

Research Article

Market-Based Approach to Mobile Surveillance Systems

Ahmed M. Elmogy,¹ Alaa M. Khamis,² and Fakhri O. Karray³

¹ *Computers & Control Engineering Department, Tanta University, Tanta 31111, Egypt*

² *Engineering Sciences Department, Suez Canal University, Suez 43721, Egypt*

³ *Electrical and Computer Engineering Department, Waterloo University, Waterloo, ON, Canada N2L 3G1*

Correspondence should be addressed to Ahmed M. Elmogy, aelmogy@pami.uwaterloo.ca

Received 30 June 2012; Revised 13 September 2012; Accepted 14 September 2012

Academic Editor: Jorge Dias

Copyright © 2012 Ahmed M. Elmogy et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The active surveillance of public and private sites is increasingly becoming a very important and critical issue. It is, therefore, imperative to develop mobile surveillance systems to protect these sites. Modern surveillance systems encompass spatially distributed mobile and static sensors in order to provide effective monitoring of persistent and transient objects and events in a given area of interest (AOI). The realization of the potential of mobile surveillance requires the solution of different challenging problems such as task allocation, mobile sensor deployment, multisensor management, cooperative object detection and tracking, decentralized data fusion, and interoperability and accessibility of system nodes. This paper proposes a market-based approach that can be used to handle different problems of mobile surveillance systems. Task allocation and cooperative target tracking are studied using the proposed approach as two challenging problems of mobile surveillance systems. These challenges are addressed individually and collectively.

1. Introduction

One of the most active research topics is how to automate surveillance tasks based on mobile and fixed sensors platforms [1]. Many benefits can be anticipated from the use of multisensor systems in surveillance applications [2, 3], such as decreasing task completion time and increasing mission reliability. Generally, monitoring of public and private sites is the main application of multisensor surveillance systems. The primary objectives of the surveillance systems are to provide the information that makes the system able to understand and predict the actions and the interactions of the observed objects in order to carry out different tasks. Examples of these tasks would include target search, identification, and tracking. Advanced surveillance systems encompass spatially distributed mobile and static sensors in order to provide effective monitoring of persistent and transient objects and events in a given area of interest (AOI) [4]. Mobile surveillance systems incorporate self-organized networks of mobile sensing nodes of different modalities, data and information fusion nodes, acting nodes, and control nodes. These self-organized nodes can collaboratively and

continuously sense within the volume of interest, as well as physically manipulate and interact with it. The main goal of the surveillance system is to adjust the sensing conditions for improved visibility, and thereby improve performance [5]. In such setting, surveillance is a complex problem posing many challenging problems.

This paper presents a market-based approach to mobile surveillance systems. The goal is to develop an approach that efficiently distributes tasks among the mobile sensor team to achieve the surveillance mission. Such approach will support the operation of the mobile sensors so that they can collaboratively perform tasks such as detecting and tracking moving targets. In order to maximize the effectiveness of the mobile sensor team collaborating as a group, the action of every mobile sensor should consider the contribution of its teammates towards the mission objectives. How to accomplish this is a complex problem, which is currently an active area of research [6, 7]. This requires tackling some of the challenging problems of mobile surveillance that have not been investigated collectively in the past. These problems include, but are not limited to, task allocation, mobile sensor deployment, cooperative object detection and tracking, and

decentralized data fusion [8]. This paper tackles two of these problems: task allocation and cooperative detection and tracking. These problems are tackled individually and collectively.

A great number of task allocation approaches have been reported in the literature [9–12]. However, many aspects have to date been given little attention. Examples of these aspects are complex task allocation and constrained task allocation. Complex tasks are those tasks which can be decomposed into different subtasks. Working with a complex task is guaranteed to produce a more efficient solution for the task allocation problem [13]. Thus, complex task allocation [14] is considered in this paper as a part of the design of the mobile surveillance framework.

The target-tracking problem [15] is concerned with estimating the state of a target or targets in some areas of interest based on some measurements from the stationary sensors, in most of the cases. However, using stationary sensors has many constraints which force the use of multiple mobile sensors dynamically moving over time especially in surveillance applications. Thus, target detection and tracking is also tackled in this paper using mobile sensors in the design of the mobile surveillance framework.

The rest of this paper is organized as follows. Section 2 presents the formulation of mobile surveillance problems tackled using the proposed market-based framework described in Section 3. Section 4 summarizes the conducted experimental results tackling complex task allocation and target detection and tracking. Finally, conclusion and future work are summarized in Section 5.

2. Problem Formulation

This section highlights the problem formulation for the major components of the proposed mobile surveillance framework as shown in Figure 1 tackling two main challenges—complex task allocation—and target detection and tracking.

2.1. Complex Task Allocation. This subsection provides problem definition for both simple and complex task allocations and formulation for only complex task allocation.

Definition 1 (simple task allocation). Given a set of mobile sensors S each looking for one task and a set of tasks T each requires one sensor. The simple task allocation can be defined by a function $A : T \rightarrow S$, mapping each task to a mobile sensor in order to be executed. Similarly, S_T is the set of all allocations of tasks T to the team of sensors S .

Definition 2 (complex task allocation). Given a set of mobile sensors S and a set of tasks T . let $G \subset T$ is a group or a bundle of tasks that is decomposable into other tasks $M \in G$. The complex task allocation can be defined by a function $B : M \rightarrow S$, mapping each subtask to a mobile sensor to be responsible of completing it. Equivalently, S_M is the set of all allocations of subtasks M to the team of sensors S .

For both simple and complex task allocation, the goal is to assign sensors to tasks so as to maximize overall expected performance, taking into account the priorities of the tasks and the skill ratings of the sensors. Appropriate functions are needed to map possible task outcomes into revenue values and to map possible schemes for performing the task into cost values. Generally, mobile sensors receive revenue and incur costs for accomplishing a specific team task. A mobile sensor can also receive revenue from another sensor in exchange for goods or services.

The problem of task allocation can be formulated in many ways. Given our surveillance application domain, it can be formulated as follows.

- (1) AOI: two-dimensional, bounded area of interest.
- (2) S : a team of mobile sensors $s_i, i = 1, 2, \dots, n$. It is assumed that each sensor carries sensors (such as cameras, sonar, and laser range finders).
- (3) T : a set of tasks $t_j, j = 1, 2, \dots, m$.
- (4) U : a set of sensors utilities, u_{ij} is the utility of sensor i to execute task j .

For single sensor task, the problem is to find the optimal allocation of sensors to tasks, which will be a set of sensors and tasks pair [16]:

$$(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k) \quad \text{for } 1 \leq k \leq m. \quad (1)$$

For the general case, the problem is to find the optimal allocation of a set of tasks to a subset of sensors, which will be responsible for accomplishing it [13]:

$$A : T \rightarrow S. \quad (2)$$

Each mobile sensor $s \in S$ can express its ability to execute a task $t \in T$, or a bundle of tasks $G \subseteq T$ through bids $b_s(t)$ or $b_s(G)$. The cost of a bundle of tasks can be simply computed as the sum of costs of the individual tasks:

$$b_s(G) = \sum_{k=1}^f b_s(t_k) \{t_k \in G\}, \quad (3)$$

where f is the number of tasks of the bundle G . The group's assignment determines the bundle $G \subseteq T$ of tasks that each mobile sensor $s \in S$ receives. These bundles can be characterized as follows:

$$\beta = \{(G_1, G_2, \dots, G_w) \mid G_{k_1} \cap G_{k_2} = \emptyset, \bigcup G_w = T\}. \quad (4)$$

The global objective function can vary depending on the requirements of the system or the preferences of the designer. The most common global objective is to minimize the sum of the team member costs, which can be described mathematically as follows:

$$C(A) = \sum_{s=1}^n b_s(G_s), \quad (5)$$

where $C(A)$ is the total required cost for executing the allocation A , and G_s is the bundle of tasks that is won by sensor s .

