Multirobot Cooperation for Surveillance of Multiple Moving Targets - A New Behavioral Approach

Andreas Kolling   Stefano Carpin
School of Engineering and Science
International University Bremen – Germany

Abstract— **This paper presents a behavior-based solution to the problem of observing multiple mobile targets by multiple mobile robots. Robots sense targets using sensors and in addition exchange information about them with other robots. Workload is shared between different robots by requesting help when targets are escaping and supporting robots requesting such help. We provide a detailed description of the proposed solution, as well as significant simulation tests to outline its performance. The described approach outperforms formerly proposed solutions.**

I. INTRODUCTION

The rapid development of wireless sensor networks related technologies and the increasing demand for data gathering over wide areas are pushing forward research for systems capable of efficiently collecting this information. Surveillance and security-related tasks are the two most obvious applicative scenarios, but not the only ones. In cases where human beings cannot safely operate, networks of sensors and autonomous robots are of big help. If the environment to be monitored is large, it exhibits significant dynamics and multiple movable elements should be tracked, the use of teams of autonomous robots appears more effective than a static arrangement of sensors, although the optimal approach would probably involve a synergy between them. In this paper we present some novel results concerning the Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT) task. First formalized in [1], the problem consists in the deployment of a team of mobile robots whose task is to *observe* a set of targets that move within a defined area of interest, potentially trying to escape the observers. The goal is to keep as many targets as possible under observation by at least one of the robots. Our contribution is in the design of a distributed behavior based control systems where robots share workload by assuming responsibilities concerning the observation of certain targets. In addition, robots may explicitly ask for help if they realize that some target will soon escape their observation, and they can provide support to robots asking for assistance. A set of three behaviors will be illustrated, which builds upon formerly developed strategies for solving the same problem.

Section II briefly discusses former work related to the problem under investigation. A sound definition of the CMOMMT task is offered in Section III, as well as a detailed discussion of the formerly developed A-CMOMMT strategy. This is needed because our strategy builds upon A-CMOMMT, and we will later compare our results with it. Our proposed idea and its motivations are introduced in Section IV. The experimental setup and the derived numerical results are illustrated in Section V and conclusions are provided in the final Section VI.

II. RELATED WORK

There has been a significant amount of research devoted to the use of sensors and intelligent devices for surveillance. A complete coverage is beyond the scope of our paper, and we shall provide only selected links to problems similar to the ones discussed in this paper. It is possible to divide related theory in two main branches. Coverage control received attention from the control theory community. Cortés et al. [2] address the problem of coordinating groups of autonomous vehicles equipped with sensors whose task is to perform distributed sensing for coverage control using a dynamic Voronoi based subdivision. Other geometric based approaches include [3]. Inside the robotics community the CMOMMT problem and similar tasks enjoyed significant popularity, also because they naturally call for coordination and cooperation. Beyond the already cited work by Parker [1], there have been numerous other contributions. Jung and Sukhatme [4] present a *region-based* approach, where stationary sensors and mobile robots cooperate while tracking multiple targets. More precisely, their approach is based on the idea of subdividing the area to be observed into different regions and assigning different robots to different regions to track targets in it. Vidal et al. [5] illustrate a complete system where a team of unmanned aerial vehicles and unmanned ground vehicles track a team of intruders, while also mapping an unknown environment. A similar problem, although limited to terrestrial vehicles only, has been studied by Saptharishi et al. [6].

III. PROBLEM FORMALIZATION AND THE A-CMOMMT STRATEGY

The formalism for CMOMMT which is presented here was stated by Parker in [1]. Let:

1) S: two dimensional, bounded, enclosed region as area of interest
2) V: team of m robot vehicles, $v_i, i = 1, \ldots, m$, with sensors
3) *sensor_coverage*(v_i): subset of S observable by robot v_i. This region varies as the robot v_i moves inside S.
4) $O(t)$: a set of n targets, $o_j(t), j = 1, 2, \ldots, n,$

5) $B(t) = b_{ij}(t)$ such that $b_{ij} = 1$ if robot v_i is observing target $o_j(t)$ in S at time t, 0 otherwise. Robot v_i observes target $o_j(t)$ if $o_j \in sensor\_coverage(v_i)$.

