Mobile Scheduling for Spatiotemporal Detection in Wireless Sensor Networks

Guoliang Xing, *Member*, *IEEE*, Jianping Wang, *Member*, *IEEE*, Zhaohui Yuan, Rui Tan, *Student Member*, *IEEE*, Limin Sun, Qingfeng Huang, *Member*, *IEEE*, Xiaohua Jia, *Senior Member*, *IEEE*, and Hing Cheung So, *Senior Member*, *IEEE*

Abstract—Wireless sensor networks (WSNs) deployed for mission-critical applications face the fundamental challenge of meeting stringent spatiotemporal performance requirements using nodes with limited sensing capacity. Although advance network planning and dense node deployment may initially achieve the required performance, they often fail to adapt to the unpredictability and variability of physical reality. This paper explores efficient use of mobile sensors to address limitations of static WSNs for target detection. We propose a data-fusion-based detection model that enables static and mobile sensors to effectively collaborate in target detection. An *optimal* sensor movement scheduling algorithm is developed to minimize the total moving distance of sensors while achieving a set of spatiotemporal performance requirements including high detection probability, low system false alarm rate, and bounded detection delay. The effectiveness of our approach is validated by extensive simulations based on real data traces collected by 23 sensor nodes.

Index Terms—Data fusion, algorithm/protocol design and analysis, wireless sensor networks.

1 Introduction

Deploying wireless sensor networks (WSNs) for mission critical applications (such as intruder detection and tracking) often face the fundamental challenge of meeting stringent spatial and temporal performance requirements imposed by users. For instance, a surveillance application may require any intruder to be detected with a high probability (e.g., $>90\%$), a low false alarm rate (e.g., $<1\%$), and within a bounded delay (e.g., 20 s). Due to the limited capability and unreliable nature of low-power sensor nodes, overprovisioning (of coverage, detection, and communication capabilities) seems to be the only choice for a static sensor network to meet such stringent performance requirements. However, overprovisioning only works up to the point where the reality meets the original expectation about the characteristics of physical phenomena and environments. If a new on-demand task arises after deployment and its requirements exceed the statically planned network capability, the task could not be accomplished. For instance, in a battlefield monitoring scenario, sensor failures in a small region may lead to a perimeter breach and the sensor nodes deployed in other regions become useless.

To better cope with the unpredictability and variability of physical reality and improve the agility of sensor networks, mobile sensors can be introduced to dynamically reconfigure the sensor network capability in an *on-demand* manner. In a static-mobile hybrid sensor network, the mobile sensors can move close to targets and increase the signal-to-noise ratio (SNR) and the fidelity of detection results beyond what is achievable by static sensor nodes alone in many situations. Furthermore, efficient collaboration between mobile and static nodes could effectively change sensing densities on demand, potentially reducing the number of sensors needed comparing to all-static network deployments.

However, several challenges must be addressed in order to take advantage of the mobility of WSNs in target detection. First, due to the higher design complexity and manufacturing cost, the number of mobile nodes available in a network is often limited. Therefore, mobile sensors must effectively collaborate with static sensors to achieve the maximum utility. Second, mobile sensors are only capable of low-speed and short-distance movement, in practice, due to the high power consumption of locomotion. For instance, the typical speed of several mobile sensor platforms (e.g., Packbot [33], Robomote [14], and XYZ [28]) is only 0.5-2 m/s. An XYZ mobile sensor powered by two AA batteries can only move about 165 meters [28] before the depletion of batteries. Therefore, the movement of mobile sensors must be efficiently scheduled in order to maximize the amount of target information gathered within a short moving distance.

In this paper, we propose a data-fusion-centric target detection model that features effective collaboration between static and mobile sensors. We derive an optimal sensor movement scheduling algorithm that minimizes the total moving distance of sensors under a set of spatiotemporal performance requirements including: 1) bounded detection delay; 2) high target detection probability; and 3) low system false alarm rate. Furthermore, we conduct

- *G. Xing and R. Tan are with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824-1226. E-mail: glxing@msu.edu, tanrui@gmail.com.*
- *J. Wang, Z. Yuan, and X. Jia are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: {jianwang, csjia}@cityu.edu.hk, yzhvincent@gmail.com.*
- *L. Sun is with the Institute of Software, Chinese Academy of Sciences, Beijing 100080, P.R. China. E-mail: sunlimin@is.iscas.ac.cn.*
- *Q. Huang is with C8 MediSensors, Inc., 727 University Ave., Los Gotos, CA 95032. E-mail: qingfeng@ieee.org.*
- *H.C. So is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. E-mail: hcso@ee.cityu.edu.hk.*

Manuscript received 20 Sept. 2008; revised 18 Aug. 2009; accepted 5 Oct. 2009; published online 16 Feb. 2010.
Recommended for acceptance by S. Olariu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-09-0360.
Digital Object Identifier no. 10.1109/TPDS.2010.41.

extensive simulations based on real data traces collected by 23 sensors in the SensIT vehicle detection and classification experiments [17]. Our results show that a small number of mobile sensors can significantly boost the detection performance of a network. Moreover, our algorithm can achieve satisfactory performance in a range of realistic scenarios with single/multiple moving vehicles and high variations in the speed of mobile sensors.

The rest of the paper is organized as follows: Section 2 reviews related work. Sections 3 and 4 introduce the background and the formulation of our problem, respectively. The optimal sensor movement scheduling is studied in Section 5. Extensions of our solution are discussed in Section 6. We present simulation results in Section 7 and conclude the paper in Section 8.

2 Related Work

Recent work demonstrated that the sensing performance of WSNs can be improved by integrating mobility. Several projects proposed to eliminate coverage holes in a sensing field by relocating mobile sensors [5], [39], [37]. Although such an approach improves the sensing coverage of a network deployment, it does not dynamically improve the network's performance after targets of interest appear. Complementary to these projects, we focus on online sensor collaboration and movement scheduling strategies that are used after the appearance of targets.

In our recent work [34], we proposed a decision-fusion-based detection model in which each mobile sensor makes its own detection decision and locally controls its movement. In this paper, we adopt a value-fusion-based detection model that significantly simplifies the task of mobile sensors. Specifically, each mobile sensor in a detection process is only required to move a certain distance and send its measurements to its cluster head. Such a model is more suitable for mobile sensors with limited capability of signal processing and motion control. In contrast, a mobile sensor in the algorithm proposed in [34] must be able to locally detect targets and adaptively control their movement. Moreover, this paper studies several important issues that are not addressed in [34] including optimal movement scheduling and multitarget detection.

Several recent studies [8], [26] analyzed the impact of mobility on detection delay and area coverage. These studies are based on random mobility model and do not address the issue of actively controlling the movement of sensors. Bisnik et al. [2] analyzed the performance of detecting stochastic events using mobile sensors. Chin et al. [9] proposed to improve the coverage of a region by patrolling static routes using mobile sensors. Different from [9], we study efficient sensor collaboration and movement scheduling strategies that achieve specified target detection performance. Mobile sensors that can move reactively are used in a networked robotic sensor architecture [1], [29] to improve the sampling density over a region. However, they did not focus on target detection under spatiotemporal performance constraints.

Collaborative target detection in stationary sensor networks has been extensively studied [4], [12], [11], [16], [25], [36]. Several recent projects also studied network deployment strategies that can achieve specified detection performance [11], [15], [41]. Our recent work [40] investigates the fundamental impacts of data fusion on the coverage of WSNs. Practical network protocols that facilitate target detection and tracking using static or mobile sensors have also been investigated [6], [20], [27], [42]. Complementary to these studies that deal with the mobility of targets, we focus on improving target detection performance by utilizing the mobility of sensors.

Several recent studies [31], [3], [7] formulate target detection and tracking in mobile WSNs as game problems and propose several motion strategies for mobile sensors. In these works, the mobile sensors move *actively* to improve the surveillance quality. However, the power consumption of locomotion is not explicitly considered. In contrast, the mobile sensors in our approach move *reactively* only when a coarse detection consensus is reached and the power consumption of locomotion is minimized.

