Proceedings of the 2006 IEEE
International Conference on Robotics and Biomimetics
December 17 - 20, 2006, Kunming, China

Multi-target Detection by Multi-sensor Systems: A Comparison of Systems

Ganesh P K
Robotics Research Center
IIIT Hyderabad
Hyderabad – 500 032, INDIA
ganesh@students.iiit.ac.in

K Madhava Krishna
Robotics Research Center
IIIT Hyderabad
Hyderabad- 500 032, INDIA
mkrishna@iiit.ac.in

Paulo Menezes
Inst. of Robotics & Systems
Univ. of Coimbra
Coimbra – 3040-290, PORTUGAL
paulo@isr.uc.pt

Abstract **– Different methodologies exist to direct the motion of sensors to detect targets moving across an environment in various scenarios. However, some of these do not model that navigation scenario in an environment in which obstacles are present. We extend our earlier algorithm for optimal target detection, as well as other algorithms reported in literature, to this case, and make a detailed comparison of their performance. This makes clear that the current algorithm is competitive in applications where target statistics are known in advance; otherwise, a heuristic technique by Sukhatme and Jung performs best.**

Keywords- Multi-sensor navigation, Target detection

I INTRODUCTION

The problem of designing target detection algorithms to maximize the number of detections has attracted attention over the past few years. Three representative algorithms for the purpose are those of :(a) Krishna et.al [15] which solves the problem of optimal motion of sensors with target emission and travel statistics known, (b) Parker [12] which employs a force-based heuristic to achieve good performance for unknown target statistics , and (c) Sukhatme and Jung [8] which uses a division based approach for unknown target statistics , and In real-life scenarios, however, obstacles scattered across the environment must be taken into account.

To this end, we extend two of the above-mentioned algorithms in the wake of obstacle presence, and compare their performance in the two cases, informally called the *known* and *unknown* case hereafter. The known case has static target sources with pre-determined emission and travel statistics, whereas in the unknown*,* target sources may be moving at random, and travel statistics are unknown.

II. REVIEW OF EXISTING LITERATURE

Multi-sensor surveillance finds applications such as border patrol, guarding of secured areas, search and rescue and warehouse surveillance [1, 2]. It involves detection of multiple intrusions and tracking through coordination between the sensors. Detection and target tracking have been researched from multiple viewpoints. Some efforts have focused on the problem of identifying targets from a given dataset through particle filters [3], and probabilistic methods [4]. The problem of data association, or assigning sensor measurements to the corresponding targets, was tackled by the Joint Probabilistic Data Association Filters by the same researchers as in [3]. Kluge and others [5] use dynamic timestamps for tracking multiple targets. Krishna and Kalra [6] presented clustering based approaches for target detection and further extended it to tracking and avoidance. The focus of these approaches has been on building reliable estimators for predicting target trajectories that is different from the objective of this effort to maximize target detections.

Parker proposed a scheme for delegating and withdrawing robots to and from targets through the ALLIANCE architecture for distributed task allocation and sensor coordination in [7], wherein allocation and withdrawal are based on notions of impatience and acquiescence. Jung and Sukhatme [8] present a strategy for tracking multiple intruders through a distributed mobile sensor network and a technique to maximize sensor coverage [8, 9]. Lesser's group has made significant advances in the domain of distributed sensor networks [10] and sensor management [11]. The authors of the current paper present a constrained optimal target detection scheme in [15] and compare various resource allocation strategies in terms of their detection performance. The author of [13] has looked at the problem of static placement of sensors in predetermined polygonal environments; [14] describes a distributed sensor approach to target tracking using fixed sensor locations.

In this paper, however, our sensors are mobile; and among the approaches that we have encountered the closest to the this scenario are [8] and [12]. In [8] a motion strategy for tracking multiple targets based on density estimates is formulated., in which the robot attempts to maximize target detections by maintaining itself at a particular distance from the center of gravity of currently observed targets. In [12] a behavioral approach, A-CMOMMT, is compared with three other heuristic approaches where the sensor's motion strategy is arbitrary or random in the first, stationary (the sensor does not move) in the second, and based on local force control in the last.

III. ORIGINAL AND EXTENDED ALGORITHMS

In this section, we briefly review the three algorithms cited in Section 1, and extend them to accommodate for obstacles when necessary. The area of interest in which targets and sensors move is called the *environment*. Obstacles, usually straight line segments for ease of computation, may be scattered over the environment. No sensor can see through an obstacle, and neither sensor nor target may cross an obstacle in course of its motion.