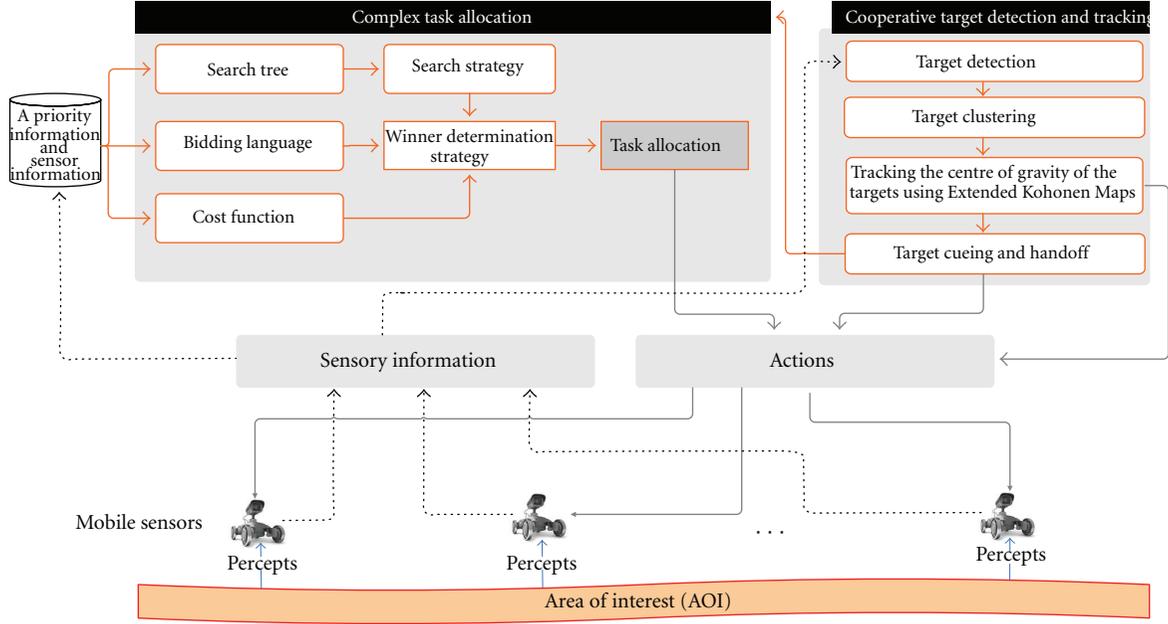


FIGURE 1: Market-based framework for mobile surveillance systems.

Though the mobile sensor team members may have well-defined cost or utility functions, these functions still rely on having accurate models of the world state and may require computationally expensive operations. When there are multiple goal locations like in surveillance application, determining the cost to perform even one task can require solving multiple path planning problems. Thus, an instance of the traveling salesman problem (TSP) [17, 18] might be used. In this work, we are using a shortest sequence planning algorithm (SSP) [19] in order to find the minimum cost path for each mobile sensor given the tasks' locations. In this algorithm, an agent is tasked with visiting a set of points and the goal is to find in which order it should visit these points with minimum traveling distance without going back to its original place, thus accomplishing the required tasks with near optimal system performance.

2.2. Target Detection and Tracking. This section provides the general formulation of target detection and tracking problem. With some variations, the same notation and terminology for target tracking as those introduced in [20] is used in this paper. Consider the following.

- (1) AOI: a two-dimensional, bounded obstacle free area of interest.
- (2) O : a set of n moving targets, $o_j, j = 1, 2, \dots, n$. The position of o_j at time tt will be denoted by $o_j(tt)$. Targets cannot leave the area of interest, that is, for all tt $o_j(tt) \in \text{AOI}$.
- (3) S : a team of m mobile sensors. It is assumed that each sensor carries sensors (laser range finders) to be able to detect the targets.
- (4) $\text{SC}(s_i, tt)$: the sensor coverage, which is the subset of AOI observable by mobile sensor s_i at time tt . This

region varies as the sensor s_i moves inside AOI, but its shape is supposed not to vary. In this paper, an omnidirectional sensing shape for each mobile sensor is assumed.

- (5) $\text{OM}(tt)$: $m \times n$ observation matrix. A sensor s_i is said to be observing a target when the target is within s_i 's sensing range:

$$\text{OM}(t) = \text{om}_{ij}(tt) = \begin{cases} 1 & \text{if sensor } i \text{ is observing target } j, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The goal is to develop an algorithm that maximizes the average coverage AC which can be defined as:

$$\text{AC} = \frac{\sum_{tt=1}^{t_e} \sum_{j=1}^n g(\text{OM}(tt), j)}{t_e}, \quad (7)$$

where

$$g(\text{OM}(tt), j) = \begin{cases} 1 & \text{if } \exists i \text{ such that } \text{om}_{ij}(tt) = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

In other words, the problem requires maximizing the number of targets that are observed by the mobile sensors. t_e is the execution time of the algorithm. It is assumed that

$$\bigcup_{s_i \in S} \text{SC}(s_i, tt) \ll \text{AOI} \quad \text{for any } tt. \quad (9)$$

Average energy saving over time AE is also used as an evaluation metric:

$$\text{AE} = \sum_{i=1}^{t_e} \frac{st}{n} * \frac{1}{t_e} * 100, \quad (10)$$

where st is the number of asleep trackers, n is the total number of the trackers, and t_e is the execution time of the algorithm.

The energy saving can be measured by considering the number of *busy* and *asleep* trackers over time.

Moreover, it is assumed that the maximum speed of the targets is smaller than the maximum speed of the sensors, in order to give the sensors the ability to reach the targets. If the targets could always move faster, then they could always evade the sensors and the problem becomes insoluble for the mobile sensor team. Finally, it is assumed that the sensors have a communication mechanism, which allows them to send and receive messages in a broadcast mode.

3. Proposed Market-Based Approach to Mobile Surveillance Systems

Market-based approaches have received significant attention and are growing very fast in the last few decades especially in multiagent domains [7, 21]. These approaches are considered as hybrid approaches that combine the centralized and distributed strategies (i.e., market-based approaches have elements that are centralized and distributed). The decision to use auctions in this paper comes from the existence of several desirable properties of auction approaches [22, 23] such as efficiency, robustness, and scalability.

3.1. Single-Shot and Combinatorial Auctioning. So far, researchers have studied single-item auctions at which items are auctioned off one at a time [24]. However, if there are strong synergies between the items of the bidders, highly suboptimal team solutions can be resulted from single-item auctions [25]. Two items are said to exhibit positive or negative synergy for a bidder if the combined bid of this bidder on these two items is larger or smaller than the sum of its individual bids on each item separately.

An example of that is shown in Figure 2. There is positive synergy between AOI-1 and station AOI-2 for the mobile sensor S_1 because they are close to each other. The mobile sensor S_1 can reach AOI-2 with a short distance (5 m) after it has reached AOI-1 (5 m). So, the sum of the single bids of S_1 on AOI-1 and AOI-2 ($12 = 5 + 7$) is more than the combined bid of S_1 on both areas ($10 = 5 + 5$). On the other hand, there is a negative synergy between station 1 and AOI-1 for S_1 because they are on opposite sides of the S_1 , and hence the mobile sensor S_1 can therefore reach either one of the areas only with a long travel distance after it has reached the other one.

Generally speaking, combinatorial auctions attempt to overcome the disadvantages of single-item auctions by allowing bidders to bid on bundles of items [13, 25]. If a bidder wins a bundle, they win all the items in that bundle.

3.2. Auction Design. The task allocation approach proposed in this paper imitates the auction process of buying and selling services through bidding. Sellers or auctioneers are responsible of processing the bids sent by buyers or bidders and determining the winning bidder. In this subsection, a

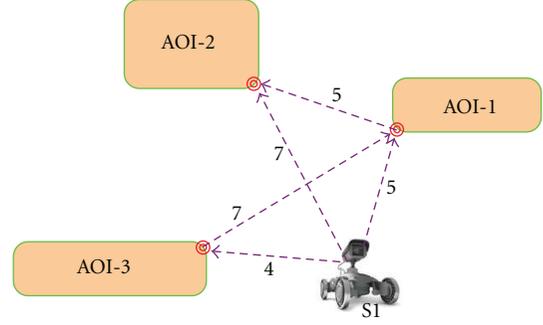


FIGURE 2: Single-shot and combinatorial auctioning.

brief idea about how to design the auction process will be based on maximizing a utility function.