As a fundamental issue in robotics, motion planning has been extensively studied [22]. We refer to [21], [10] for comprehensive surveys on this topic. Recent works [30], [23] consider the motion planning/control of autonomous robots for searching/tracking targets. Sensor movement scheduling in mobile WSNs for target detection poses several new challenges that have not been addressed in the existing robotic motion planning literature, which include limited mobility of sensors, resource constraints, and stringent quality-of-service requirements such as low false alarm rate, high detection probability, and bounded detection delay.

3 Preliminaries

In this section, we describe a single-sensor sensing model and a multisensor fusion model that are used in our solutions.

3.1 Target and Sensing Model

Sensors detect targets by measuring the energy of signals emitted by targets. Suppose a target is at location u and emits a signal of power $\mathcal{W}$. The signal power decays as a function of the distance from the target. The signal power measured by a sensor that is d meters away from the target is given by

$$W(d) = \mathcal{W} \cdot w(d),$$

where $w(d)$ is a nonincreasing function satisfying $w(0) = 1$ and $w(\infty) = 0$. The $w(d)$ is referred to as the *signal decay function*.

Measurements at a sensor are corrupted by noise modeled as the Gaussian distribution with zero mean. Let $N_i^2(T)$ denote the noise energy measured by sensor i during time interval T. Let x_i be the distance between sensor i and the target. When the target remains stationary during T, the total energy sensor i measures during T is:

$$U_i(T) = W(x_i) \cdot T + N_i^2(T). \quad (1)$$

We note that the above sensing and noise models have been widely assumed in the literature of signal detection [17], [36], [12], [34], [40], [41] and also have been empirically verified [24], [19]. We note that the algorithm proposed in this paper does not depend on the specific form of the signal decay function $w(d)$. In practice, the parameters of sensing and noise models are often estimated using a training data set before deployment or during initialization phase of the network. In particular, for acoustic sensors, the

signal decay function can be expressed as follows [17], [12], [34], [41]:

$$w(d) = \begin{cases} \dfrac{1}{(d/d_0)^k}, & if\ d > d_0, \\ 1, & if\ d \le d_0, \end{cases} \tag{2}$$

where k is the decay factor and d_0 is a constant determined by the size of the target and the sensor. We note that the decay factor k is typically between 2 to 5. In this paper, we adopt the signal decay function in (2) for the numerical examples and simulations that are based on the acoustic data traces from the SensIT experiments [17].

3.2 Multisensor Fusion Model

We assume that the network is organized into clusters. Sensors send their energy measurements to the cluster head, which, in turn, compares the average of all measurements to a threshold η. If the average is greater than η, the cluster head decides that a target is present. Otherwise, it decides there is no target. η is referred to as the *detection threshold* hereafter. Such a value-based data fusion model has been studied in the literature [12].

The performance of detection is characterized by the probability of false alarm (or false alarm rate) and probability of detection, denoted by P_F and P_D, respectively. The P_F is the probability that a target is regarded to be present when the target is actually absent. The P_D is the probability that a target is correctly detected. Suppose there exist n sensors and each sensor measures signal energy for duration T, P_F can be expressed as

$$\begin{aligned} P_F &= P\left(\frac{1}{n}\sum_{i=1}^{n} N_i^2(T) > \eta\right) \\ &= 1 - P\left(\sum_{i=1}^{n} N_i^2(T) \le n\eta\right). \end{aligned} \tag{3}$$

We assume that the noise signal strength is a random variable that follows zero-mean normal distribution. Hence, $\sum_{i=1}^{n} N_i^2(T)$ follows the Chi-square distribution with n degrees of freedom whose cumulative distribution function is denoted as $\mathcal{X}_n(\cdot)$. So, (3) becomes:

$$P_F = 1 - \mathcal{X}_n(n\eta). \tag{4}$$

The probability of detecting a target is

$$\begin{aligned} P_D &= P\left(\frac{1}{n}\sum_{i=1}^{n}(W(x_i)\cdot T + N_i^2(T)) > \eta\right) \\ &= P\left(\sum_{i=1}^{n} N_i^2(T) > n\eta - \sum_{i=1}^{n} W(x_i)\cdot T\right) \\ &= 1 - \mathcal{X}_n\left(n\eta - \sum_{i=1}^{n} W(x_i)\cdot T\right). \end{aligned} \tag{5}$$

4 Mobility-Assisted Spatiotemporal Detection Problem

This section formulates our problem called the Mobility-assisted Spatiotemporal Detection (MSD). We first provide a brief overview of our basic approach. We describe the assumptions made in this paper in Section 4.3 and formally formulate the MSD problem in Section 4.4.

4.1 Approach Overview

The MSD problem is characterized by a 4-tuple (A, α, β, D). Specifically, for a given set of static and mobile sensors and any target that appears at one of the locations in set A, our objective is to minimize the total expected moving distance of the mobile sensors subject to the following constraints: 1) P_D is no lower than β; 2) P_F is no higher than α; and 3) the expected detection delay is no greater than D seconds. The objective of minimizing the total expected moving distance of mobile sensors is motivated by the following practical considerations. First, as discussed in Section 1, reducing the moving distance of mobile sensors prolongs network lifetime as the power consumption of locomotion is high. Second, short moving distances mitigate the side effects of mobility such as disruptions to network topology and compromise on stealthiness of a network which is not desirable for many applications deployed in hostile environments like battlefields.

We assume that targets appear at a set of known physical locations referred to as *surveillance location*s with certain probabilities. Surveillance locations are often identified by the network autonomously after the deployment. We note that the monitored physical phenomenon in many applications has complex or unknown spatial distribution. The network is organized into clusters around surveillance locations by running a clustering protocol. Clustering has considerable impact on the detection performance of the network. We note that clustering in WSNs has been extensively studied. In particular, if the surveillance locations are static due to constant target distribution, we can employ static clustering algorithms [41]. The clustering algorithm in [41] maximizes the system detection performances at surveillance locations. If the surveillance locations are dynamic due to changeable target distribution, we can employ dynamic clustering algorithms such as the one proposed in [6]. In this paper, the sensor closest to the surveillance location is elected as the cluster head. Each static sensor belongs to only one cluster. However, a mobile sensor may belong to multiple clusters because it can contribute to the target detection at different surveillance locations. In the rest of this paper, we focus on a surveillance location and the corresponding cluster.

We propose a data-fusion-centric detection model as follows: Initially, all sensors in a cluster periodically send the measurements to the cluster head that compares the average energy against a threshold λ_1. Once a positive detection decision is made, the cluster head initiates the second phase of detection by sending mobile sensors on a movement schedule S that specifies which sensors should move, the time instances to start moving, and the distances to move. Mobile sensors then move toward the surveillance location according to the schedule. After a certain delay, each sensor sends the cluster head the sum of its energy measurements. Finally, the final detection decision is made by the cluster head by comparing against another threshold λ_2. The detection thresholds λ_1, λ_2 and the movement schedule S are determined under the constraints that the aggregate delays, P_D and P_F of the two phases must satisfy the requirements specified by D, β, and α, respectively.

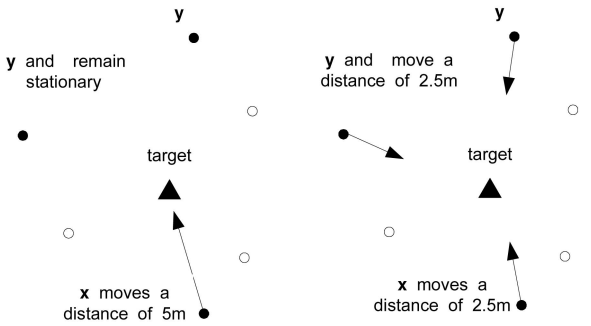

Fig. 1. Two examples of sensor movement scheduling. Mobile and static sensors are represented by solid and void circles, respectively. When the maximum movement delay is 5 seconds, sensor x moves 5 m toward the target. When the maximum movement delay is 5 seconds, sensors x, y, and z move 2.5 m toward the target. The distances between the target and the three static sensors are 8, 9, and 10 m, respectively. The distances between the target and x, y, and z are 11, 12, and 13 m, respectively.