A Krishna's algorithm

The algorithm of Krishna et .al. [15], relies on a predetermined knowledge of target source positions and target emission statistics to find an optimal path for each sensor over the next T time instants. The latter is computed with the following assumptions: The emission process from each source is Poisson distributed in time. The angle of target emission is uniformly distributed over $[0,\pi]$ with respect to (say) the x axis, and, targets travel in straight lines at uniform velocity over the environment. Note, however, that the algorithm is adaptable to any known emission process and any known distribution of target emission angles.

We give a brief sketch of the algorithm, leaving specifics to the source. Krishna et. al. divide the environment into a lattice of cells. Each sensor computes, from target statistics, the expected escape time of a target and the expected detection rate, for each of its cells reachable from its current position in at most to T time instants. These cells are inserted into a space-time tree, from which an optimal path is computed, penalizing paths corresponding to overlapping detections. This indicates implicit coordination among sensors. In the absence of such coordination, finding the optimal path in the tree would be combinatorially hard.

Extending Krishna's Algorithm

Krishna's algorithm considers only sensors with a rectangular Field of View (FOV), which is no longer the case when obstacles are introduced into the environment. We see this with Fig.1, wherein the rectangular FOV *ABCD* of *S* is truncated by the obstacle *OO'* into the polygon *EBCDFO'O*. Thus, the introduction of obstacles necessitates the re-computation of transit times over polygonal FOV's.

An analytic solution to the problem takes the following shape. Note that when targets have uniform velocity, finding the expected transit time within the polygon differs from the expected length of a chord in the polygon by a constant factor; as a result, it is sufficient to determine the latter.

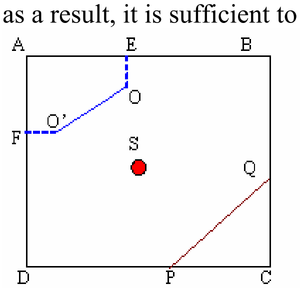

Fig.1 Estimating the transit time for a polygonal cell. *S* is the sensor, *OO'* the obstacle, *E* and *F* the feet of perpendiculars from *O* and *O'* to *AB* and *AD*. *P* and *Q* are sample points in the FOV *EBCDFO'O*.

With this in mind, consider the polygonal FOV $R = A_1A_2..A_n$; note that any target traveling in a straight line will enter one of the edges and leave by another. More specifically, suppose that the entering edge, A_iA_{i+1}, is given by the equation $y = ax + b, x_1 \le x \le x_2$ and the leaving edge A_jA_{j+1} by $y = a'x + b', x_1' \le x \le x_2'$. Note that when $i = n, i+1 = 1$ .Choose a sample point $P_1 = (x_3, y_3)$ on the A_iA_{i+1} and a sample point $P_2 = (x_4, y_4)$ on A_jA_{j+1} such that the P_1P_2 lies entirely within R ; the distance between them is

$$|P_1P_2| = \sqrt{(x_4 - x_3)^2 + (a'x_4 - ax_3 + b' - b)^2} \qquad [3.1]$$

The probability that a target entering R at P_1 will exit via A_jA_{j+1} is $\frac{\varphi(P_1)}{\pi}$, where

$$\varphi(P_1) = \arctan(slope(P_1A_{j+1})) - \arctan(slope(P_1A_j)) \qquad [3.2]$$

If $P_1 = (x_1, y_1)$, then $\varphi(x_1, y_1)$ is understood to mean $\varphi(P_1)$.

Thus, the expected transit time edge of a target entering by A_iA_{i+1} and the leaving by A_jA_{j+1}

$$< T(A_iA_{i+1}, A_jA_{j+1}) >= \frac{\int_{x_1}^{x_2} \varphi(x_3, ax_3 + b) \int_{x_1'}^{x_2'} D dx_3 dx_4}{\pi |A_iA_{i+1}| |A_jA_{j+1}|}$$

[3.4]

where D is the right side of Eq.[3.1] . Note that when $i = j$, expected transit time is zero, as $\varphi(x_3, ax_3 + b)$ is zero everywhere on the region of integration.