The goal is to develop and algorithm that maximizes the following metric for surveillance:

$$A = \sum_{t=1}^{T} \sum_{j=1}^{n} \frac{g(B(t), j)}{T}$$

where $g(B(t), j)$ is 1 if there exist an i such that $b_{ij}(t) = 1$ and 0 otherwise. Informally stated, the problem requires to maximize the average number of targets that are observed by at least one of the robots. It is furthermore assumed that the overall sensor coverage is much smaller than the area to be monitored. Moreover, we assume that the maximum speed of the targets is smaller than the maximum speed of the robots. If this is not the case, it would be easy for an intelligent target to always escape robots by just moving at maximum speed. Finally, all robots share a common global reference system, and have a communication mechanism which allows them to send/receive messages in broadcast mode, provided that the receiver/sender is closer than a fixed distance r_c Concerning sensing, we instead will not necessarily assume that robots are equipped with omnidirectional sensors, nevertheless supposing that only targets within distance r_s from the robot can be detected. Throughout the paper we will adopt the same hypotheses stated in [1], i.e. that the sensing rage r_s is smaller than the communication range r_c. All the above hypothesis are nowadays achievable with off the shelf components. We will concentrate only on *distributed* approaches, i.e. we reject the possibility that decisions about robots' motion are taken by a single entity (within or outside the team itself). Ignoring the two trivial approaches where robots stand still or wander randomly, the *local force* strategy is probably the simplest one. It drives the robot by computing a desired direction vector that incorporates the following principles:

1) Stay close to targets that are not too far away.
2) Stay away from other robots
3) Stay away from targets that get too close
4) Have all surrounding robots and targets influence the movement of the robot.

These principles can be expressed by the following formula:

$$f(v_i, t) = \sum_{k=1}^{n} t_{ik}(t) + \sum_{k=1}^{m} r_{ik}(t)$$

where $f(v_i, t)$ is the force vector applied to robot v_i at time t, t_{ik} is the force vector from robot i to target k, and r_{ik} the force vector from robot i to robot k with $r_{ii} = 0$, at time t. These last two terms depend exclusively on the target-robot and robot-robot distance, and their precise profiles are given in figure 1. Precise values for the constants d_{oi} and d_{rj} will be provided while discussing experimental results. The *predictive tracking range* appearing in figure 1 is the range within targets located by robots other than v_i or targets that left the sensing range influence v_i's motions. The location of targets beyond sensing range and closer than tracking range is predicted linearly using the last known movement of the target. The A-CMOMMT improves the local force method by introducing a weight factor in the first summation. More precisely, each factor t_{ik} is weighted by a factor w_{ik}. The w_{ik} factors account for cooperation between different robots. The idea is to share the workload between different robots. If target o_j is currently being observed by robot v_k, then robot v_i should be less attracted to observe o_i. This turns out in a choice of w_{ij} smaller than 1. Extensive simulation results presented in [1] illustrate that this approach outperforms the local force method.

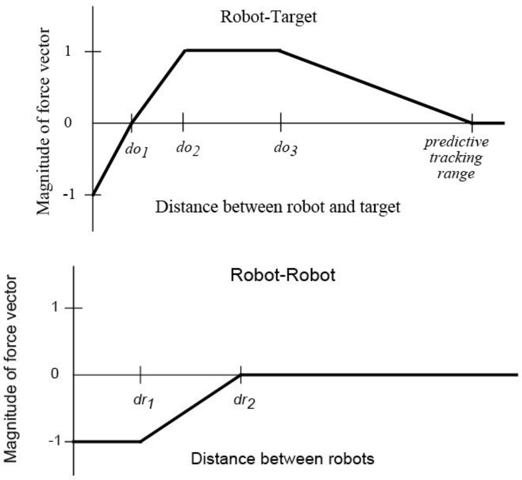

Fig. 1. Magnitude of the force vectors from robot to target and robot to robot (taken from [1]).