A key advantage of the above two-phase detection model is the reduced total distance of moving as the mobile sensors move in a reactive manner. Moreover, this model facilitates the collaboration between static and mobile sensors. As the decision of the first phase is made based on the measurements of all sensors in a cluster, the static sensors help filter out false alarms that would trigger unnecessary movement of mobile sensors. In addition, the accuracy of the final detection decision is improved in the second phase because the signal to noise ratios are increased as the mobile sensors move closer to the surveillance location.

4.2 A Numerical Example

We now illustrate our approach using a numerical example depicted in Fig. 1. To simplify the discussion, suppose there is only one surveillance location, which is monitored by three static and three mobile sensors. The required P_D and P_F are 90 and 5 percent, respectively. The average speed of a mobile sensor is 0.5 m/s. During initialization, the cluster head estimates the parameters of target energy model (see (2)) using a training data set. We use the following parameters: $\mathcal{W} = 0.51$ (after normalization), $d_0 = 2.6$ m, and $k = 2$, which are estimated using the data set collected in a vehicle detection experiment [17] (the details are given in Section 7).

Initially, each sensor periodically measures acoustic energy and reports to the cluster head every 0.75 seconds. According to (4) and (5), the maximum achievable P_D can be computed to be 81.5 percent under a P_F of 5 percent. Suppose the maximum time that a mobile sensor can spend on moving is 10 seconds, which is determined by the allowable detection delay and other processing delay. To improve P_D to 95 percent, the cluster head computes a movement schedule in which sensor x moves 5 m toward the target. As a result, the SNR of sensor x is increased from -3.14 to 4.5 dB. When each sensor can only move for 5 seconds due to a shorter detection delay requirement, three sensors x, y, and z are scheduled to move 2.5 m toward the target. The average SNR of the three sensors is increased from -3.69 to -0.82 dB.

This example shows that the detection delay can be reduced by scheduling more sensors to move simultaneously. In our solution, the detection thresholds of the two detection phases and the movement schedule are jointly determined to satisfy the detection performance requirements specified by α, β, and D. In addition, we prove that our solution can minimize the total moving distance of sensors (see Section 5).

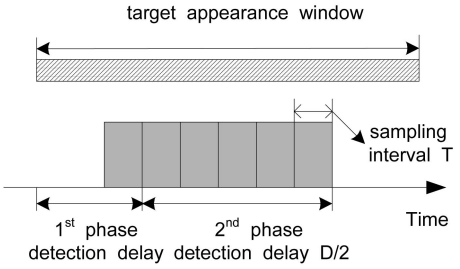

Fig. 2. In the first-phase detection, all sensors sample at a period of D. Each sampling lasts for T time. Expected detection delay is $D/2$. In the second-phase detection, sensors continuously sample at an interval of T for $D/2$ time.

4.3 Assumptions

We make the following assumptions before we formulate the problem formally. First, the clocks of all sensors are synchronized. Second, we assume that each mobile node knows its own location (through a GPS unit mounted on it or a localization service in the network) and can orient its movement in a given direction.

In the first phase of detection, all sensors operate in a synchronous sleep schedule in which they wake up to sample energy at a period of S seconds. We assume that the probability that a target may appear at any time instance is uniform. Therefore, the expected detection delay due to sleep scheduling is $S/2$. Suppose $S = 2\gamma D$, where D is the required detection delay bound. Thus, the expected delay of the first-phase detection is $S/2 = \gamma D$, where $\gamma \in (0, 1)$ is a constant chosen according to the desirable trade-off between detection delay and power consumption. For the convenience of discussion, we assume $\gamma = 1/2$ in the rest of discussion. After waking up, each sensor samples energy for T seconds and sends to the cluster head. For instance, the acoustic data are recorded at a frequency of 4,960 Hz in every 0.75 s in the SensIT experiments [17]. That is, T is 0.75 s.

In the second phase of detection, all sensors in the cluster sample energy at a period of T. After a delay of $D/2$, sensors report the sum of their energy measurements to the cluster head. This is necessary to bound the total expected detection delay within D as the expected delay of the first-phase detection is $D/2$. The mobile sensors belong to multiple clusters and may return to their original locations after the second phase of detection as they may be requested to detect targets at other locations. Fig. 2 illustrates the temporal view of the two-phase detection model employed by our approach.

We assume that the average speed of a mobile sensor is v unless it is stationary. To simplify the motion control of mobile sensors, we assume that the moving distance of a mobile sensor in the second phase is always multiple of vT. Furthermore, to simplify our problem formulation, we assume that the distance between a sensor and a surveillance location is also multiple of vT. Both v and T are small in practice. For instance, T is 0.75 s in the experiments in [17] and v is 0.5-2 m/s for typical mobile sensor systems [14],

[28], [33]. Under such settings, vT is about 0.4-1.5 meters. As the sensor locations in real deployment are typically tens to hundreds of meters [17], the assumption that the sensor locations are multiple of vT does not introduce significant errors. We also evaluate the impact of step length, i.e., vT, on system performance in Section 7.3.

4.4 Problem Formulation

In this section, we present the formulation of the MSD problem. We assume that targets appear at low frequencies and the probability that two targets appear in the same detection window is negligible. Thus, our following discussion focuses on one surveillance location u. In Section 6, we relax this assumption and extend our approach to the case of detecting multiple targets. We define the following notations:

1. The P_u denotes the probability that a target appears at location $u \in A$ during time D, which is known or can be estimated by the history of detection.
2. The x_i represents the distance between sensor i and location u. We assume that u is the origin, and hence, x_i also represents sensor i's location.[1]
3. A sensor *move*, denoted by $\mathcal{M}_i(x,t)$, is the process in which sensor i moves from location x to $x - vT$ in time interval $[t, t+T]$, where T is the sampling interval (see Section 4.3).
4. A *movement schedule*, denoted by $\mathcal{S} = \{\mathcal{M}_i(x,t)\}$, is a list of moves. The $\|\mathcal{S}\|$ represents the cardinality of $\mathcal{S}$, i.e., the total number of moves in the schedule. Therefore, $\|\mathcal{S}\|$ quantifies the total movement distance of mobile sensors.
5. The cluster that monitors location u contains a set of sensors indexed as $1, 2, \ldots, n$. The sensors are initially located at $(x_1^0, \ldots, x_n^0)$.
6. N_s and N_m represent the sets of indices of static and mobile sensors, respectively.

Our objective is to find a 3-tuple $<\eta_1, \eta_2, \mathcal{S}>$ in which η_1 and η_2 are two detection thresholds and $\mathcal{S}$ is a sensor movement schedule, such that the total expected distance that the mobile sensors move away from their original positions is minimized:

$$\left(P_u \cdot P_{D_1} + (1 - P_u) \cdot P_{F_1}\right) \cdot \|\mathcal{S}\|, \tag{6}$$

subject to

$$P_{F_1} \cdot P_{F_2} \leq \alpha, \tag{7}$$

$$P_{D_1} \cdot P_{D_2} \geq \beta, \tag{8}$$

$$\forall \mathcal{M}_i(x_i,t) \in \mathcal{S},$$

$$(i \in N_m) \wedge \left(vT \leq x_i \leq x_0^i\right) \wedge \left(0 \leq t \leq \frac{D}{2} - T\right), \tag{9}$$

$$\eta_1 \in \left\{\eta_{1(0)}, \eta_{1(1)} \cdots \eta_{1(k)}\right\}, \tag{10}$$

$$\eta_2 \in \left\{\eta_{2(0)}, \eta_{2(1)} \cdots \eta_{2(k)}\right\}. \tag{11}$$