Let $S = \{(i, j) : 1 \le i < j \le n\}$; then the expected transit time across R is

$$< T(R) >= \frac{\sum_{(i,j) \in S} < T(A_i A_{i+1}, A_j A_{j+1}) >}{|S|} \quad [3.6]$$

Note that it is not possible to evaluate [3.4] in closed form, and a series expansion solution would be prohibitive to implement. Conequently, we are forced to use a numerical approximation scheme to determine the transit time across R. Choose two random points on different edges of R, as for example P and Q in Fig.1; determine the distance PQ. The mean value of this distance over a very large number, say a million such pairs, will give a fairly accurate value of the mean transit time.

B Parker's Algorithm

With no predetermined knowledge of target statistics [12], Parker uses the concept of force to achieve close-to-optimal performance. There are two types of forces discussed, sensor-target force and sensor-sensor force, which have a piecewise-linear profile with respect to target-sensor and inter-sensor distance respectively. The objective of the former is to attract sensors to targets within their FOV, and the latter to repel away nearby sensors to reduce the number of overlapping detections. Both these forces fail to operate beyond a threshold distance. Further, Parker suggests weighting these force vectors manually to fine-tune the algorithm's performance.

Our endeavor has been to extend Parker's algorithm in the presence of obstacles. While Parker has taken note of obstacles in her paper, she nonetheless assumes that sensors can see over them, which we hold impossible. In addition, while Parker does mention a repulsive sensor-obstacle force, she leaves the exact details unspecified. We experimented with various sensor-obstacle force profiles, and settled on one identical to her sensor-sensor repulsive force, reproduced in Fig.2, where dr_1 is one of the cutoff points for the sensor-sensor force referred to in [12].

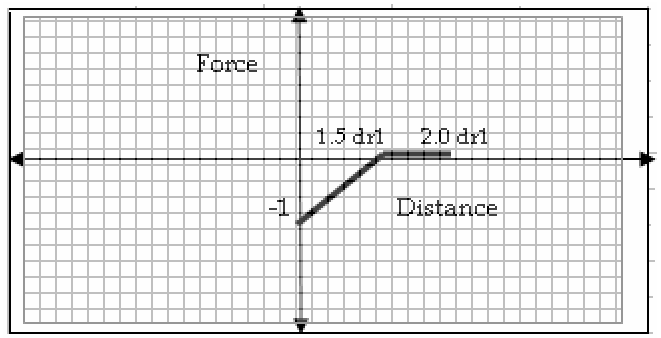

Fig.2 Sensor-Obstacle force for Parker's algorithm. dr_1 is the cutoff point for the sensor-sensor force in [12].

C Sukhatme-Jung algorithm

The keynote of the Sukhatme-Jung algorithm [8] is that the environment can be partitioned into a set of disjoint regions by topological landmarks. Two parameters, the sensor density and the target density, computed for each region, govern the algorithm, using the heuristic requiring sensors to navigate toward regions with higher target density. This has the additional consequence of making sensors move toward the center of gravity of the targets they detect. This heuristic, combined with the coarse deployment strategy, which optimizes inter-region sensor navigations, forms the main body of the algorithm. Sensors cooperate explicitly by broadcasting target information among them. Due to the fact that the algorithm incorporates obstacles directly in its finding of regions, we left it unchanged.

D Summary

The above-mentioned algorithms and the assumptions they make are summarized in Table 1.

Algorithm	**Assumptions**	**Cooperation Mode**
Krishna et.al.	1.Target emission statistics known 2. Distribution of emission angles known	Implicit, by penalties for overlapping detections
Parker	1.Target emission statistics unknown 2.Distribution of emission angles unknown	Implicit, by inter-sensor force
Sukhatme-Jung	Presence of topological landmarks	Explicit, by broadcasting of packets among sensors

Table 1 A comparison of target detection algorithms

IV SIMULATION AND RESULTS

The above algorithms were coded and tested on a Pentium 4, 2.1 GHz system on Linux (Fedora Core-3). We used C++ /Qt for developing the programs and the GUI.

Standard Test Case

The test environment consists of a square of side 600 pixels. Four sensors are initially positioned close to the vertices of the environment, and one at the center, each with FOV equal to 10% of the environment size, which is 60 pixels. Five obstacles with randomly chosen lengths, orientations and starting positions are distributed over the environment; two target sources are introduced, one to its top and the other to its left. In the known case, the target sources obey Poisson statistics with $\lambda = .025$, and the angles of emission are uniformly distributed on $[0, \pi]$ with respect to the x-axis. For the unknown case, the targets randomly pick an angle in $[0, \pi]$ at each time instant, and move ahead in the corresponding direction with uniform velocity. The standard test case is depicted in Figs.2a, 2b and 2c, and tabulated in Table 2(a).

