates. Added above the low-level control is a higher-level reasoning system that generates weights to be applied to the force vectors. These weights are based upon previous experiences of the robot, and can be in the form of motivations of behavior or rule-based heuristics. The high-level reasoning system of an individual robot is thus able to influence the local, low-level control of that robot, with the aim of generating an improved collective behavior across robots when utilized by all robot team members.

Target and robot detection

Ideally, robot team members would be able to passively observe 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 [13] that is in use at our CESAR laboratory) to determine their own position and the position of targets within their sensing range, and communicate this information to the robot team members within their communication range[1].

For each robot r_i, we define the *predictive tracking range* as the range in which targets localized by other robots $r_k \neq r_i$ can affect r_i's movements. Thus, a robot can know about two types of targets: those that are directly sensed or those that are "virtually" sensed through predictive tracking. 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. We

[1]This approach to communication places an upper limit on the total allowable number of robots and targets at about 400. Since the communication is $O(nm)$, we compute this upper limit by assuming a 1.6 Mbps Proxim radio ethernet system (such as the one in our laboratory) and assuming that messages of length 10 bytes per robot per target are transmitted every 2 seconds. With these numbers, we find that nm must be less than 4×10^4 bps to avoid saturation of the communication bandwidth.

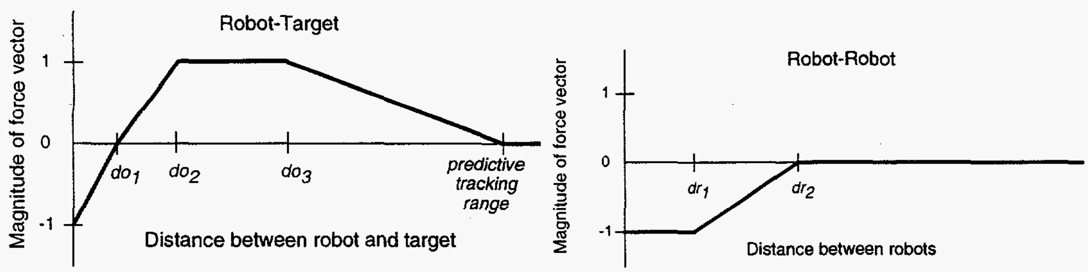

Figure 1: Functions defining the magnitude of the force vectors to nearby targets and robots.

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. Bar-Shalom [8]) should be used to uniquely identify each target based upon likely trajectories.

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. The first function in figure 1 defines the relative magnitude of the attractive forces of a target within the predictive tracking range of a given robot. Note that to minimize the likelihood of collisions, the robot is repelled from a target if it is too close to that target ($distance < do_1$). The range between do_2 and do_3 defines the preferred tracking range of a robot from an object. In practice, this range will be set according to the type of tracking sensor used and its range for optimal tracking. The attraction to the object falls off linearly as the distance to the object varies from do_2. The attraction goes to 0 beyond the predicted tracking range, indicating that this object is too far to have an effect on the robot's movements.

The second function of figure 1 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.

One problem with using only force vectors, however, is that of local minima. As defined so far, the force vector computation is equivalent for all targets, and for all robots. Thus, we need to inject additional high-level reasoning control into the system to take into account more global information. This reasoning is modeled as predictive weights that are factored into the force vector calculation, and are described in the next subsection.

High-level reasoning control

To help resolve the problems of local minima, the higher-level reasoning control differentially weights the contributions of each target's force field on the total computed field. This higher-level knowledge can express any information or heuristics that are known to result in more effective global control when used by each robot team member locally. Our present approach expresses this high-level knowledge 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. In the future, we plan to incorporate the ALLIANCE motivation of "impatience", if a nearby robot does not appear to be satisfactorily observing its local targets (perhaps due to faulty sensors). This impatience will 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 higher-level weight information is combined with the local force vectors to generate the commanded direction of robot movement. This direction of movement is given by:

$$\sum_{i=0}^{N}(FVO_i \times Pr(exists_i) \times Pr(NT_i)) + \sum_{j=0}^{M} FVR_j$$

where FVO_k is the force vector attributed to target o_k, $Pr(exists_k)$ is the probability that target o_k exists, $Pr(NT_k)$ is the probability that target o_k is not already being tracked, and FVR_l is the force vector attributed to robot r_l. This movement command is then sent to the robot actuators to cause the appropriate robot movements. We also incorporate a low-level obstacle avoidance behavior that overrides these movement commands if it would likely result in a collision.

EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the effectiveness of the CMOMMT algorithm, we conducted experiments both in simulation and on a team of mobile robots. In the simulation studies, we compared four possible cooperative observation algorithms: (1) CMOMMT (high-level plus local control), (2) *Local control only*, (3) *Random/linear robot movement*, and (4) *Fixed robot positions.*

In all of these experiments, targets moved according to a "random/linear" movement, which causes the target to move in a straight-line until an obstacle is met, followed by random turns until the target is able to again move forward without collision. The *local control only* algorithm computed the motion of the robots by calculating the unweighted local force vectors between robots and targets. This approach was studied to determine the effectiveness of the high-level reasoning that is incorporated into the CMOMMT algorithm. The last two algorithms are control cases for the purposes of comparison: the *random/linear robot movement* approach caused robots to move according the the "random/linear" motion defined above, while the *fixed robot positions* algorithms distributed the robots uniformly over the area $\mathcal{S}$, where they maintained fixed positions. In both of these control approaches, robot movements were not dependent upon target locations or movements (other than obstacle avoidance).

We compared these 4 approaches by measuring the average value of the $A(t)$ matrix (see PROBLEM DESCRIPTION section) during the execution of the algorithm. Since the algorithm performance is expected to be a function f of the number of robots n, number of targets m, the range of a given robot's sensor r, and the relative size of the area $\mathcal{S}$, we collected data for a wide range of values of these variables. To simplify the analysis of our results, we defined the area $\mathcal{S}$ as the area within a circle of radius R, fixed the range of robot sensing at 2,600 units of distance, and included no obstacles within $\mathcal{S}$ (other than the robots and targets themselves, and the boundary of $\mathcal{S}$).

We collected data by varying n from 1 to 10, m from 1 to 20, and R from 1,000 to 50,000 units. For each instantiation of variables n, m, and R, we computed the average $A(t)$ value every $\Delta t = 2$ seconds of a run of length 2 minutes; we then repeated this process for 250 runs for each instantiation to derive an average $A(t)$ value for the given values of n, m, and R. In all runs of all 4 algorithms, the targets were placed randomly at the center of $\mathcal{S}$ within a circle of radius 1,000. In all runs of all algorithms (except for *fixed robot positions*), the robots were also placed randomly within the same area as the targets.

To analyze the results of these experiments, we speculated that the function $f(n, m, r, R)$ would be proportional to ratio of the total collective area that could be covered by the robot sensors (i.e. $n\pi r^2$) over the area that would be allotted to one target (call it a *target slot*), were $\mathcal{S}$ divided equally over all targets (i.e. $\frac{\pi R^2}{m}$), we have:

$$f(n, m, r, R) = \frac{n\pi r^2}{\frac{\pi R^2}{m}} = \frac{nmr^2}{R^2}$$

Thus, this function was used to compare the similarity of experiments that varied in their instantiations of n, m, and R.

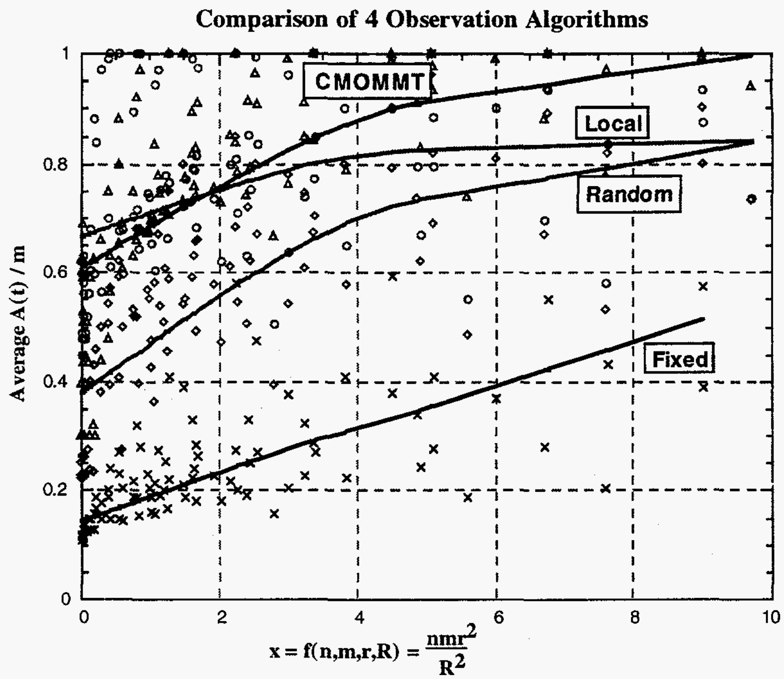

Figure 2: Comparison of 4 cooperative observation algorithms.

Since the optimum value of the average $A(t)$ for a given experiment depends upon the value of m (and, in fact, equals m), we normalized the experiments by plotting the average $A(t)/m$ which is the average percentage of targets that are within some robot's view at a given instant of time.