P_{F_1}, P_{D_1}, P_{F_2}, and P_{D_2} are given by

$$P_{F_1} = 1 - \mathcal{X}_n(n\eta_1), \tag{12}$$

$$P_{F_2} = 1 - \mathcal{X}_{nm}(nm\eta_2), \tag{13}$$

$$P_{D_1} = 1 - \mathcal{X}_n\left(n\eta_1 - \sum_{i=1}^{n} W(x_i^0) \cdot T\right), \tag{14}$$

$$P_{D_2} = 1 - \mathcal{X}_{nm}\left(mn\eta_2 - \sum_{i=1}^{n}\sum_{j=0}^{m-1} \mathcal{E}_i(j,\mathcal{S})\right), \tag{15}$$

$$m = \frac{D}{2T}, \tag{16}$$

where $\mathcal{E}_i(j,\mathcal{S})$ is the energy sampled by sensor i during interval $[jT, (j+1)T]$ under the movement schedule $\mathcal{S}$:

$$\mathcal{E}_i(j,\mathcal{S}) = \begin{cases} \int_{jT}^{(j+1)T} W(x - vt)dt, & if\ \mathcal{M}_i(x, jT) \in \mathcal{S}, \\ W(x) \cdot T, & if\ \mathcal{M}_i(x, jT) \notin \mathcal{S}. \end{cases} \tag{17}$$

The objective function (6) quantifies the total expected distance that sensors move away from their original locations. The movement of sensors is the result of a positive decision in the first-phase detection, which has a probability of $P_u \cdot P_{D_1}$ being correct and a probability of $(1 - P_u) \cdot P_{F_1}$ being a false alarm.

Inequalities (7) and (8) require that the joint P_F and P_D of the two phases must meet the constraints specified by the application. Equation (9) specifies the spatial and temporal constraints of sensor movement. Each mobile sensor must move between its initial location and the target location, and the movement must complete within $D/2$. At the end of $D/2$, all sensors send their energy measurements to the cluster head which then makes the final detection decision.

Equations (10) and (11) specify that the values of two detection thresholds are discrete. In practice, the precision of a sensor is determined by the bandwidth of its ADC converter. Detection probabilities of the two phases P_{D_1} and P_{D_2} are given by (14) and (15), respectively. According to definition (17), $\mathcal{E}_i(j,\mathcal{S})$ is equal to the integral of power over T if sensor i moves from x to $x - vT$ in $\mathcal{S}$. Otherwise, it is equal to the product of T and power measured at x, which is the position of sensor i after the last move that occurs before time instance jT or its initial position x_i^0 if it has not moved.

5 Optimal Solution of MSD Problem

In this section, we first discuss the structure of the optimal solution of the MSD problem. A dynamic-programming-based optimal movement scheduling algorithm is then presented in Section 5.2. In Section 5.3, we discuss how to determine the detection thresholds of the two-phase detector.

5.1 Structure of Optimal Solution

The formulation in Section 4.4 shows that the MSD problem is a nonlinear optimization problem with as many as $nD/2T + 2$ variables (η_1, η_2, and the movement schedule $\mathcal{S}$ composed of at most $nD/2T$ moves). An exhaustive search of all possible values of these variables incurs exponential complexity. In this section, we first analyze the structure of the MSD problem, which allows us to develop an optimal solution that has a polynomial-time complexity.

An MSD solution $<\eta_1, \eta_2, \mathcal{S}>$ is *valid* if all constraints can be satisfied. A valid solution is *optimal* if it minimizes the

1. As the detection performance of a sensor only depends on its distance to the target, the sign of x_i is insignificant.

cost function among all valid solutions. We note that when the movement schedule $\mathcal{S}$ is known, unique values of η_1 and η_2 can be found. According to $\mathcal{S}$, the total sampled energy can be computed by (17), and hence, constraints (7) to (11) can be evaluated. An exhaustive search in the domains of η_1 and η_2 can find the values that minimize the cost function (6) under the constraints. The $\mathcal{S}$ is said to be valid/optimal, if the solution constructed by $\mathcal{S}$ and η_1 and η_2 (that are found by the exhaustive search) is valid/optimal. We now focus on finding the optimal movement schedule. The search of η_1 and η_2 for a given movement schedule is discussed in Section 5.3.

We define the following notation. For a movement schedule X, $E(X) = \sum_{i=1}^{n} \sum_{j=0}^{m-1} \mathcal{E}_i(j, X)$, where $\mathcal{E}_i(j, X)$ is defined in (17), representing the total energy sampled in the second-phase detection. For a solution $<\eta_1, \eta_2, \mathcal{S}>$, $c(\eta_1, \mathcal{S})$ represents the value of the cost function (6). We have the following theorem:

Theorem 1. *Suppose $\mathcal{S}$ and $\mathcal{S}'$ are two valid movement schedules. If $\|\mathcal{S}\| = \|\mathcal{S}'\|$ and $E(\mathcal{S}) \geq E(\mathcal{S}')$, there must exist η_1 and η_1' such that $c(\eta_1, \mathcal{S}) \leq c(\eta_1', \mathcal{S}')$.*

Proof. Suppose $<\eta_1, \eta_2, \mathcal{S}>$ and $<\eta_1', \eta_2', \mathcal{S}'>$ minimize the cost function among all valid solutions with schedules $\mathcal{S}$ and $\mathcal{S}'$, respectively. As $\mathcal{S}$ and $\mathcal{S}'$ are known, such solutions can be found by exhaustive searches of values of $\{\eta_{1(0)}, \eta_{1(1)} \cdots \eta_{1(k)}\}$ and $\{\eta_{2(0)}, \eta_{2(1)} \cdots \eta_{2(k)}\}$ in polynomial time. We construct a new solution $<\eta_1', \eta_2', \mathcal{S}>$. We now show that it is a valid solution. Compared to $<\eta_1', \eta_2', \mathcal{S}'>$, this new solution only changes the value of P_{D_2} in all constraints. As detection probability function P_D always increases with total measured energy and $E(\mathcal{S}) \geq E(\mathcal{S}')$, we have $P_{D_2}(\eta_2', \mathcal{S}) \geq P_{D_2}(\eta_2', \mathcal{S}')$. Therefore, constraint (8) can be met and $<\eta_1', \eta_2', \mathcal{S}>$ is a valid solution. Since $\|\mathcal{S}\| = \|\mathcal{S}'\|$ and $<\eta_1, \eta_2, \mathcal{S}>$ minimizes the cost function among all valid solutions with $\mathcal{S}$, we have

$$\begin{aligned} c(\eta_1, \mathcal{S}) &= (P_u \cdot P_{D_1}(\eta_1) + (1 - P_u) \cdot P_{F_1}(\eta_1))\|\mathcal{S}\| \\ &\leq (P_u \cdot P_{D_1}(\eta_1') + (1 - P_u) \cdot P_{F_1}(\eta_1'))\|\mathcal{S}\| \\ &= (P_u \cdot P_{D_1}(\eta_1') + (1 - P_u) \cdot P_{F_1}(\eta_1'))\|\mathcal{S}'\| \\ &= c(\eta_1', \mathcal{S}'). \end{aligned}$$

□

Theorem 1 shows that the expected number of moves decreases with the total amount of energy sampled by sensors. Therefore, the optimal movement schedule must maximize the amount of energy gathered by mobile sensors for a given number of moves.

5.2 Optimal Sensor Movement Scheduling

In this section, we present an optimal movement scheduling algorithm that enables sensors to gather the maximum amount of energy for a given number of moves.

According to (2), target energy decays with distance d in the order of $1/d^k$ $(2 \leq k \leq 5)$. Therefore, for the same moving distance, a sensor senses more energy when it gets closer to the target. This observation leads to the intuition that moving the sensors that are closer to the target first may maximize the total measured energy. A simple greedy heuristic motivated by this observation works as follows to schedule N moves: Move sensors according to their distance to the target, starting with the sensor closest to the target. Stop moving a sensor if it reaches the location of the target or the number of moves that have scheduled is N. However, this heuristic is not always optimal because it does not consider the temporal duration of energy sensing, which can be illustrated by the following example.