3.2.1. Utility Function. Utility is the quality or state of being useful. For task allocation problem, utility is a satisfaction (value of profit) derived by a mobile sensor s from accomplishing a task t .

Given a mobile sensor s and a task t , if s is capable of executing t , utility can be defined [13] on some standardized scales as

$$u = p(t) - d(t), \quad (11)$$

where $p(t)$ is the total payment it receives after executing the task t , and $d(t)$ is the total distance it travels to reach the task. The priorities of tasks to be executed should be taken into account while designing the task allocation framework. Our objective is to find the optimal assignment of tasks T to sensors S in order to minimize cost and thus maximize the overall utility. Consequently, system performance is ideally optimized. Thus, the goal is to assign sensors to tasks so as to maximize the cost as we assume that there is no payment received after executing the task.

3.2.2. Search Tree. Most of task allocation approaches treated tasks as atomic units [23, 26–28]. Thus, allowing only static description for each task and so the only degree of freedom is determining to which sensor the task will be assigned. While this description is fine in case of simple tasks, it is not with complex tasks. Given the bid submitted, search over all possible allocations can be used as a winner determination strategy. In this case, a search tree can be used as a better description for the tasks. In this tree, mobile sensor team members are permitted to bid on nodes representing varying levels of task abstraction, thereby enabling hierarchical planning, task allocation, and optimization among the team members.

In our work, the complex tasks to be allocated are structured as an ordered tree. The nodes (elements of the tree) that are immediately greater than a node are called its children, while the node that is immediately less is its parent (if it exists). Any node less is an ancestor and any node greater is a descendant. A node with no ancestors is a root. The partial ordering represents distance from the

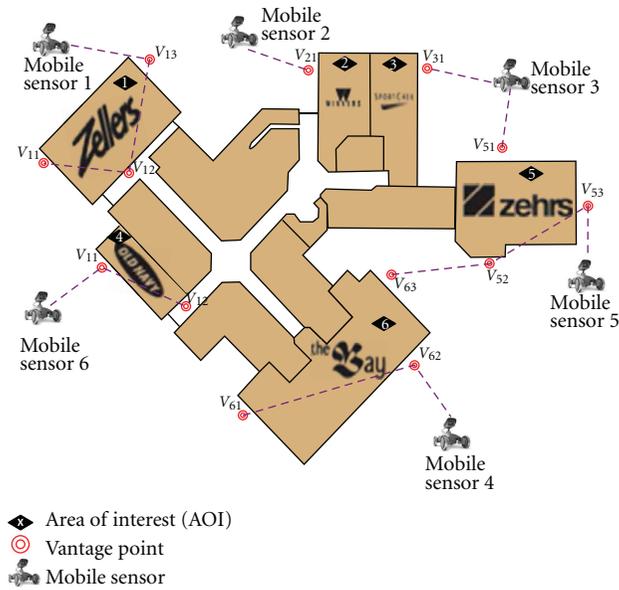


FIGURE 3: Surveillance scenario.

root, and the well-ordering requirement prohibits any loops or splits below a node (that is, each node has at most one parent, and therefore at most one grand-parent, and so on). In other words, if xFz , then there is exactly one yFz such that xFy and there is nothing between x and y . Perhaps the best way to illustrate the mechanics of the task tree is through a simple example. Figure 3 shows a surveillance scenario, which represents a shopping mall in the city of Waterloo, Ontario, Canada. The mission is to monitor a set of areas of interest (AOIs) such as Zellers, Old Navy, the Bay, Zehrs, Winners, and Sports Check. For the small areas (like Old Navy, Winners, and Sports Check), only one or two vantage points are enough to achieve the monitoring task, while three points are required to visit for the large areas (like Zellers, the Bay, and Zehrs).

The AND/OR task tree is constructed by decomposing the surveillance mission as complex task into two subtasks (scan large areas and scan small areas) as illustrated in Figure 4. Accomplishing the requested mission requires achieving both scan large and small areas. In other words, these two subtasks are related to each other by the logical operator AND, which means that both tasks are required to be executed. The subtask (scan large areas) is in turn decomposed to other simpler tasks such as scan AOI-1, AOI-5, and AOI-6. The simple tasks can be executed by one of two plans, which contain the most primitive tasks. For example, to scan AOI-1, Plan-1 or Plan-2 can be chosen. These two alternative covering plans are computed based on the minimum traveling distance and the second minimum traveling distance. Plan-1 contains a list of primitive tasks (Goto V_{13} , Goto V_{12} , Goto V_{11}) that must be executed sequentially. Similarly, Plan-2 contains same primitive task but with different order.

This decomposition is done initially by an operator or by the selected initial auctioneer. Once the task tree

is constructed and the decomposition is complete, the auctioneer holds a task tree auction, distributing tasks among the team and allowing other robots to use their own plans when appropriate. The auctions then proceed in rounds in which each mobile sensor holds a task tree auction (if it has any tasks) in a round-robin fashion.

3.3. Winner Determination Strategies. The winner determination strategy addresses how to optimally find the set of bids that maximize the bidder's revenue. In combinatorial auction, winner determination is NP-hard problem [29] as searching for all possible allocations of items to mobile sensors is computationally intractable and no approach will work in polynomial time.

Winner determination strategy is highly affected by the type of description of tasks to be allocated. As mentioned previously, the complex tasks to be allocated are represented as an ordered tree. A breadth-first search algorithm is used to find the task allocation solution from this task tree structure. Two organizational paradigms, namely, centralized and hierarchical allocation are used during the allocation process. These paradigms determine the roles, the relationships, and the structures, which govern the auction process. In centralized auctioning, an auctioneer holds a series of auctions in rounds to allocate the surveillance tasks to the mobile sensors in order to maximize the system utility. On the other hand, in hierarchical auctioning, the tasks are allocated initially to the mobile sensors via a central auctioneer. Each mobile sensor can hold auctions in rounds for the tasks it wins in the initial auction.

3.4. Fixed Tree Task Allocation. Consider a team of mobile sensors assembled to perform a particular task. Consider further that each mobile sensor is capable of executing one task at once, and each task can be accomplished by one sensor. The task information is continuously available to the mobile sensors team. Thus, the proposed approach in this case can be framed as iterated instances of ST-SR-IA (Single-Task Single-Robot Instantaneous Assignment). The goal of the team is to perform the task efficiently while minimizing costs. In the context of fixed task tree allocation, a set of constraints dictates that the whole auction mechanism is based only on one task tree, which is proposed by the operator or the auctioneer. The proposed algorithm allows using only one auctioneer from the start to the end of auctioning, and so considered as a centralized task allocation. It also allows changing the auctioneer during auctioning while considering only the plan of the original operator. In this case, our proposed mechanism can be seen as a hierarchical task allocation mechanism. Another constraint dictates that at most one node can be sold to each bidder per auction. This is because upon awarding one node to a bidder, the bid prices on other nodes become invalid due to the fact that bid prices are conditioned on the current commitments of each participant. This can be done by modeling each mobile sensor as self-interested agent which aims to minimize its individual cost and so maximize the whole team performance by following a greedy algorithm.

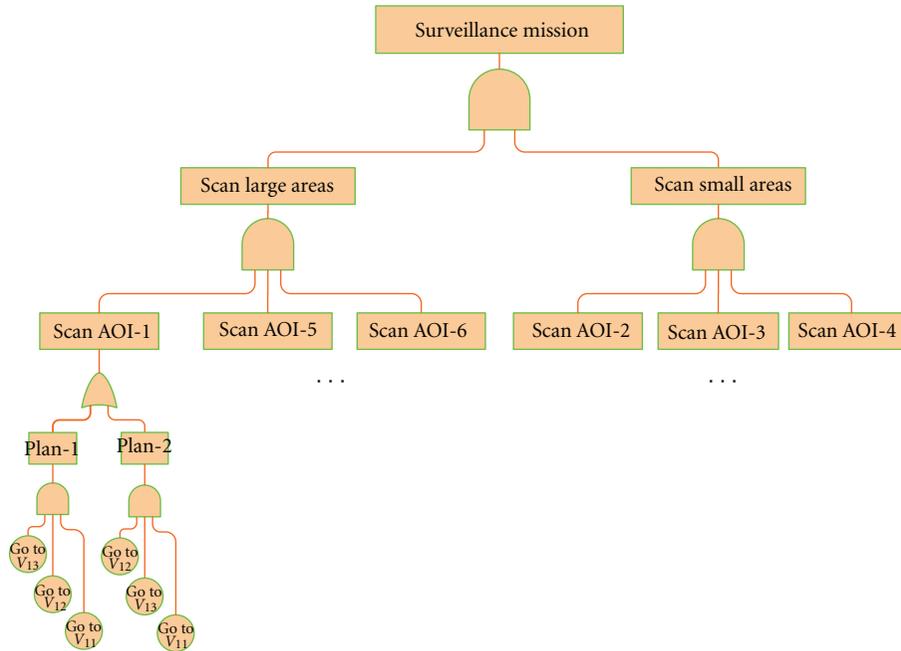


FIGURE 4: AND/OR task tree.