IV. THE B-CMOMMT STRAGEGY

In our approach we would rather view CMOMMT problem in the framework of a decision process. This view is justified by the fact that we are concerned with a decoupled system, in which each robot has access to local information via sensors, and via other robots that communicate with it. Provided with this local information, a robot has to decide which targets to follow, which robots to be repelled from, and which ones to be attracted to. More precisely, we would like to overcome some problematic situations that may arise in the A-CMOMMT framework. A difficult situation is depicted in figure 2. We have one robot following two targets that move into opposite directions and another robot within communication range. It is not strictly necessary that the two targets are at precisely opposite directions. The important fact is that under the influence of both targets robot v_1 attempts to move into the center of gravity of the two targets which will vary only slightly if we vary the target positions slightly. It is quite clear that within a few time steps at least one of the targets will have left the predictive tracking range and hence not have any influence on the robots movement. But even if one target leaves the tracking range several time steps before the other target, then

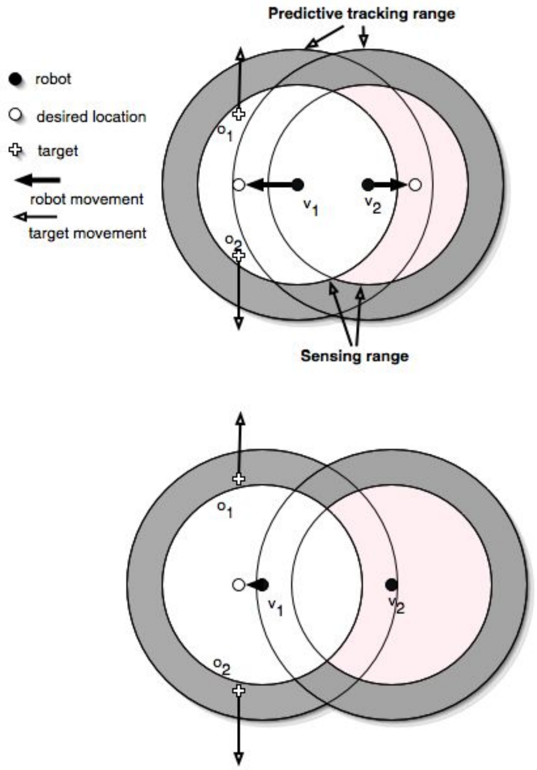

Fig. 2. Schematic figure of two screenshots of a situation with undesirable behavior.

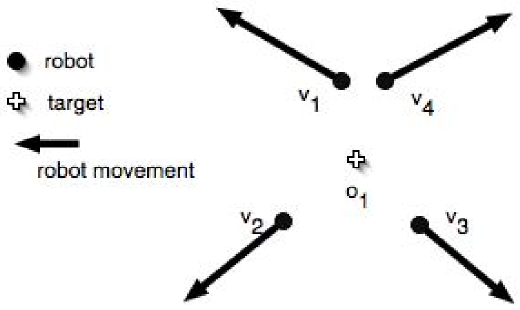

Fig. 3. Schematic figure of a situation with repulsion of robots governing the movement.

v_1 still has to turn and move into the direction of one of these targets. If the robot is subject to differential constraints it will take several time steps to turn and accelerate. In the meantime, the target might well escape. With no differential constraints this situation only becomes a problem if both targets vanish exactly at the same time step. Depending on the differential constraints this problem can become more significant. On the other hand, it is also unfortunate that robot v_2's resources are not being utilized while knowing that robot v_1 will at least lose one target out of observation. Another problematic situation arises from reducing the weight for a target if also observed by another robot, as this leads to a reduced attraction for both robots. If both of these robots have another target under surveillance on opposite ends, then the reduced weight of the target observed by both, the repulsion between the two robots, and the attraction to the other end due to the other targets can lead to a loss of the target in the middle even if it was possible to observe it for longer. If one robot was to assume full responsibility for the middle target, preferably the closer one, we would have a more efficient sharing of workload, as other robots are totally freed from the obligation to observe the middle target. In particular, with many robots and targets there is a disadvantage, also mentioned in [1], as the repulsion effect between robots becomes larger than the attraction to the shared targets and starts to determine a robot movement similar to the one seen in figure 3.