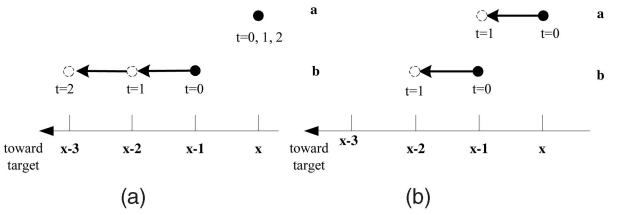

Fig. 3. Two different movement schedules of two sensors. The number of total moves in both schedules is two. (a) Sensor b moves two steps, while sensor a remains fixed. (b) Both sensors a and b move one step.

In Fig. 3, two moves are scheduled for sensors a and b that are initially x and $x-1$ away from the target, respectively. As b is closer to the target, it moves two steps, while a remains stationary. Fig. 3b illustrates another schedule in which both a and b move one step from time 0. To simplify our discussion, we set $d_0 = 1, k = 2$ in the energy attenuation model defined by (2), and $v = T = 1$. Suppose E_1 and E_2 represent total amount of energy gathered in a duration of l seconds under the two schedules, respectively. E_1 and E_2 are derived in (18) and (19) according to (17). In the first schedule, b senses for two seconds when it moves from $x-1$ to $x-3$, and $l-2$ seconds at $x-3$, while a does not move and senses for l seconds at x. E_2 can be computed similarly. When $l = 10$ and $x = 20$, $E_1 = 0.0589 > 0.0583 = E_2$. However, when $l = 3$ and $x = 20$, $E_1 = 0.0172 < 0.0173 = E_2$. We can see that moving a sensor two steps is superior to moving two sensors one step only when the duration of sensing is long (i.e., $l = 10$):

$$\begin{aligned} E_1 &= \int_0^2 \frac{dt}{(x-1-t)^2} + \frac{l-2}{(x-3)^2} + \frac{l}{x^2} \\ &= \frac{1}{x-3} - \frac{1}{x-1} + \frac{l-2}{(x-3)^2} + \frac{l}{x^2}, \end{aligned} \tag{18}$$

$$\begin{aligned} E_2 &= \int_0^1 \frac{dt}{(x-t)^2} + \frac{l-1}{(x-2)^2} + \int_0^1 \frac{dt}{(x-1-t)^2} + \frac{l-1}{(x-1)^2} \\ &= \frac{1}{x-2} - \frac{1}{x} + \frac{l-1}{(x-2)^2} + \frac{l-1}{(x-1)^2}. \end{aligned} \tag{19}$$

This example shows that sensors' locations and the duration of sensing should be jointly considered in order to maximize the total amount of sensed energy. We now consider the optimal movement schedule of H moves when there is only one sensor i. Obviously, the measured energy always decreases with i's distance to the target and increases with the sensing duration. Therefore, the optimal schedule for i is to move H steps consecutively from time zero, which allows it to sense at the closest location possible at any time instance. Interestingly, this conclusion still holds when there are more than one sensors. This is because sensors can move in parallel, and hence, optimizing the movement of each sensor individually maximizes the total

amount of energy sensed by all sensors. We have the following theorem:

Theorem 2. *Suppose an optimal schedule has total L moves. There are n sensors such that sensor i moves l_i steps with $L = \sum_{1\le i\le n} l_i$. For each sensor i, l_i moves occur consecutively from time zero.*

Proof. Suppose sensor i moves l_i steps in an arbitrary schedule $\mathcal{S}$. The l_i moves of sensor i can be partitioned into K groups and each group consists of continuous moves. Denote t_j and l_i^j $((1 \le j \le K))$ as the time sensor i waits before starting the jth group of moves and the number of moves in the jth group, respectively. For example, sensor i waits for t_1 before moving l_i^1 steps continuously without a stop. Denote e_i as the total amount of energy gathered by sensor i, which can be expressed as follows:

$$\begin{aligned} e_i = &\int_0^{l_iT} W\big(x_i^0 - vt\big)dt + t_1 W(x_i^0) \\ &+ \sum_{2\le j\le K} t_j W\left(x_i^0 - vT \sum_{1\le m\le j-1} l_i^m\right) \\ &+ \left(\frac{D}{2} - \sum_{2\le j\le K} t_j - l_iT\right)\cdot W\big(x_i^0 - vl_iT\big). \end{aligned}$$

The integral in the above equation is equal to the energy gathered during the j moves. The last term corresponds to the energy gathered after sensor i stops at the final location and each other term corresponds to the energy gathered between two groups of moves. Obviously, $\sum_{1\le j\le K} l_i^j = l_i$. Then, we have

$$\begin{aligned} e_i = &\int_0^{l_iT} W\big(x_i^0 - vt\big)dt + \left(\frac{D}{2} - l_iT\right)\cdot W\big(x_i^0 - vl_iT\big) \\ &+ t_1\big(W\big(x_i^0\big) - W\big(x_i^0 - vl_iT\big)\big) \\ &+ \sum_{2\le j\le K} t_j\left(W\left(x_i^0 - vT\sum_{1\le m\le j-1} l_i^m\right) - W\big(x_i^0 - vl_iT\big)\right). \end{aligned}$$

As $l_i \ge \sum_{1\le m\le j-1} l_i^m$ and $W(\cdot)$ is a decreasing function, the above equation is maximized when $t_i = 0$ $(1 \le i \le K)$. That is, all the moves of sensor i are continuous and start at time zero. □

According to Theorem 2, the number of possible move combinations in the optimal schedule is significantly reduced. We now present a dynamic programming algorithm that finds the optimal schedule for a given number of sensor moves.

Let h_i be the number of consecutive moves of sensor i in the optimal schedule. The location of sensor i after the moves is $x_i^0 - vh_iT$, where x_i^0 is the initial location of i. The total amount of energy sensed by sensor i during the second-phase detection, denoted by $e_i(h_i)$, can be calculated as follows:

$$e_i(h_i) = \int_0^{h_iT} W\big(x_i^0 - vt\big)dt + \left(\frac{D}{2} - h_iT\right)\cdot W\big(x_i^0 - vh_iT\big). \tag{20}$$

We number mobile sensors by $1,\ldots,n$. Let $E(j,h)$ be the maximum total amount of energy sensed by sensors $1,\ldots,j$ with a total number of h moves. Then, we have a dynamic programming recursion:

$$E(j,h) = \max_{0\le h_j\le H_j}\{E(j-1,h-h_j) + e_j(h_j)\}, \tag{21}$$

$$H_j = \min\left(\frac{D}{2T}, \frac{x_j^0}{vT}\right), \tag{22}$$

where H_j is the maximum number of moves of sensor j as it will stop moving if it reaches the location of the target or the second-phase detection finishes at time $D/2$. The initial condition of the above recursion is $E(0,h) = 0$.

According to (21), at the jth iteration of the recursion, the optimal value of $E(j,h)$ is computed as the maximum value of H_j cases which have been computed in previous iterations of the recursion. Specifically, for the case where sensor j moves h_j steps, the total sensed energy can be computed as $E(j-1,h-h_j) + e_j(h_j)$, where $E(j-1,h-h_j)$ is the maximum total amount of energy sensed by sensors $1,\ldots,j-1$ with a total number of $h-h_j$ moves. According to Theorem 2, sensor j's moves are consecutive from time zero if it moves in the optimal schedule. Therefore, at most H_j cases need to be considered when computing $E(j,h)$. The maximum amount of energy sensed by all sensors in h moves is given by $E(n,h)$.