Each mobile sensor is either cooperating with other members of the team to achieve an outcome greater than that possible by each member alone or competing with other members to the required task at the lowest possible cost, thus eliminating waste and inefficiency. A system such as this can be highly seen in the economy and so many desirable characteristics from the market mechanisms might be used.

3.5. Dynamic Tree Task Allocation. The proposed fixed task tree allocation described in the previous section could be seen as an instance of decompose-then-allocate approach. The main drawback of this approach is that the cost of the final plan cannot be fully considered because the complex task is decomposed by the auctioneer without knowledge of the eventual task allocation. Also, backtracking is not allowed in this approach, and so any costly mistakes in the auctioneer decompositions cannot be rectified. Generally, the allocate-then-decompose method tries to avoid the drawbacks of the decompose-then-allocate method. However, there are still some disadvantages. Motivated by the drawbacks of both methods, we are proposing dynamic tree allocation to allow backtracking in order to recover the bad plans made by the auctioneers. The algorithm allows auctioning on all levels of abstraction of the mission task implemented by the task tree from the top to the bottom. Each mobile sensor evaluates its ability to execute the required task based on its plan not on the plan of auctioneer. Our proposed dynamic algorithm is either executed by allowing only one auctioneer (centralized allocation) or allowing different auctioneers (hierarchical allocation).

3.6. Proposed Market-Based Target Tracking Methodology. The main objective of this section is to give a detailed

description of the developed methodology for tracking multiple objects, which will be incorporated into the proposed mobile surveillance framework.

The proposed mobile surveillance framework requires knowledge of the current targets' positions as well as their future positions, in order to be able to determine the sensor assignments and states. In this context, the overall problem of tracking can be summarized as follows.

- (1) Detecting every target within a given scene.
- (2) Distinguishing and categorizing objects as obstacles or objects of interest.
- (3) Find the centre of gravity of targets within the scene using a hybrid subtractive- K -means clustering technique.
- (4) Tracking the centre of gravity of the targets within the scene using Extended Kohonen Maps.
- (5) Tracking the exit targets by the best suitable trackers using a cueing/handoff market-based approach.

Ideally, sensor team members would be able to passively detect nearby sensors and targets to ascertain their current positions. Many machine vision algorithms have been developed for this type of position calculation. Although vision systems have a very good resolution, they do not provide accurate distance information [30]. Also, vision-based systems frequently suffer from occlusion and sudden changes in illumination [31]. On the other hand, other types of sensors, such as laser range sensors [30–32], are able to obtain accurate distance and geometric information about the objects in their sensing range. Moreover, laser range sensors do not have the problems of occlusion, and

sudden changes in illumination, as with vision systems. The laser range sensors have attracted more attention in the field of detection and tracking [32, 33]. Thus, in this paper, we propose a detection algorithm that uses laser scanners to detect moving targets and objects. The outputs of the proposed algorithm are the positions of the targets and objects. Each sensor communicates the positions of targets and other objects within its sensing range to its sensor team members.

3.6.1. Target Clustering. In order to make our approach dependent on the distribution of the targets rather than the density of targets as in [20, 34], we chose to cluster the targets in the environment and then track the clusters' centers instead of tracking each target separately. Clustering the targets makes our approach energy efficient one because not all the trackers will be active at all the time. Out of the numerous available clustering techniques, two were selected: subtractive clustering and K -means [35] clustering techniques. The reason behind using K -means clustering technique is its high level of accuracy [36]. However, K -means clustering technique has the problem of selecting the initial locations of clusters and the number of clusters, which will affect the speed of convergence and the accuracy of the algorithm. So, we propose using the subtractive clustering technique to find the initial number and locations of the clusters' centers to be fed into the K -mean clustering technique in order to find the final exact locations of clusters' centers.

3.6.2. EKM-Based Tracking Algorithm. After clustering the detected targets which lie within the tracker sensing range. The center of gravity of the detected targets is reached by adopting a representation of the sensory input vector: $u_p = (\theta, d)$, where θ and d are the angle difference and distance between the tracker and the target, respectively. Each tracker uses an Extended Kohonen Map (EKM) [37] in order to reach the target. The Extended Kohonen Map is an extension of Kohonen Map [38] and is considered as one of the most famous unsupervised learning neural network. Extending the Kohonen Map, which is done by adding fan-in input weights to its input layer gives the map the ability to learn by supervised learning. However, this is not always the case; unsupervised learning could still be used. Each neuron in the EKM has a sensory weight vector $w_i = (\theta_i, d_i)$. Figure 5 shows the supervised learning process in the Extended Kohonen Map.

As shown, suppose we have trained an EKM on a set of examples $\langle x; y \rangle$, where x is a point in the input data space X and y is a point in the output data space Y . The input pattern x is presented to the network input layer to retrieve the corresponding output value, which will be an approximation to y . This is accomplished in the usual way, namely, by letting the fan-in weight vectors w_r compete on x , and by taking the fan-out weight vector z_{ne} of the winner ne (which has the closet fan-in weights to x) as the network output value. As mentioned above, ne is awarded the right to learn the input pattern x by moving closer to it. Also, the fan-out z_{ne}

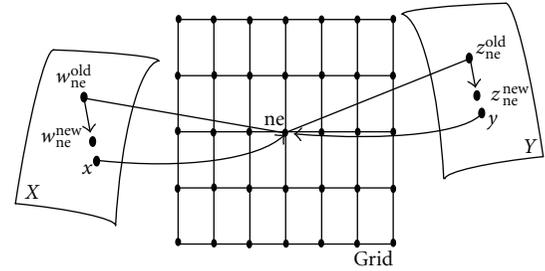


FIGURE 5: Supervised Extended Kohonen Map learning.

is awarded the right to learn the output pattern y by moving closer to it:

$$z_{ne}^{new} = z_{ne}^{old} + \alpha(y - z_{ne}^{old}), \quad (12)$$

where z_{ne}^{old} is the fan-out weights of the winner ne before learning, z_{ne}^{new} is the fan-out weights of the winner ne after learning, and α is the learning rate.

Unsupervised EKM has the problem of selecting the initial neurons' weights. Selecting the initial weights will affect the accuracy and speed of mapping from the space of sensory data to the space of motor commands for tracking purposes. So, in this paper, we propose to use the supervised EKM parameters (fan-in and fan-out) as the initial parameters of unsupervised EKM in order to increase the accuracy and speed of convergence of unsupervised Map. Thus, unsupervised EKM is used to get the function of mapping from space of sensory data to space of motor commands for tracking purposes.

3.6.3. Cueing/Handoff Market-Based Methodology. The mobile sensors used in this paper are assumed to be equipped with high-bandwidth communications and an array of sensors and actuators, which give the sensors the ability to achieve cooperative behavior at the group level. A cueing/handoff market-based method like the one in [6] is used to guarantee that there is only one mobile sensor that will respond to the help call coming from the sensor that detects a target about to exit its sensing range (or in its predictive tracking range). The mobile sensor that detects an exiting target will do the task of an auctioneer (i.e., every tracker can do the function of the auctioneer). This makes our algorithm more robust than methods that use only one coordinator because there is no central point of failure in this case. Using a cueing/handoff market-based algorithm will guarantee that the most suitable sensor will track the exit target. In other words, there is no need to explore the environment or to check the answered help calls as in [20]. This makes the proposed method a more energy-efficient one than the method in [20]. The cueing/handoff market-based method proceeds as follows.