Extended Test Cases

The environment remains the same as the standard test case. Six variations of the standard test case were tried out:

1. Increasing the FOV to 75 pixels
2. Decreasing the FOV to 30 pixels
3. Increasing the number of sensors to 7
4. Increasing the number of obstacles to 8
5. Removing obstacles altogether
6. Increasing the number of target sources to 4

These are tabulated in Tables 2(b)-2(f) in the same order.

B Simulation Snapshots

Figures 2a, 2b and 2c are sample snapshots of the sensor trails produced by the algorithms.

C Test Results

Each of the algorithms above was run by a script 50 times in both the known and unknown case, and the results averaged to produce Tables 2a to 2f. The comparisons have been done for various numbers of target sources, obstacles, sensors and their fields of vision. We have used the fraction of single detections as the index for performance, as a higher percentage of double or triple detections indicate a large overlap in areas seen by the sensor, and thus poor sensor coordination.

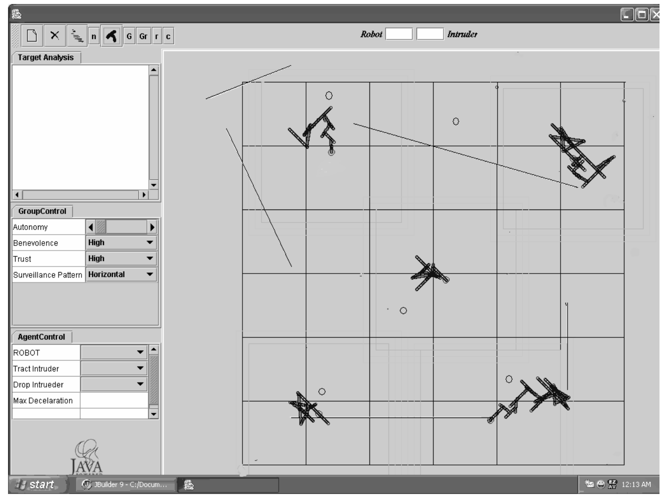

Fig.2a Snapshot of Krishna's algorithm in the known case. The sensor trails are the polygonal lines, and the black circles are targets.

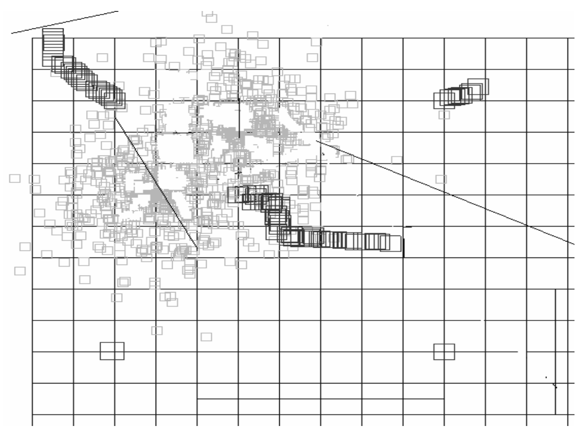

Fig.2b Parker algorithm snapshot in the unknown case, the large black squares indicating the sensor trails, and the grey squares the targets.

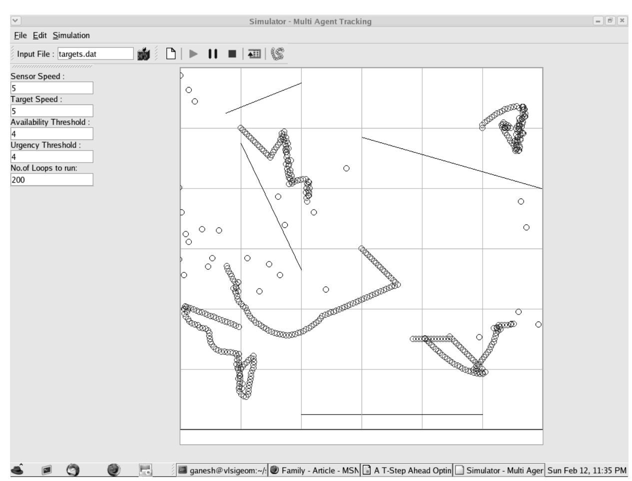

Fig.2c Sukhatme-Jung algorithm in the known case, the curved lines indicating the sensor trails, and the small black circles indicating the targets.