We now describe how to construct the optimal schedule using the dynamic programming recursion. For each $E(j,h)$, we define a schedule $\mathcal{S}(j,h)$ initialized to be empty. The $\mathcal{S}(j,h)$ is filled incrementally in each iteration when computing $E(j,h)$. Specifically, in the jth iteration of the recursion, if $E(j-1,h-h_x) + e_j(h_x)$ gives the maximum value among all cases, we add h_x moves of j to $S(j,h)$. Formally,

$$\begin{aligned} \mathcal{S}(j,h) &= \mathcal{S}(j-1,h-h_x) \cup \{\mathcal{M}_i(x,vxT) \mid 0 \le x \le h_x - 1\}, \\ h_x &= arg\max_{0\le h_j\le H_j} E(j-1,h-h_j) + e_j(h_j). \end{aligned}$$

The complexity of the dynamic programming procedure is $\mathcal{O}((nD/T)^2)$.

5.3 Procedure of Solving MSD Problem

We now present the procedure of solving the MSD problem. For each possible number of moves l, we first compute $E(n,l)$ and the movement schedule $\mathcal{S}(n,l)$ using the scheduling algorithm described earlier. Then, the values of η_1 and η_2 are searched to minimize the expected sensor moving distance under the constraints. The maximum number of moves is given by $H = \sum_{1\le i\le n} H_i$, where H_i is given by (22). The optimal movement schedule and η_1 and η_2 can then be found in H iterations. Fig. 4 shows the pseudocode of the procedure.

For each value of η_1, P_{D_1} and P_{F_1} can be computed according to (14) and (12). Furthermore, unique P_{F_2} can be determined as the minimum value that satisfies constraint (7). This is because that a higher P_{F_2} leads to a higher P_{F_2}, which may cause constraint (7) to be violated. Then, η_2 can be solved from P_{F_2} according to (13). So far, constraints (7), (9), (10), and (11) have been satisfied. For instance, constraint (7) is enforced in solving η_2. It remains to check if constraint (8) is met. A new cost is computed according to (6) if (8) is met. A zero cost may occur when all constraints are satisfied without moving the sensors toward the target.

Input: D, $\{E(n,j) \mid 0 \le j \le H\}$, P_u, $(x_1^0, x_2^0, \cdots, x_n^0)$, $[\eta_{1(0)}, \eta_{1(k)}]$, $[\eta_{2(0)}, \eta_{2(k)}]$
/*output movement schedule and two detection thresholds*/
Output: $\mathcal{S}$, η_1, η_2

```
1)  cost = ∞;
2)  for l = [0 : H]
3)      for n1 = [η1(0) : η1(k)]
4)          Compute P_D1 and P_F1 using (14) and (12);
5)          Find the minimum n2 ∈ {η2(0)..η2(k)} using (7);
6)          Compute P_D2 using E(n,l) according to (15);
7)          if ((8) holds)
8)              Compute current cost C using (6);
9)              if (C = 0) exit; fi;
10)             if (C < cost)
11)               cost = C; S = S(n,l); η1 = n1; η2 = n2
12)             fi
13)         fi
14)     end
15) end
```

Fig. 4. The procedure of solving the MSD problem.

If the new cost is lower than the current cost, the current movement schedule and detection thresholds are recorded. As $E(n,l)$ and $\mathcal{S}(n,l)$ can be precomputed using the scheduling algorithm, the complexity of the procedure is $\mathcal{O}(H \cdot k)$.

6 Extensions

In this section, we extend our two-phase detection model to an M-phase model and describe a coordination mechanism that allows cluster heads to handle multiple targets.

6.1 Multiphase MSD Problem

In the two-phase model, a false alarm in the first phase inevitably results in the movement of sensors. We address this issue by a multiphase detector in which sensors carry out multiple rounds of detection before making a decision of moving.

Suppose the number of detection phases increases from 2 to M. Initially, all sensors perform measurements at a period of G ($G \ge T$) seconds. In each period, sensors sample signal energy for T seconds and send their measurements to the cluster head which makes a detection decision by comparing the average of measurements against threshold η_1. If a positive decision is made, sensors then start measuring at a period of T, and a detection decision is made by the cluster head in each period. If all continuous $M-1$ positive decisions are made by the cluster head, the last phase is initiated and mobile sensors start to move toward the target according to the movement schedule S. If a negative decision was made, the M-phase detection is restarted and all sensors perform measurements at a period of G seconds.

Figs. 5 and 6 illustrate two possible cases of the M-phase detection. In Fig. 5, all $M-1$ decisions made are positive, and hence, sensors start their movement in the last phase. In contrast, Fig. 6 shows that a negative decision is made in the jth phase and the M-phase detection is restarted. We can

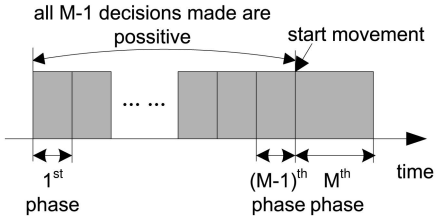

Fig. 5. All the decisions made in the previous M-1 phases are positive. Mobile sensors start their movement in the Mth phase.

see that a system false alarm occurs only when all the first $M-1$ phases produce a false alarm. Therefore, the unnecessary sensor movements measurement due to false alarms can be effectively reduced.

To simplify our design, we assume that the detection thresholds of sensors are identical in the first $M-1$ phases.

The M-phase MSD problem is to minimize:

$$\left(P_u \prod_{j=1}^{M-1} P_{D_j} + (1-P_u) \prod_{k=1}^{M-1} P_{F_k}\right) \cdot \|\mathcal{S}\|, \tag{23}$$

subject to

$$\prod_{i=1}^{M} P_{F_i} \le \alpha, \tag{24}$$

$$\prod_{i=1}^{M} P_{D_i} \ge \beta. \tag{25}$$

Constraint (9) is also applicable and not shown. $PD_i(PF_i)$ $(1 \le i \le M-1)$and $PD_M(PF_M)$ have the same expressions as $PD_1(PF_1)$ and $PD_2(PF_2)$ in (14) and (15), respectively. The cost function (23) is the total expected moving distance in the Mth phase. The M-phase MSD problem can be solved similarly as the two-phase MSD problem. The major difference is the calculation of the PD and PF. The time complexity of solving the M-phase MSD problem is also $\mathcal{O}(H \cdot k)$.

Although more detection phases improve the performance of detection and reduce the expected moving distance of sensors, they inevitably lead to higher detection delay. We now derive the maximum number of detection phases under the given detection delay bound D. As sensors perform measurements every G seconds in the first phase, the expected delay between the start of the first phase and the appearance of a target is $G/2$ seconds. After the target is detected, the sampling delay of first $M-1$

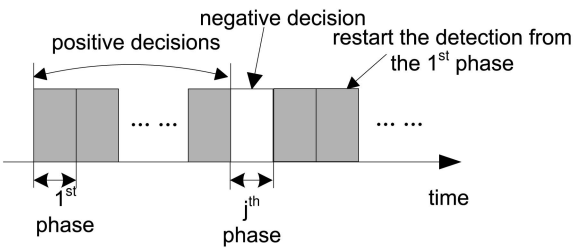

Fig. 6. A negative decision is made in the jth phase, and the detection is restarted from the first phase.

phases is $(M-1)T$, and the delay of the last phase is γD, where $\gamma \in (0,1)$ is a constant specified by the user. Therefore, the expected system detection delay is $G/2 + (M-1)T + \gamma D$. As the total delay must be no longer than D, the number of phases M must satisfy: $G/2 + (M-1)T + \gamma D \leq D$, then we have:

$$M \leq \frac{2(1-\gamma)D - G}{2T} + 1. \tag{26}$$

As the duration of first phase must be no shorter than the sampling interval T, i.e., $G \geq T$, the maximum number of phases is $\lfloor \frac{(2(1-\gamma)D-T)}{2T} + 1 \rfloor$. In the simulations conducted in this paper, we set γ to be 0.5.

6.2 Detection of Multiple Targets

When multiple concurrent targets are close to each other, efficient coordination among different clusters is needed as the movement of sensors toward a target may affect the detection performance of other targets. In this section, we discuss two scenarios where coordinated scheduling is necessary.