- (1) Help cueing: the mobile sensor that needs help (detects an exiting target) broadcasts a help call to its teammate. One sensor can issue multiple help calls according to the number of targets about to be lost.

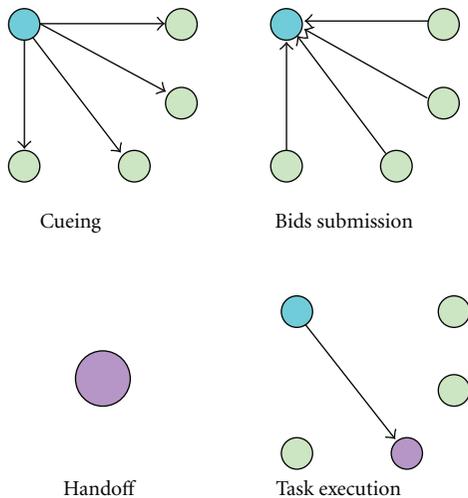


FIGURE 6: Cueing/handoff market-based algorithm.

Each help call includes the position of the target to be lost.

- (2) Bid submission: after each sensor receives the help call, they send their bids to the auctioneer (the sensor that issued the help call). The bids include the distances of the bidders to the target.
- (3) Close of auction: the auctioneer processes the bids, determines the winner, and notifies the bidders with a message which indicates who is the winner.
- (4) Task handoff: the winner will proceed by orienting itself to face the target and move with its maximum speed to put the target inside its sensing range.

The existence of the auctioneer does not mean that the proposed system is completely centralized. The system is still distributed and the function of the auctioneer is only to start the action of tracking. Once started, the tracker will do its task on its own, without any dependency on the auctioneer. Furthermore, there is no single point of failure because the auctioneer changes in the system according to the sensor that requests help (i.e., the sensor that requests help will be the auctioneer). The process of the cueing/handoff market-based algorithm shown in Figure 6 can be summarized as follows.

The cooperative tracking capability of a team of mobile sensors, each fitted with Extended Kohonen Maps, is utilized to maximize the coverage of multiple mobile targets. The main goal is to develop an energy-efficient cooperative tracking strategy. To achieve this goal, a minimum number of mobile sensors is used to track the moving targets in the environment.

4. Results and Discussion

As mentioned previously in this paper, two main phases should be considered in order to accomplish the surveillance mission: task allocation and target detection and target tracking. In this section, the simulations and results of these phases are presented in detail.

4.1. Complex Task Allocation Simulations and Results. In order to evaluate the proposed approach, we consider an area surveillance application where the goal is to monitor some areas in a public place with a team of mobile sensors, each equipped with a vision system, and laser ranger sensor. To tackle this application, it is assumed that for each area, a set of surveillance points (vantage points) is selected from which the mobile sensors can view the interior of the area. The architecture under study achieves the surveillance task while keeping in mind the minimization of the total traveling distance of the whole team. Sensors frequently use the SSP algorithm when bidding, and when reordering schedules after trades or task completion.

This subsection describes a surveillance simulations and results for an indoor scenario. An example of this is the use of a team of mobile sensors to survey an indoor environment, such as malls or airports. For that, the proposed algorithms and simulations are applied on Waterloo airport in the city of Waterloo, ON, Canada. The Waterloo airport consists of six main areas, and so the goal of the proposed system is to track targets within these areas, such as people, in order to secure the airport. In order to accomplish this, the airport areas (areas of interest (AOIs)) should first be allocated to the available mobile sensors. Each mobile sensor will scan the allocated area, if any, looking for targets to track. Not all sensors will have areas to scan. In other words, the proposed task allocation approach may allocate more than one area to one mobile sensor in order to minimize the traveling cost. As the number of areas of interest in this scenario is fixed (six areas), the presented simulations in this section assumes a fixed number of areas with varying the number of mobile sensors required to achieve the surveillance mission.

The average cost is computed by calculating the cost of executing the mission task using 50 runs and then taking the average. In terms of this average cost, the results in Figures 7 and 8 show that both fixed and dynamic tree allocations (centralized and hierarchical) consistently outperform the other algorithms. It is also seen that the dynamic tree allocation outperforms fixed tree allocation, which was expected as the replanning ability is added to the sensors in the dynamic tree allocation.

On average, the hierarchical task tree algorithm is better than the centralized task tree algorithm besides its good feature of relying on different auctioneers compared to one auctioneer in the centralized algorithm. This is because the hierarchical auctioning allows more auction rounds to happen and so the mobile sensors may find themselves in better positions to win more tasks than their old positions if it is found beneficial to the whole system. Thus, the hierarchical auctioning increases the possibility of improving the system performance than the centralized auctioning.

In order to assess the scalability of proposed task allocation algorithms, an outdoor scenario is considered. An example of this scenario is a battlefield in which it is required to know where the enemy is. In order to address this, the mobile sensors team divides the battlefield into various areas. A set of surveillance points is selected from which the mobile sensors can view the interior of each area. To simplify this scenario, an example of an outdoor scenario was run with six

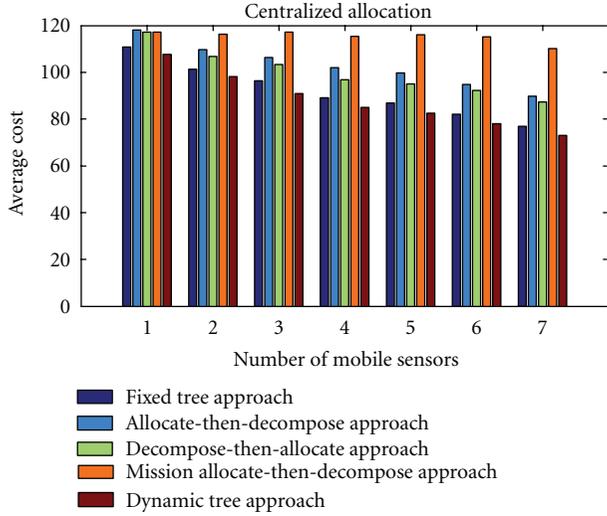


FIGURE 7: Comparison of the average cost for centralized allocation mechanism.

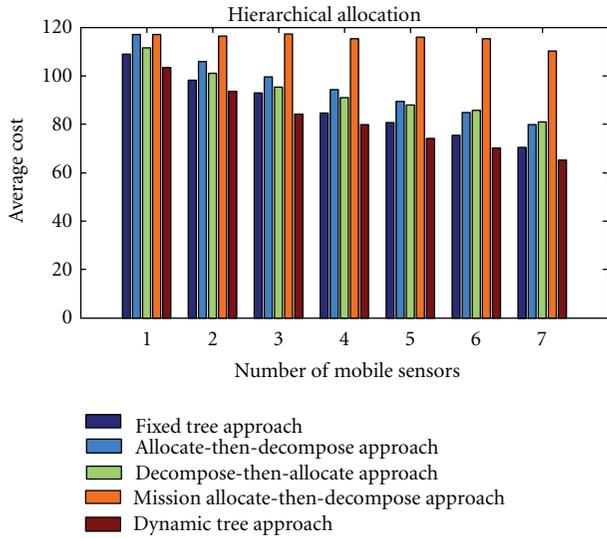


FIGURE 8: Comparison of the average cost for hierarchical allocation mechanism.

different cases as shown in Table 1 with 50 runs for each case. These runs consider different-sized areas, different locations, and different initial positions for the mobile sensor team. In this type of scenarios, the mobile sensors are permitted to move from one vantage point to another passing into the areas of interest.

The proposed fixed and dynamic tree allocation algorithms are compared with a higher level allocate-then-decompose [13], allocate-then-decompose [16], and decompose-then-allocate [13] algorithms. In terms of this average cost, the results in Figures 9 and 10 show that both fixed and dynamic tree allocations (centralized and hierarchical) consistently outperform the other algorithms, especially for the complex cases such as cases 4, 5, and 6. It can also be seen that the dynamic tree allocation outperforms

TABLE 1: Test cases.