A. *Basic ideas*

Our approach, which we dub B-CMOMMT, operates under similar assumptions as in the problem definition of CMOMMT with the difference that it accommodates for varying restrictions of the sensors. It uses the same magnitudes for the force vectors, and can be seen as a high level control to set the weights in the formula for the weighted force vectors with the important difference that we also have weights for robots. B-CMOMMT realizes a simple behavioral architecture. A robot can be in one of three modes and set one cooperation variable. The modes are three: *Follow Targets*, *Help* and *Explore*. The mode *Follow Targets* is set as soon as a robot is following one or more targets. *Help* and *Explore* are set if there are no known targets. The *Help* modus has precedence and is set if there are robots within communication range that predict a target loss. *Explore* mode is set if the latter does not occur. The cooperation variable "request help" is used to communicate to other robots that a target loss is about to occur. It is set to zero if the robot is observing no targets or multiple targets without predicting a loss of a target. It is set to one as soon as a target loss is predicted. A simple heuristic for the prediction of target loss, that we used, is to set "request help" when a target enters the predictive tracking area. This behavioral architecture is supported by a tagging system operating under the principle that one target should only influence the movement of one robot at a particular time.

B. *Tagging*

The proper tagging of targets requires the additional hypothesis that targets are distinguishable, a prerequisite not necessarily present in the original CMOMMT problem formulation, but also made in [1]. As mentioned in [1], to achieve target identification one can use the global reference frame and identify targets by their position. The tagging proceeds as follows. The closest known target to a robot is tagged regardless of its position. All other targets within a safe distance, i.e. in our case within sensing range, are then also tagged. For each tag the robot communicates the distance of the tag to the other robots within communication range. It suffices to have the communication range at twice the value as the tracking range for this mechanism to work reliably.

If a tag from another robot, received via the communication device, is placed on the same target, then the robot with the larger distance from the tag discards it. The idea of the tagging only within a safe area is that a robot should be able to still follow any tagged target if it was to decide to follow this target exclusively. This should be considered for determining the safe area for a particular application. To illustrate this, let y be the maximum speed of the target, then in the worst possible case the robot is going with full speed into the opposite direction of the target. Let it be possible for the robot to reverse direction within x time steps, keeping the speed. In these x time steps the target should not have escaped the predictive tracking range to enable the robot to catch up again. For $y = 150$ units per time step and $x = 8$, we obtain a $safe\_distance = tracking\_range - 150 \cdot 8$. While targets are in the safe area and tagged by the robot they will all influence the robots desired direction for the movement in the next time step with full weights, such that the robot attempts to move towards the center of mass of these targets. Analogue to the A-CMOMMT algorithm, this tries to optimize the use of resources. But as another advantage we free resources as we will not have multiple robots following the same targets in safe situations. After the tagging is done, the mode of the robot can easily be determined. If a robot has a tag on a target it switches into *Follow Target* mode. If no tags are available and it receives a "request help" message from a robot it switches into *Help* mode. If no tags and no call messages are available it switches to *Explore* mode. Ideally, tags on targets are actively communicated to other robots, hence the need for a target identification, but the basic idea could also be used by an indirect tagging, i.e. checking whether any other robot is closer to the target and then discarding the tag.

C. Robot modes

The three possible modes are here briefly summarized:

- **Follow mode**: In the follow mode a robot calculates its desired direction of movement by considering all and only tagged targets. The formula for the force vector for robot i is
$$\sum_{o_j \in T_i} w_{ij} t_{ij} + \sum_{v_j \in C_i} w^r_{ij} r_{ij}$$
where T_i is the set of targets tagged by v_i and C_i is the set of robots within communication range.
- **Help mode**: In the help mode a robot chooses the closest robot that it receives a "request help" message from, and moves into this direction. Due to the predicted target loss the helping robot has a good chance on gaining a tag in the area of the robot that requested the help. Currently the help call simply redirects robots that are not following a target into areas with a higher density of targets.
- **Explore mode**: When a robot enters this mode it employs a general exploration strategy in order to seek new targets. To allow a fair comparison with A-CMOMMT we currently use the position of all robots in communication range and compute the desired direction of movement precisely as done in A-CMOMMT when there are no targets observed.