In the first scenario, multiple targets may appear simultaneously where we need to determine which set of sensors should move toward each target such that detection criteria are satisfied and the total expected distance that mobile sensors move away from their original positions is minimized. Suppose that two targets may appear in locations u and v simultaneously. Thus, for a sensor i, we need to determine whether it should move to u or v and the number of movements. Following (20), we use $e_i^u(h_i)$ and $e_h^v(h_i)$ to denote the energy sensed by sensor i if it has h_i movements toward u and v, respectively. Note that $e_i^u(h_i) = 0$ (or $e_i^v(h_i) = 0$) if sensor i is out of the sensing range of target u (or v). Consider sensors $1, 2, \ldots, j$ with a total number of h movements. Suppose that our aim is to achieve that the energy sensed for target u is at least g^u and the energy sensed for target v is at least g^v. Let $f(j, h, g^u, g^v) = 1$ denote the existence of such a movement schedule, and $f(j, h, g^u, g^v) = 0$ denote the nonexistence. Then, we have a dynamic programming recursion as

$$f(j,h,g^u,g^v) = \max_{h_j} \big\{ f\big(j-1, h-h_j, g^u - e_j^u(h_j), g^v\big), \\ f(j-1, h-h_j, g^u, g^v - e_j^v(h_j)) \big\},$$

where the initial condition is given by $f(1,h,g^u,g^v) = 1$ for $g^u \geq e_1^u(h)$ and $g^v = 0$, and $g^v \geq e_1^v(h)$ and $g^u = 0$, and $f(1,h,g^u,g^v) = 0$ for other values of h, g^u, and g^v. Note that the implementation of the dynamic program needs to discretize the level of energy where the maximum value of g^u (or g^v) can be calculated by (22) assuming that all sensors move to target u (or v). With the information of $f(n,h,g^u,g^v)$, we can derive the optimal scheduling by a procedure similar to the procedure described in Fig. 4 where for each possible l, if $f(n,l,g^u,g^v) = 1$, we check whether g^u and g^v satisfy the constraints of P_D and P_F of both targets u and v. If so, the algorithm stops. Otherwise, the algorithm continues with a new l.

We now discuss the second scenario where only one target may appear at a time. However, before finishing detection of the previous target, a new target may appear. In such a scenario, offline scheduled movement may not be valid as some sensors may not be available for detecting the following target before they can move back to their original positions. Due to its online fashion in this scenario, optimal scheduling is not practical as we cannot predict the overlap between the detections of the previous target and the following target. We propose an on-demand coordination strategy to handle such a scenario. Suppose that A_u and A_v are two cluster heads which monitor locations u and v and share sensor i. We assume that a sensor can correctly sense the target signal with the highest SNR when multiple targets appear in its detection range. When A_u detects a possible target and requests i to move toward u, sensor i sends A_v its distance and direction to u. A_v then adjusts its detection thresholds and sensor movement schedules as follows: First, A_v labels sensor i's initial position by the time instance it is available, which can be calculated according to sensor i's round-trip time. For instance, it takes sensor i, 30 seconds to move to the location of target u, and then, returns to its original position. Therefore, i can be scheduled to move toward target v only after 30 seconds. In the movement scheduling algorithm, the total energy measurement of a sensor (20) is then calculated according to its available time intervals. The coordination among different clusters maximizes the utility of mobile sensors by taking into account their available times when detecting multiple targets.

7 Simulations

In this section, we conduct extensive simulations based on the real data traces collected in the SensIT vehicle detection experiments [17]. In the experiments, 75 WINS NG 2.0 nodes are deployed to detect Assault Amphibian Vehicles (AAVs) driving through several intersected roads. The data set used in our simulations includes the ground truth data and the time series recorded by 23 nodes at the frequency of 4,960 Hz. The ground truth data include the positions of sensors and the trajectories of the AAVs. Received energy is calculated every 0.75 s. We refer to [17] for more detailed setup of the experiments.

7.1 Simulation Methodology and Settings

The simulation code is written in C++. We use the data traces for an AAV run as the training data for estimating the energy attenuation model defined by (2). Our estimated parameters are: $\mathcal{W} = 0.51$ (after normalization), $d_0 = 2.6\ \mathrm{m}$, and $k = 2$. The estimated energy model is used by cluster heads to run the algorithm shown in Fig. 4 that computes the detection thresholds and the movement schedule of sensors. In the simulations, a sensor is associated with a real sensor in the vehicle detection experiment [17]. We note that multiple sensors can be associated with the same real sensor. In each run of simulations, when a sensor makes a measurement, the energy is set to be the real measurement gathered by a sensor at a similar distance to target in the data trace. Such a strategy realistically mimics the real performance of sensors in our simulations.

We adopt two sensor deployments in the simulations. In the first deployment, sensors are randomly distributed in a sensing field of $50 \times 50\ \mathrm{m}^2$ surrounded by four roads, as illustrated in Fig. 10. The centers of the four road sections are chosen to be the surveillance locations. Vehicles drive along the roads at a constant speed of $2.5\ \mathrm{m/s}$, which is

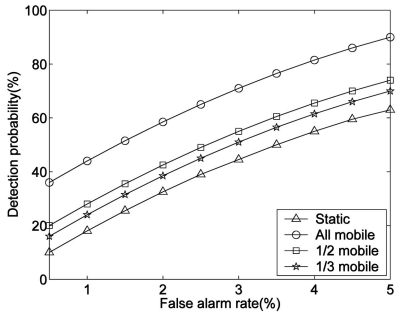

Fig. 7. ROC curves.

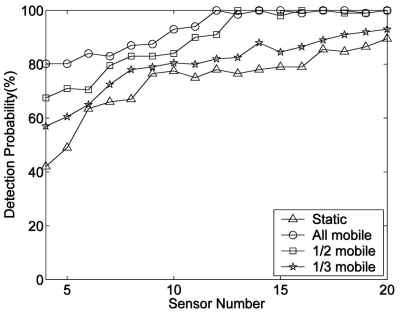

Fig. 8. PD versus number of mobile sensors. $\mathrm{PF} = 5$ percent.

similar to the average speed of the AAVs in the data traces. The simulation time of each run is 10^5 seconds. The settings of such deployment allow us to evaluate the performance of our approach in a large geographic region where multiple targets may appear. The simulation results for such deployment are presented in Sections 7.2-7.6. In the second deployment, sensors are deployed according to the sensor positions in the SensIT experiments [17] and vehicles follow the trajectories of AAVs. Fig. 23 plots the layout of sensors and the trajectory of an AAV run. We choose a location as the surveillance location, as shown in Fig. 23. The purpose of this deployment is mainly to evaluate the impact of target mobility in realistic settings. For both sensor deployments, the probability that a vehicle appears at any time instance is set to be 5 percent unless otherwise specified. Once a vehicle appears on a road, the minimum interval before the next vehicle appears on the same road is set to be 30 seconds. The detection delay requirement D is set to be $16\ \mathrm{s}$ unless otherwise specified. The requested false alarm rate (α) and detection probability (β) are set to be 0.01 and 0.9 unless otherwise specified. In each run of simulations, the speed of mobile sensors is randomly chosen within $0.5\text{-}1\ \mathrm{m/s}$. The simulation results for such deployment are presented in Section 7.7.

We note that our simulation settings account for several realistic factors. First, as we use a specific run (AAV3) in the data traces to estimate the sensor measurement model, there exists considerable deviation between the measurements of sensors in our simulations and the training data. This deviation is due to various reasons including the difference between vehicles and the changing noise level caused by wind. Therefore, the simulations evaluate the robustness of our approach with respect to these impacts. Moreover, our movement scheduling algorithm assumes that targets remain stationary at each surveillance site before disappearance. However, each AAV in our simulations drives along a road. As a result, the actual SNRs received by sensors are considerably lower than those used in the movement scheduling algorithm. The performance of our solution can be improved if the mobility of targets is explicitly taken into consideration, e.g., by integrating with target tracking algorithms [6].

7.2 Detection Performance

We now evaluate the detection performance of our two-phase detection approach with different number of mobile sensors and target localization errors.