Case	Number of areas	Number of mobile sensors
1	1	2
2	4	2
3	5	3
4	7	5
5	6	6
6	4	7

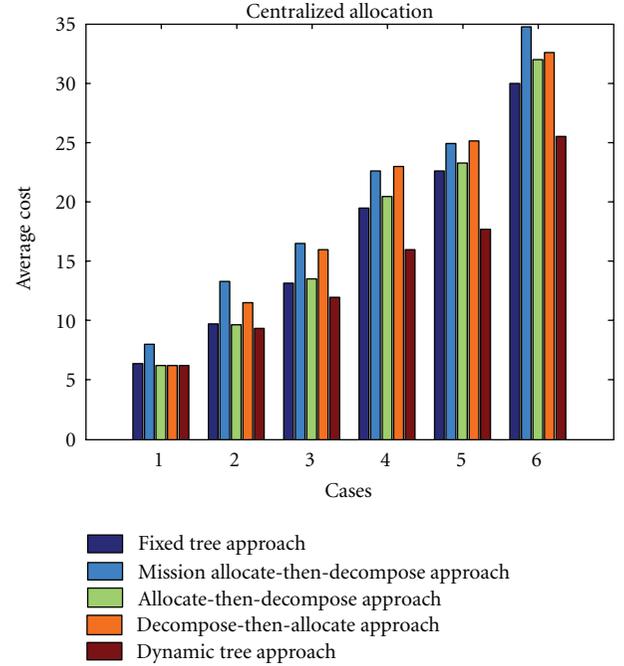


FIGURE 9: Comparison of the average cost for centralized allocation mechanism for outdoor environment.

fixed tree allocation, which was expected as the replanning ability is added to the sensors in the dynamic tree allocation.

4.2. Target Tracking Simulation and Results. As mentioned above, the target tracking algorithm for an individual target is decoupled from the cooperative tracking algorithm for a multitasking system. So, the simulation results for single target tracking system, as a basis layer of the cooperative multitarget system, are presented first. The target tracking algorithm design depends on targets, sensors, and environments. The simulation results of two types of trackers are presented—supervised and unsupervised Extended Kohonen Maps. The tracking performance is judged by both accuracy of tracking and speed of convergence. Also, the simulation results for a cooperative multitarget tracking system are presented.

4.2.1. Tracking Using Supervised Learning EKM. When tracking using supervised learning EKM, the EKM is trained on the forward mode, namely, on a transformation from the

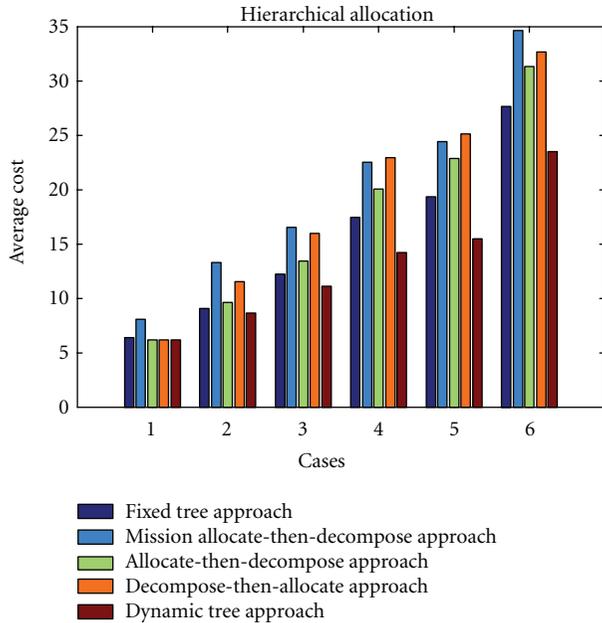


FIGURE 10: Comparison of the average cost for hierarchical allocation mechanism for outdoor environment.

space of motor commands to the space of visual perceptions. Now the trained network is used in backward mode to compute the inverse function, which transforms a visual perception into a motor command. The task is to guide the mobile sensor to a target location placed at arbitrary angle θ and distance dis in the workspace. The angle and distance data are provided by the laser range sensor attached to the tracker. To be more specific, θ is now defined as the angle between the sensor's heading direction and the vector connecting the sensor axle mid-point and the target, while dis is the Euclidean distance between the sensor axle mid-point and the target location. The observed θ and dis values are supplied to the EKM in backward mode to retrieve a velocity pair (s_r, s_l) . For this particular application, the competition on the pattern (θ, dis) has been designed to consider its components in sequence. First, θ is processed. Thus, the competition is restricted to the weight vector component of the neuron, which stores the angle information. As the result of this preliminary step, a subset of grid neurons which match θ equally well is selected. Second, dis is processed, but only on neurons selected at the previous step. The competition is restricted to the weight vector component of the neuron which stores the distance information. The overall competition process leads to the selection of a velocity pair for the mobile sensor, namely, the fan-in weight vector of the winning neuron. In the simulations introduced next, 30×30 neurons are used in the structure of the Extended Kohonen Map.

Figure 11 shows tracking a moving target using supervised learning EKM. The small circle and square are the initial positions of the target and the tracker, respectively. The indicated arrows show the direction of movement of both, the target and the tracker. As seen from the trajectories of the

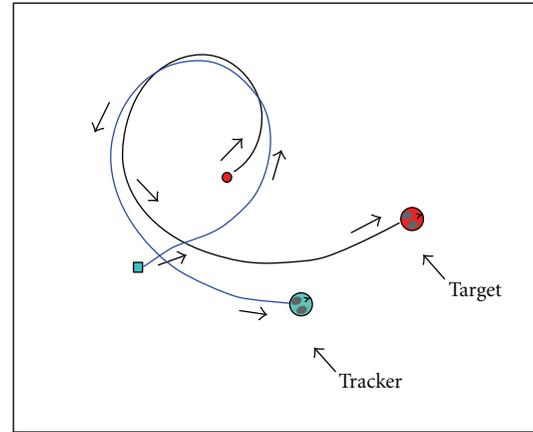


FIGURE 11: Single target tracking using supervised learning EKM.

target and the tracker, the supervised learning EKM does not provide a good tracking performance in terms of accuracy and tracking speed.

4.3. Tracking Using Unsupervised Learning EKM. In this case, the same structure for the extended Kohonen map is used but without any idea about the correct parameters of it (i.e., the fan-in and fan-out weights). Random parameters can be assumed initially, which will be changed during unsupervised learning. Selecting the initial parameters is very challenging, because it will affect the speed of convergence and the accuracy of the tracking algorithm. In this paper, the parameters of the Extended Kohonen Map are initially selected from the supervised learning stage (mentioned above). After that, a single target tracking algorithm is used to adjust these parameters, while the mobile sensor is trying to track the target (online training). At each training cycle, the weights of the winning neuron and its neighbors are modified to be closer to the sensory data (which are the angle and distance from the tracker). The input weights of the Extended Kohonen Map are updated towards the actual displacement of the tracker and the output weights are also updated. Thus, there will be a map between the actual displacement of the tracker and its motor commands. After the extended Kohonen Map self-organization has converged, the neurons will stabilize in a state such that the input weights represent the displacements traveled by the tracker and the output weights represent the corresponding motor commands that produce these displacements. For any winning neuron ne , given the sensory input $x_p = w_{ne}$, the neuron will produce motor commands y_p which yields a desired displacement $v = w_{ne}$. For sensory input $x_p \neq w_{ne}$ but close to w_{ne} , the motor commands output produced by the winning neuron will still yield the corresponding displacement. Therefore, the more neurons in the Extended Kohonen Map structure, the finer and smoother the motion of the tracker.

Figure 12 shows tracking a moving target using unsupervised learning EKM. The input-outputs weights of the unsupervised EKM are initialized using the final weights

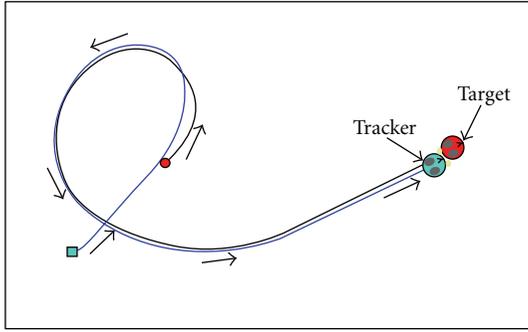


FIGURE 12: Single target tracking using unsupervised learning EKM.