V. EXPERIMENTAL RESULTS

Extensive simulations have been performed to validate the B-CMOMMT algorithm, as well as to compare its performance with other approaches, and measure its sensitivity to the involved parameters. Simulations have been executed both in Matlab and using the Player1.6.5/Stage1.6.1 software [7][8]. In Matlab the approaches were implemented under ideal hypothesis, while more realistic settings have been used in the Player simulation.

A. Experimental set up

Matlab simulations used the same values for most parameters as in [1]. The only differences being that the circle was set to a fixed radius of 15,000 units and the sensing range to 2000 units. The area of initial deployment of robots and targets was set to one third of the radius of the environment with targets and robots randomly distributed in this area. The movement of the targets was random with a 5 percent chance of changing the direction between $-90°$ and $90°$. The target speed was fixed to a random value between 0 and the maximum target speed of 150 units/second. Parameters were set as follows: $d_{o1} = 400, d_{o2} = 800, d_{o3} = 2600, predictive\_tracking\_range = 3000, d_{r1} = 1250, d_{r2} = 2000$. All robots had an acceleration of 20 units/second and could turn 0.5 radians/second independent of the current speed. For each set of parameters 200 runs with randomized target placement were performed to derive an average coverage. In Player/Stage both approaches have been implemented according to the availability of the resources. The predictive tracker had to be implemented without the communication of the target's positions between robots within the predictive tracking range because of technical problems. As for now this can be seen as an example of the adaptability of both approaches as they were readily implementable despite the constraint of the communication bandwidth. All parameters in the Player/Stage simulations were equivalent to those from the Matlab simulations and have been translated via the ratio of 1m=500units. As robots we used the position device with 2 lasers with one fiducial finder on each. The first laser had a range equal to the sensing range and could only identify targets. The second laser had a range equal to the communication range and served to identify the position of robots. The use of the laser to identify robots freed communication bandwidth by not having to communicate the position of all robots, but robots could be hidden by other robots and targets. Communication was only used once for B-CMOMMT upon discovering a target and checking for a tag or to call or check for help. All robots had a GPS localization and a linear predictive tracker that predicted the target movement of any target that had left the sensing range by updating the position according to the last known movement of the target for at most 10 update cycles or until the target left the tracking range. The differential constraint on the robots were a limited forward speed of 0.1 m/s for turn rates bigger than 0.5 radiens/s. To

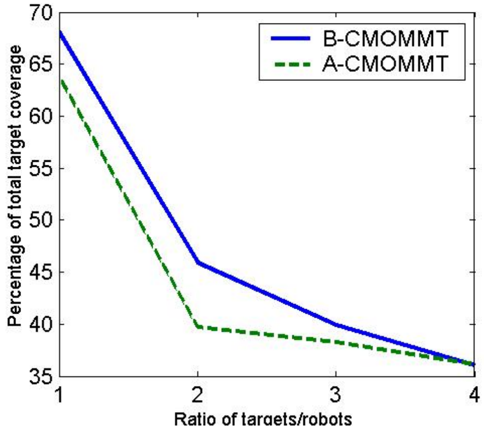

Fig. 4. Performance of A-CMOMMT and B-CMOMMT with evasive target movement for target/robot ratios of 10/10, 10/5,15/5,20/5

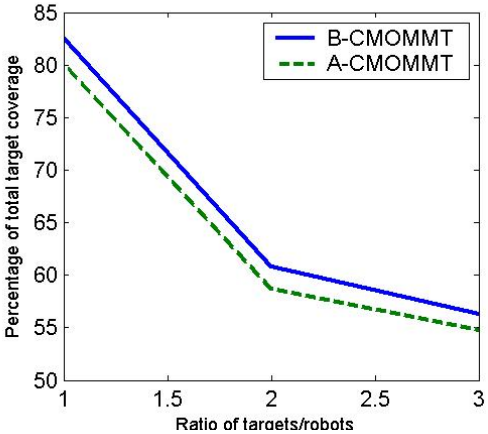

Fig. 5. Performance of A-CMOMMT and B-CMOMMT with random target movement for target/robot ratios of 10/10,10/5,15/5,20/5