The first column of each table is the algorithm; Sukhatme-Jung is denoted by SJ. (K) / (U) refers to the known / unknown case respectively. The second column refers to the fraction of undetected targets, computed as the number of targets not detected by any sensor divided by the total number of targets within the surveillance area, at each iteration of the algorithm. This value is averaged over the number of iterations for every run of the algorithm and further averaged across the 50 runs executed by the script for the respective known and unknown cases. In the same fashion, the remaining columns refer to the fraction of targets detected by exactly 1, 2 and 3 sensors. As mentioned before, we require the number of targets detected by exactly one sensor (the value in column 3) to be high and all others to be low.

D Discussion

From the table it is evident that Krishna's algorithm performs best in the known case, and Sukhatme-Jung in the unknown. We also note that Parker's performance in the unknown case is better than Krishna's, and worse in the known. The superior performance of Krishna's algorithm in known cases is due to its modeling of target statistics and placing sensors in locations best estimated to detect targets, and that of Sukhatme-Jung due to its explicit cooperation strategy. Subtler reasons could be present which lead to a difference in performance; these require a much more careful study of the algorithm in question, possibly by theoretical approaches.

Questions may be raised about the need for algorithms where prior statistics of targets are known. This situation is analogous to that in mobile robotics where optimal planning algorithms exist in a known environment, vis-à-vis navigation algorithms that operate in an unknown one but are suboptimal. As much as optimal planning algorithms have become indispensable in mobile robotics, we feel that algorithms which can model target statistics to be incorporated while computing sensor paths are of significant utility. For example, traffic and vehicular accidents on a highway are often modeled by a Poisson distribution, and the flow of people across a square in a city's center with malls around that requires surveillance by a two-dimensional queue. Note that the strategy in Krishna's algorithm is extendible to any distribution of targets emissions and statistics.

Algorithm	**0**	**1**	**2**	**3**
Krishna(K)	0.46	0.51	0.03	0.01
Parker(K)	0.67	0.33	0.03	0.01
SJ(K)	0.46	0.49	0.07	0.00
Krishna(U)	0.76	0.20	0.04	0.00
Parker(U)	0.55	0.45	0.04	0.00
SJ(U)	0.19	0. 59	0.21	0.00

Table 2(a) Comparison of target detection performance by various algorithms. Standard test case: 5 sensors, 5 obstacles, and 2 target sources. FOV is 10% of the side of the square. K indicates the known case, U the unknown. SJ is Sukhatme-Jung. The headings k=0, 1, 2, 3 specify the fraction of targets detected by precisely k sensors.

Algorithm	**0**	**1**	**2**	**3**
Krishna(K)	0.30	0.60	0.10	0.01
Parker(K)	0.58	0.40	0.02	0.00
SJ(K)	0.27	0.58	0.15	0.00
Krishna(U)	0.55	0.40	0.05	0.00
Parker(U)	0.19	0.71	0.10	0.00
SJ(U)	0.14	0.65	0.20	0.01

Table 2(b) Increasing the FOV to 75 pixels

Algorithm	**0**	**1**	**2**	**3**
Krishna(K)	0.56	0.43	0.01	0.00
Parker(K)	0.85	0.15	0.02	0.00
SJ(K)	0.75	0.24	0.01	0.00
Krishna(U)	0.83	0.15	0.02	0.00
Parker(U)	0.75	0.25	0.00	0.00
SJ(U)	0.50	0.38	0.12	0.00

Table 2(c) Decreasing the FOV to 30 pixels

Algorithm	**0**	**1**	**2**	**3**
Krishna(K)	0.40	0.56	0.03	0.01
Parker(K)	0.58	0.38	0.04	0.00
SJ(K)	0.40	0.55	0.18	0.07
Krishna(U)	0.68	0.26	0.05	0.01
Parker(U)	0.27	0.58	0.05	0.00
SJ(U)	0.10	0.55	0.34	0.01

Table 2(d) Increasing the number of sensors from 5 to 7

	0	**1**	**2**	**3**
Krishna(K)	0.43	0.52	0.03	0.01
Parker(K)	0.46	0.39	0.15	0.00
SJ(K)	0.35	0.52	0.20	0.03
Krishna(U)	0.75	0.21	0.04	0.00
Parker(U)	0.14	0.70	0.15	0.01
SJ(U)	0.18	0.60	0.20	0.02

Table 2(e) Increasing the number of obstacles from 5 to 8

	0	**1**	**2**	**3**
Krishna(K)	0.31	0.65	0.04	0.00
Parker(K)	0.45	0.20	0.30	0.05
SJ(K)	0.20	0.54	0.25	0.01
Krishna(U)	0.60	0.35	0.05	0.01
Parker(U)	0.25	0.45	0.30	0.00
SJ(U)	0.20	0.60	0.2	0.01