(after learning) of the supervised EKM. The small circle and square are the initial positions of the target and the tracker, respectively. The indicated arrows show the direction of movement of both the target and the tracker. As seen from the trajectories of the target and the tracker, the unsupervised learning EKM has provided a good tracking performance in terms of accuracy and tracking speed compared to the supervised learning EKM performance shown in Figure 11.

4.4. Cooperative Multitarget Tracking Simulations. The proposed multitarget tracking algorithm is analyzed in various configurations. The environment was designed to be a $6\text{ m} \times 9\text{ m}$ rectangular shape. The sensors were initially deployed in the environment based on the task allocation algorithm that assigned a set of mobile sensors to a set of vantage points as described above. Sensor sensing range was set to be 1.5 m . The cooperative multi target algorithm performance is judged by two parameters: the average coverage AC (7) over time and the average energy saving over time AE (11).

The energy saving can be measured by considering the number of *busy* and *sleeping* trackers over time. In order to do that, three different cases are analyzed. In the first case, the number of mobile sensors n is equal to the number of targets m , and in the second and third cases, the number of sensors is $3/4$ and $1/2$ the number of targets, respectively. In each case, extensive simulations have been done by changing the locations of trackers and targets, the speed of targets and trackers, the direction of targets and trackers, and the number of clusters. The average coverage and average energy saving have been recorded for around 25 trials in each case. The average coverage and the average energy saving for all the cases are shown in Figures 13 and 14, respectively. Each point shows the average coverage or the average energy saving obtained in 25 simulated runs.

As shown in Figures 13 and 14, the average coverage and the average energy saving percentage decrease with time. This was expected because the clustered targets will spread in the environment over time and which will require more trackers to track them, which decrease the energy savings. Also, the average coverage will decrease especially when the number of targets is more than the number of trackers ($n = (3/4)m$ and $n = (1/2)m$).

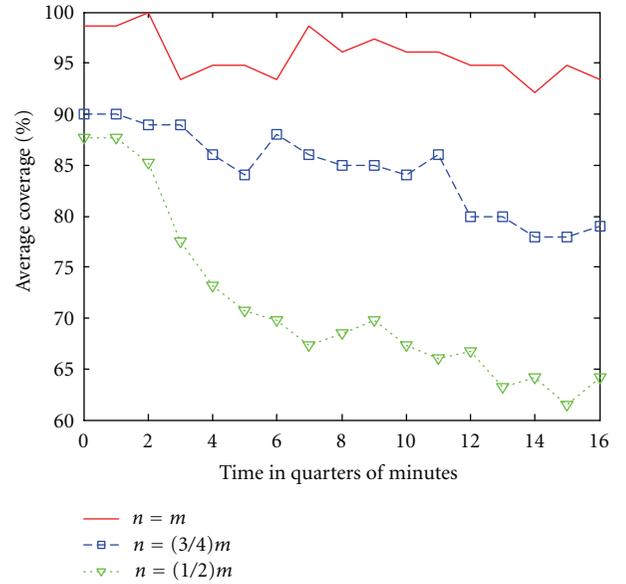


FIGURE 13: Comparison between EKM average coverage for different n and m ratios.

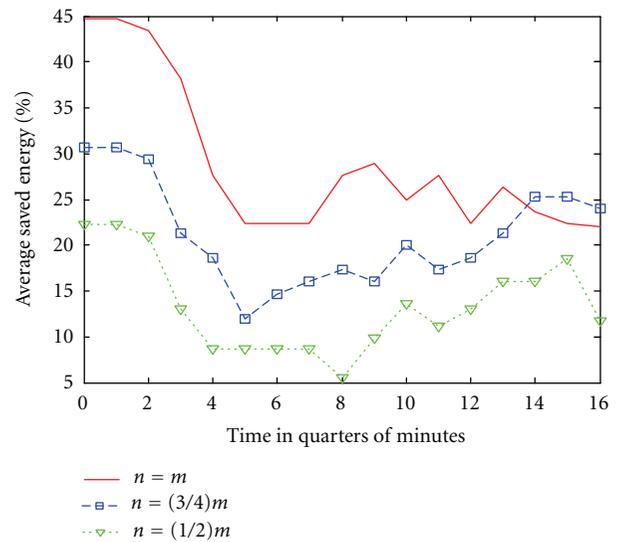


FIGURE 14: Comparison between EKM average energy saving for different n and m ratios.

The performance of the proposed approach is evaluated by comparing it with four different approaches with different observation policies: (1) local approach [34], (2) A-CMOMMT approach [34], (3) B-CMOMMT approach [39], and (4) modified B-CMOMMT [20]. The local approach controls the tracker motion by computing the summation of the attractive forces coming from nearby targets and the repulsive forces coming from nearby trackers. The A-CMOMMT approach was studied to determine the effectiveness of weighting the force vectors coming from the nearby targets. Adding the weights to the force vectors in

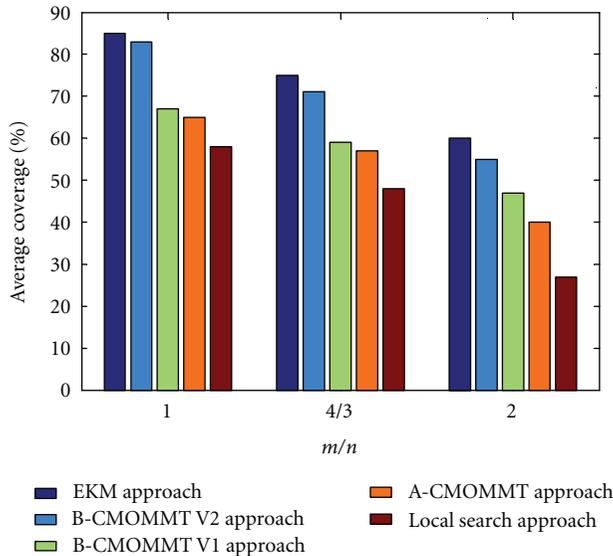


FIGURE 15: Comparison between EKM and other approaches.

A-CMOMMT causes the trackers to be less attractive to targets that are already observed by other trackers. The B-CMOMMT approach is essentially proposed to overcome some problematic situations that may arise in the A-CMOMMT approach. More detailed discussions of these situations can be found in [20, 39]. The most important difference between A-CMOMMT and B-CMOMMT is that the force vectors coming from nearby trackers are weighted as well. The B-CMOMMT approach is improved by introducing more refined techniques for target loss prediction in [20]. The comparison results are shown in Figure 15.

Figure 15 summarizes the results of this comparison with different m/n ratios (every point is the average of 25 simulated runs). These results show nearly equivalent performance of our proposed algorithm and the improved B-CMOMMT approach for small m/n ratios, but also the superior performance of our target tracking approach for larger m/n ratios. Although, we did not prove that the simulation results would hold for larger m/n ratios, it is expected that results similar to what we did will continue for larger m/n ratios.

More interesting is the detailed comparative performance of the proposed approach with the improved B-CMOMMT when m/n equal to one. As shown in Figure 16, the proposed algorithm provides a good coverage performance compared to the B-CMOMMT algorithm.