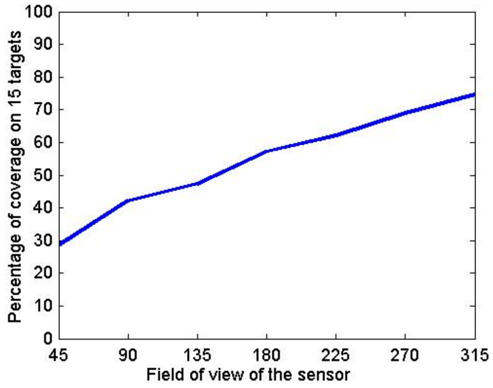

Fig. 6. Coverage under varying field of view in degrees in Player simulations of BCMOMMT

avoid large turn rates, which would lead to too large turns whenever the next update cycle is not executed precisely one second later, half the desired turn rate was submitted to the position device. The Player/Stage simulations had two types of target movement, evasive and random. For the evasive type all targets were equipped with a laser and fiducial finder with a range of 3m to detect robots and calculate a force vector into opposite direction than all visible robots analogue to the repulsion that robots experience from other robots in the A-CMOMMT approach. The random target movement was set as in the Matlab simulation with a chance of 5% per second to turn between $-90°$ and $90°$. Each average value is derived from 100 runs with a random placement of robots and targets. In total all figures shown are based on 9100 simulations.

B. Comparison of A-CMOMMT and B-CMOMMT

The first set of experiments in Player was carried out with evasive target movement. Whenever a robot discovered a target and moved closer than 3m the target recognized the robot and moved away with maximum speed of 0.3m/s. Surveillance of more than one target hence becomes more difficult and target loss is likely to occur. Discovering and observing a single target, however, remains comparatively manageable. If a target enters the sensing range of a robot while the robot is facing into the opposite direction it will take at least two seconds until the robot can turn and adjust into the direction of the target and another second to accelerate. As the target can move at most 0.9 m in 3 seconds and first has to travel an additional meter to see the robot after entering the sensing range it is possible under good conditions for a robot to follow any target that enters the sensing range. Experiments for target to robot ratios of 10/10,10/5,15/5, 20/5 were carried out. The results are summarized in figure 4. Using a Welch two sample t-test the difference between the two approaches were shown to be significant with a $p < 0.01$ for the ratios of 10/10 and 10/5. The second set of experiments in Player/Stage served to compare the performance in a setting with random target movement. On average targets moved with a speed of 0.15 m/s. Targets had a chance of 50% to change direction at least once within 10 second, so on average a chance of 50% to change direction on 1.5m. Target observation once the target is discovered is relatively easy and even the observation of multiple targets by one robot can in many cases happen without target loss as the target movement is rather limited. The results for this set are summarized in figure 5. With 100 runs for each average we could not detect a statistically significant difference in the means presented in the figure.

C. Investigating B-CMOMMT

The third set of experiments in Player/Stage investigated the change in performance with varying the field of view of 10 robots observing 15 targets. The laser with the fiducial finder to detect targets was restricted to $45°$, $90°$, $135°$, $180°$, $225°$, $270°$ and $315°$ as field of view. The total area that can be covered by the robots reduces linearly with the restriction of the field of view. Figure 6 shows the percentage of the coverage on 15 targets for all the above angles. The performance of B-CMOMMT scales as expected, suggesting