Table 2(f) Removing all obstacles

V CONCLUSIONS

The motivation for this paper was to compare multi sensor based target tracking algorithms reported in the literature. While authors have compared their most recent method with their previous ones [12, 19] there have not been comparisons across their best performing methods. The following are concluded based on the experimental results obtained by comparing three reported algorithms:

1. The current algorithm is consistently better than Parker's in the known case due to the predetermined target statistics. Its performance improvement over Sukhatme-Jung is marginal, around 3% in the known case. Often the performances of Krishna's and Sukhatme-Jung algorithms are similar as seen from the tables even for known cases, attributed to the latter's explicit cooperation. Krishna's algorithm is best used over others if target statistics are known and can be modeled. In addition, due to the implicit cooperation among sensors in Krishna's algorithm, there is minimal broadcast of packets, and hence reduced bandwidth requirements. Consequently, it lends itself to use in sensor networks of limited bandwidth, or when the number of targets is too large for broadcasting of target information.
2. Sukhatme-Jung's algorithm achieves good results in both the known and the unknown cases due to the explicit broadcasting of tracking information among sensors, which minimizes overlapped detections. When the number of targets is very large, then explicit communication of target positions and detection information consumes considerable bandwidth. Thus the algorithm is best used in distributed systems with large bandwidth, or a small number of targets i.e. when broadcast of packets is limited.
3. Parker's algorithm finds utility when target statistics are unknown and implicit cooperation is the rule due to very limited bandwidth availability.

REFERENCES

[1] S. A. Stoerer, P. E. Rybski, M. D. Erickson, M. Wyman, M. Gini, D. F. Hougen, N. Papanikolopoulos, "A Robot Team for Exploration and Surveillance: Design and Architecture," Proceedings of the International Conference on Intelligent Autonomous Systems 6, pp. 767-774, Venice, Italy, July 2000.

[2] H R Everett, G A Gilbraeth and R T Laird, "Coordinated control of multiple security robots", *Proceedings of SPIE Mobile Robots,* 292-305, 1993

[3] D Schulz; W Burgard, D Fox and A Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association", *IEEE International Conference on Robotics and Automation*, 1165-1170, 2001

[4] D Schulz and W Burgard, "Probabilistic state estimation of dynamic objects with a moving mobile robot", *Robotics and Autonomous Systems,* 2001.

[5] B. Kluge, C Kohler and E Prassler, "Fast and robust tracking of multiple objects through a laser range finder", *IEEE International Conference on Robotics and Automation*, 1165-1170, 2001

[6] K Madhava Krishna and P K Kalra, "Detection tracking and avoidance of multiple dynamic objects", *Journal of Intelligent and Robotic Systems*, 33(4): 371-408, 2002

[7] L Parker, "Cooperative robotics for multi-target observation", *Intelligent Automation and Soft Computing,* 5[1]:5-19, 1999

[8] B Jung and G.S. Sukhatme, "A Region-based approach for Cooperative Multi-Target Tracking in a Structured Environment", Proc., *Proceedings of International Conference on Intelligent Robots and Systems,* 2002

[9] S Poduri and G S Sukhatme, "Constrained Coverage for mobile sensor networks", *IEEE ICRA,* 165-171, 2004

[10] B Horling, R Vincent, R Miller, J Shen, R Becker, K Rawlins, and V Lesser, "Distributed Sensor Network for Real Time Tracking", In *Proceedings of the 5th International Conference on Autonomous Agents*:. 417-424, 2001.

[11] B Horling, R Miller, M Sims, and V Lesser, "Using and Maintaining Organization in a Large-Scale Distributed Sensor Network", In *Proceedings of the Workshop on Autonomy, Delegation, and Control, (AAMAS 2003)*.

[12] L Parker, "Distributed algorithms for multi-robot observation of multiple moving targets", *Autonomous Robots,* 12, 3 (2002)

[13] A. J. Briggs, "Efficient geometric algorithms for robot sensing and control", Ph.D. thesis, Cornell, 1995

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

[15] K Madhava Krishna, H Hexmoor and S Sogani, " An optimal A T-Step Ahead Constrained Optimal Target Detection Algorithm for a Multi Sensor Surveillance System", in *Proceedings of International Conference on Intelligent Robots and Systems,*2005.