5. Conclusion

A market-based framework for mobile surveillance systems has been presented in this paper. The proposed framework capitalizes on the strengths of market economies that enable mobile sensing agents to collectively execute complex tasks efficiently and reliably. Task allocation and cooperative target-tracking have been studied in this paper using the

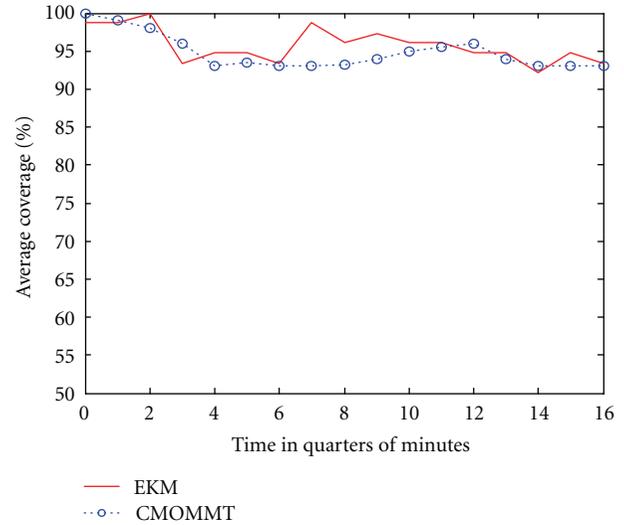


FIGURE 16: Comparison between EKM and CMOMMT average coverage for $n = m$.

proposed framework as two challenging problems of mobile surveillance systems. These challenges are addressed individually and collectively. The results of the conducted experiments showed that hierarchical dynamic tree task allocation outperforms all other techniques. Moreover, good coverage performance and energy saving have been achieved during target detection and tracking. In the future, we consider extending the proposed algorithms so that constrained and tight tasks can be handled. An example for constrained tasks is two tasks that cannot be done independently as the same sensor would obviously have to do both of them. Tight tasks cannot be decomposed into further single sensor tasks. In this case, a subgroup of mobile sensors could determine their joint costs and submit joint bids for such type of tasks.

References

- [1] F. M. D. Fave, S. Canu, L. Iocchi, D. Nardi, and V. A. Ziparo, "Multi-objective multi-robot surveillance," in *Proceedings of the 4th International Conference on Autonomous Robots and Agents (ICARA '09)*, pp. 68–73, Wellington, New Zealand, February 2009.
- [2] G. L. Foresti, C. S. Regazzoni, and P. K. Varshney, *Multisensor Surveillance Systems: The Fusion Perspective*, Kluwer Academic Publishers, 2003.
- [3] J. C. Castillo, F. A. Cano, J. S. Cuerda, and A. F. Caballero, "Multisensory architecture for intelligent surveillance systems—integration of segmentation, tracking and activity analysis," in *Proceedings of 13th ICEIS International Conference*, pp. 157–162, 2011.
- [4] A. Elmogy, F. Karray, and A. Khamis, "Auctionbased consensus mechanism for cooperative tracking in multi-sensor surveillance systems," in *Proceedings of 4th International Conference in Humanoid, Nanotechnology, Information Technology, Communication Control, Environment and Management (HNICEM '09)*, pp. 149–158, Manila, Philippines, March 2009.

- [5] H. F. Durrant-Whyte, "Sensor models and multisensor integration," *International Journal of Robotics Research*, vol. 7, no. 6, pp. 97–113, 1988.
- [6] B. P. Gerkey and M. J. Mataric, "Sold!: auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.
- [7] M. B. Dias and A. Stentz, "A market approach to multi-robot coordination," Tech. Rep. CMURI-TR-01-26, Carnegie Mellon University, Pittsburgh, Pa, USA, 2002.
- [8] A. M. Elmogy, *Market-based framework for mobile surveillance systems [Ph.D. thesis]*, University of Waterloo, Waterloo, ON, Canada, 2010.
- [9] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 3, pp. 397–408, 2005.
- [10] X. Wang, S. Wang, and D. Bi, "Distributed visual-target-surveillance system in wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 5, pp. 1134–1146, 2009.
- [11] H. Sayyaadi and M. Moarref, "A distributed algorithm for proportional task allocation in networks of mobile agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 405–410, 2011.
- [12] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 34, no. 2, pp. 138–153, 2004.
- [13] R. Zlot and A. Stentz, "Market-based multirobot coordination for complex tasks," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 73–101, 2006.
- [14] A. M. Elmogy, A. M. Khamis, and F. O. Karray, "Dynamic complex task allocation in multisensor surveillance systems," in *Proceedings of the 3rd International Conference on Signals, Circuits and Systems (SCS '09)*, Djerba, Tunisia, November 2009.
- [15] G. Isbitiren and O. B. Akan, "Three-dimensional underwater target tracking with acoustic sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 8, pp. 3897–3906, 2011.
- [16] N. Kalra, D. Ferguson, and A. Stentz, "Hoplites: a market-based framework for planned tight coordination in multirobot teams," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1170–1177, April 2005.
- [17] X. F. Xie and J. Liu, "Multiagent optimization system for solving the traveling salesman problem (TSP)," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 2, pp. 489–502, 2009.
- [18] H. K. Tsai, J. M. Yang, Y. F. Tsai, and C. Y. Kao, "An evolutionary algorithm for large traveling salesman problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 4, pp. 1718–1729, 2004.
- [19] C. Wurrll, D. Henrich, and H. Worn, "Multi-goal path planning for industrial robots," in *Proceedings of International Conference on Robotics and Application (RA '99)*, Santa Barbara, Calif, USA, 1999.
- [20] A. Kolling and S. Carpin, "Cooperative observation of multiple moving targets: an algorithm and its formalization," *International Journal of Robotics Research*, vol. 26, no. 9, pp. 935–953, 2007.
- [21] S. Martinez-Jaramillo and E. P. K. Tsang, "An heterogeneous, endogenous and coevolutionary GP-based financial market," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 33–55, 2009.
- [22] R. M. Zlot, *An auction-based approach to complex task allocation for multirobot teams [Ph.D. thesis]*, Pittsburgh, Pa, USA, 2006.
- [23] M. B. Dias, *TraderBots: a new paradigm for robust and efficient multi-robot coordination in dynamic environments [Ph.D. thesis]*, Robotics Institute, Carnegie Mellon University, 2004.
- [24] X. Zheng, S. Koenig, and C. Tovey, "Improving sequential single-item auctions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 2238–2244, Beijing, China, October 2006.
- [25] M. Berhault, H. Huang, P. Keskinocak et al., "Robot exploration with combinatorial auctions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1957–1962, October 2003.
- [26] B. P. Gerkey and M. J. Mataric, "Sold!: auction methods for multi-robot control," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.
- [27] S. C. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '99)*, pp. 1234–1239, May 1999.
- [28] F. Tang and S. Saha, "An anytime winner determination algorithm for time-extended multi-robot task allocation," in *Proceedings of the International Conference on Automation, Robotics, and Control Systems*, pp. 123–130, 2008.
- [29] M. H. Rothkopf, A. Pekeć, and R. M. Harstad, "Computationally manageable combinatorial auctions," *Management Science*, vol. 44, no. 8, pp. 1131–1147, 1998.
- [30] J. Sparbert, K. Dietmayer, and D. Streller, "Objecttracking in traffic scenes with multi-hypothesis approach using laser range images," in *Proceedings of IEEE 4th International Conference on Intelligent Transportation System*, 2001.
- [31] H. Zhao, J. Cui, H. Zha, K. Katabira, X. Shao, and R. Shibasaki, "Monitoring an intersection using a network of laser scanners," in *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC '08)*, pp. 428–433, Beijing, China, December 2008.
- [32] H. Zhao and R. Shibasaki, "A novel system for tracking pedestrians using multiple single-row laser-range scanners," *IEEE Transactions on Systems, Man, and Cybernetics Part A*, vol. 35, no. 2, pp. 283–291, 2005.
- [33] B. Kluge, C. Köhler, and E. Prassler, "Fast and robust tracking of multiple moving objects with a laser range finder," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 1683–1688, May 2001.
- [34] L. E. Parker, "Distributed algorithms for multi-robot observation of multiple moving targets," *Autonomous Robots*, vol. 12, no. 3, pp. 231–255, 2002.
- [35] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.
- [36] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [37] H. Ritter, T. Martinetz, and K. Schulden, *Neural Computation and Self-Organizing Maps: An Introduction*, Addison-Wesley Longman, Boston, Mass, USA, 1st edition, 1992.

- [38] T. Kohonen, *Self-Organization and Associative Memory*, Information Sciences, Heidelberg, Germany, 3rd edition, 1989.
- [39] A. Kolling and S. Carpin, “Multirobot cooperation for surveillance of multiple moving targets—a new behavioral approach,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 1311–1316, May 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