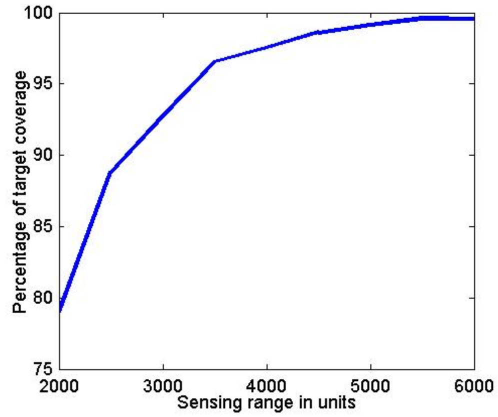

Fig. 7. Coverage under varying sensing range in a Matlab simulation of BCMOMMT

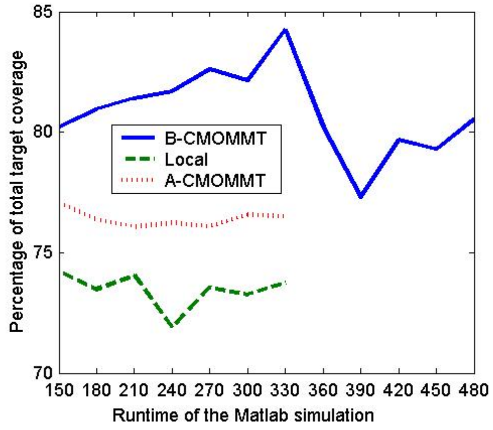

Fig. 8. Performance with increasing duration of simulations

a stable behavior even with restricted sensors. Two sets of simulations have been performed in Matlab to investigate the behavior under longer simulation times and coverage under varying sensing range. The main focus in these runs was to investigate the behavior of the algorithm in a perfect implementation rather than the absolute performance. For the simulation time we also included A-CMOMMT and the local approach. Varying the sensing range for a simulation with 10 robots and 10 targets lead to the results shown in figure 7. For small sensing ranges the slope is quite steep, showing that with scarce resources small increases are utilized well. At a sensing range of 5000 units we already get almost complete coverage as most targets are observed at deployment and thanks to the tagging, robots disperse properly into all directions into which targets are moving. In general, a large sensing range leads to many overlapping sensing areas of the robots. The more overlap in the sensing area of two robots the more similar their movement will be in A-CMOMMT. Hence an increase in sensing range naturally leads to the undesired effect of a similar direction of movement across robots. Of particular interest and one area for further research is the investigation of a saturation of the performance with increasing runtime. The initial deployment within an area of the environment certainly has an influence on the performance. An understanding on how the behavior evolves over time while the targets disperse in the environment is essential for any long term application. A first series of tests with 10 robots and 10 targets, seen in figure 8, suggests that A-CMOMMT and the local approach saturate fairly early, while B-CMOMMT still seems to change its performance significantly beyond the standard runtime of 120s.

VI. CONCLUSIONS

The new behavioral approach to CMOMMT provides several extensions and improvements to the existing approaches. Firstly, it realizes a different principle of shared workload, namely that of assigning responsibilities instead of dividing them. Secondly, a sharing of resources can be established via the help components and upon mutual agreement of the helper and the requesting robot. These enhancements already lead to a significant improvement in performance in scenarios with difficult target movements, while in scenarios with easy target movements the former performance could be maintained. Furthermore, it was shown that B-CMOMMT scales well with respect to restricting the field of view and increasing the sensing range. This indicates that the principles of B-CMOMMT allow a better use of the available resources for target coverage for multiple mobile sensors. Furthermore, it is straightforward to integrate exploration strategies into B-CMOMMT. Following the first sets of experiments presented in this paper a wide range of possible scenarios is still to be investigated, in particular more realistic environments with obstacles, cooperation with static sensors, and targets distributed not in the area of deployment of the robots.

REFERENCES

[1] L. E. Parker, "Distributed algorithms for mulit-robot observation of multiple moving targets," *Autonomous robots*, vol. 12, no. 3, pp. 231–255, 2002.

[2] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing netorks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[3] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[4] B. Jung and G. Sukhatme, "Tracking targets using multiple mobile robots: the effect of environment occlusion," *Autonomous robots*, vol. 13, no. 2, pp. 191–205, 2002.

[5] R. Vidal, O. Shakernia, H. Kin, D. Shim, and S. Sastry, "Probablistic pursuit-evasion games: theory, implementation and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002.

[6] M. Saptharishi, C. S. Oliver, C. Dihel, K. Bhat, J. Dolan, A. Trebi-Ollennu, and P. Kohsla, "Distributed surveillance and reconnaisance using multiple autonomous atvs: Cyberscour," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 826–836, 2002.

[7] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of ICAR*, 2003, pp. 317–323.

[8] "Player/stage project," http://playerstage.sourceforge.net, 2005.