Figure 2 gives the results of our experiments, plotting the average $A(t)/m$ versus $f(n,m,r,R)$ for all of our experimental data. For each algorithm, we fit a curve to the data using the locally weighted Least Squared error method. Since there is considerable deviation in the data points for given values of $f(n,m,r,R)$, we computed the statistical significance of the results using the Student's t distribution, comparing the algorithms two at a time for all 6 possible pairings. In these computations, we used the null hypothesis: $H_0 : \mu_1 = \mu_2$, *and there is essentially no difference between the two algorithms.* Under hypothesis H_0:

$$T = \frac{\overline{X}_1 - \overline{X}_2}{\sigma\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}} \quad where \quad \sigma = \sqrt{\frac{n_1 S_1^2 + n_2 S_2^2}{n_1+n_2-2}}$$

Then, on the basis of a two-tailed test at a 0.01 level of significance, we would reject H_0 if T were outside the range $-t_{.995}$ to $t_{.995}$, which for $n_1 + n_2 - 2 = 250 + 250 - 2 = 498$ degrees of freedom, is the range -2.58 to 2.58. For the data given in figure 2, we found that we could reject H_0 at a 0.01 level of significance for all pairing of algorithms that show a visible difference in performance in this figure. Thus, we can conclude that the variation in performance of the algorithms illustrated by the fitted curves in figure 2 is significant.

We see from figure 2 that the CMOMMT and *local control only* algorithms perform better than the two naive control algorithms, which is expected since the naive algorithms use no information about target positions. Note that all approaches improve as the value of $f(n,m,r,R)$ increases, corresponding to a higher level of robot coverage available per target. The *random/linear robot movement* approach performed better than the *fixed robot positions*, most likely due to the proximity of the initial starting locations of the robots and objects in the *random/linear robot movement* approach. This seems to suggest that much benefit can be gained by learning areas of the environment $\mathcal{S}$ where targets are more likely to be found, and concentrate on locating robots in those areas.

Of more interest, we see that the CMOMMT approach is superior to the *local control only* approach for values of $f(n, m, r, R)$ greater than about 2; the *local control only* approach is slightly better for $f(n, m, r, R)$ less than 2. This means that when the fraction of robot coverage available per target is low (< 2), relative to the size of $\mathcal{S}$, then robots are better off *not* ignoring any targets, which is essentially what happens due to the high-level control of CMOMMT. Examples of experimental scenarios where the *local control only* approach is better than the CMOMMT approach are (n, m, R) = (2,1,5000-50000), (2,2,4000-50000), (3,1,5000-50000), (3,2,5000-50000), (3,3,8000-50000), and (3,4,8000-50000). However, for more complex cases, where the number of targets is much greater than the number of robots, and the environmental area is not "too large", we find that the higher-level reasoning provided by CMOMMT works better. Examples of scenarios where CMOMMT is better include (n, m, R) = (2,4,1000-5000), (2,6,1000-6000) (2,20,1000-10000),(3,3,1000-5000), (3,4,1000-6000), (3,6, 1000-7000), and (3,12,1000-11000). Note that CMOMMT approaches perfect performance as $f(n, m, r, R)$ reaches 10, whereas the results of the *random/linear robot movement* and /it local control only approaches begin to level off at around 85%.

In continuing and future work, we are determining the impact of these results on multi-robot cooperative algorithm design.

We have also implemented the CMOMMT algorithm on a team of a team of four Nomadic Technologies robots to illustrate the feasibility of our approach for physical robot teams. We have demonstrated a very simple case of cooperative tracking using these robots. Refer to [14] for details.

CONCLUSIONS

Many real-world applications in security, surveillance, and reconnaissance tasks require multiple targets to be monitored using mobile sensors. We have presented an approximate, distributed approach based upon the philosophies of the ALLIANCE architecture and have illustrated its effectiveness in a wide range of cooperative observation scenarios. This approach is based upon a combination of high-level reasoning control and lower-level force vector control that is fully distributed across all robot team members and involves no centralized control. Empirical investigations of our cooperative control approach have shown it to be effective at achieving the goal of maximizing target observation for most experimental scenarios, as compared to three other feasible control algorithms.

ACKNOWLEDGEMENTS

This research is sponsored by the Office of Engineering Research, Basic Energy Sciences of the U.S. Department of Energy, under contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation.

REFERENCES

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

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

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

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

[5] S. M. LaValle, D. Lin, L. J. Guibas, J-C. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In submitted to 1997 International Conference on Robots and Automation, 1997.

[6] S. M. LaValle, H. H. Gonzalez-Banos, C. Becker, and J-C. Latombe. Motion strategies for maintaining visibility of a moving target. In submitted to 1997 International Conference on Robots and Automation, 1997.

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

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

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

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

[11] 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.

[12] 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).

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

[14] Lynne E. Parker and Brad Emmons. Cooperative multi-robot observation of multiple moving targets. In Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pages 2082–2089, Albuquerque, New Mexico, April 1997.