In the first set of simulations, targets only appear at one surveillance location and remain stationary. The performance of detecting moving targets is evaluated in the rest of this section. To evaluate the impact of mobility on system detection performance, we plot four receiver operating characteristic (ROC) curves under different numbers of mobile nodes in Fig. 7. ROC curves characterize a detection system's achievable trade-off between PD and PF. In Fig. 7, *Static* refers to the deployment in which all sensors remain stationary. Total six sensors are deployed. We can see that the system detection performance increases rapidly when the false alarm rate is from 0 to 5 percent. Many detection systems have to sacrifice much detection performance in eliminating rare false alarms. Therefore, the ROC curve often increases rapidly when the false alarm rate is low [36]. We can see that the system detection performance increases significantly with the number of mobile sensors. In particular, when all six sensors are mobile, the improvement of detection probability is about 20-40 percent.

Fig. 8 shows the detection probability when the number of sensors varies from 4 to 20. In each setting, the detection threshold is computed to maximize the system PD under a PF of 0.05. We can see that PD reaches about 81 percent when only four sensors are mobile. In contrast, PD is only about 42 percent if all sensors are static. When the total number of sensors increases, the system performance under different settings becomes similar because a near 100 percent PD can be achieved without moving sensors. Fig. 8 also shows that the use of mobile sensors can significantly reduce the density of sensors needed in a deployment. For example, eight mobile sensors achieve a similar detection performance as 20 static sensors.

In previous sections, we assume that the target appears at the surveillance location. However, in practice, due to the spatial distribution of target as discussed in Section 4.1, there may exist a distance between the target position and the surveillance location, which is referred to as *target localization error*. In this section, we evaluate the impact of target localization error on the system detection performance. In the simulations, the target appears at a fixed position away from the surveillance location and remains stationary. Five static sensors and five mobile sensors are deployed. Fig. 9 plots the PD versus the target localization error under various requested PF. We can see from the figure that the system detection performance decreases with the target localization error. However, the PD decreases by only 0.1 when the localization error is up to $5\ \mathrm{m}$. Several

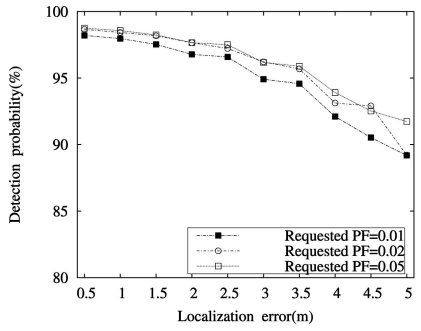

Fig. 9. PD versus localization error.

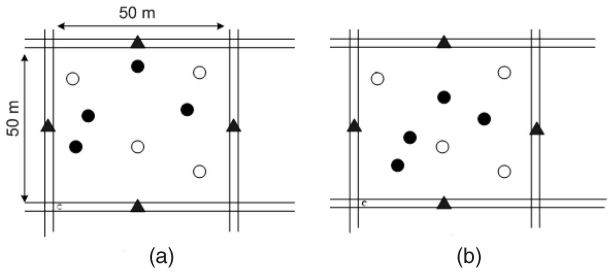

Fig. 10. (a) The sensor distribution in the initial deployment. (b) The sensor distribution in the end of a detection process. Vehicles drive through the four road sections. Mobile sensors moved toward the center of the bottom road section after a target is detected. Void and solid circles represent static and mobile sensors, respectively. Triangles represent the surveillance locations.

recent studies showed that the target localization error can be reduced to be as small as $1\ \mathrm{m}$ [35].

7.3 Moving Distance

We now evaluate the performance of our movement scheduling algorithm. To distinguish from baseline algorithms, our two-phase detection algorithm is referred to as the *mobility-assisted detector (MD)*. We compare MD again two baseline algorithms. *Greedy* is the implementation of the greedy heuristic algorithm mentioned in Section 5.2. *MD-random* is a variant of MD that employs a random movement scheduling algorithm. Specifically, at each scheduling step, a sensor is randomly chosen to move until the required detection performance is achieved. Fig. 11 shows the average number of moves of 10 mobile sensors when the requested PD varies from 0.8 to 0.95. The PF is set to be 0.01. MD significantly outperforms the two baseline algorithms, which demonstrates the effectiveness of our optimal movement scheduling algorithm.

As discussed in Section 4.3, the movement of sensors is scheduled in the unit of *step*, i.e., vT. Thus, the value of vT affects the accuracy of movement scheduling and the total moving distance of sensors. In this section, we evaluate the impact of move step length, i.e., vT, on the system performance. In the simulations, six mobile and six static sensors are deployed. The requested PF and PD are 1 and 90 percent, respectively. The speeds of mobile sensors are set to be $1\ \mathrm{m/s}$. Fig. 12 plots the total moving distance versus the move step length. Note that we change the move step length by varying the sampling interval T. From the figure, we can see that the total moving distance increases slowly with the move step length in the MD algorithm. This result demonstrates that the movement

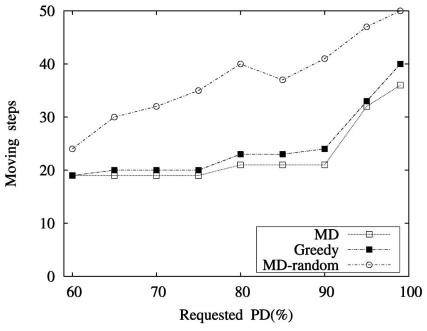

Fig. 11. Number of moves versus requested PD. $\mathrm{PF} = 1$ percent.

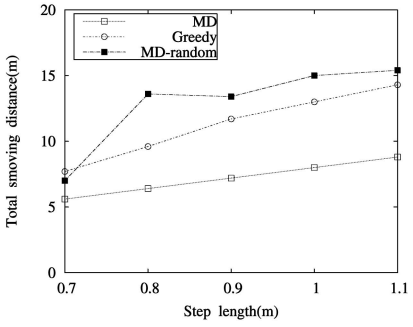

Fig. 12. Moving distance versus step length.

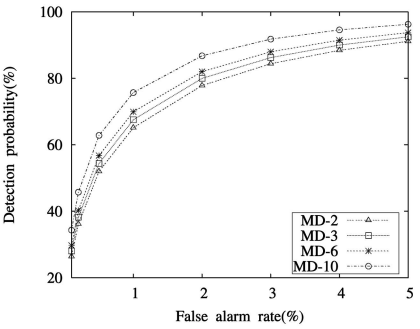

Fig. 13. ROC curves of multiphase detectors.

scheduling algorithm of MD is robust to the granularity of move step. Moreover, the move step length should be set to be the minimum length which can be achieved by the motion control of mobile sensors.

7.4 Multiphase and Multitarget Detection

We now evaluate the performance of our multiphase detection. We simulate multiphase detection algorithm (as described in Section 6) with five static and five mobile sensors. The objective of the multiphase detection approach is to improve the system detection performance by reducing the false alarm rate in the first phase, such that the expected total moving distance is reduced. In this section, we evaluate the system detection performance characterized by the ROC curves under various numbers of phases. The ROC curves of four different multiphase detectors are plotted in Fig. 13. In the figure, MD-x represents the x-phase detection algorithm. From the figure, we can see that the performance of detection increases with the number of phases of the detector.

Fig. 14 plots the maximum PD that can be achieved by different multiphase detectors when the number of mobile

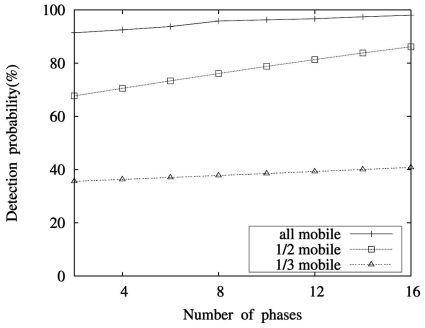

Fig. 14. PD versus number of phases.

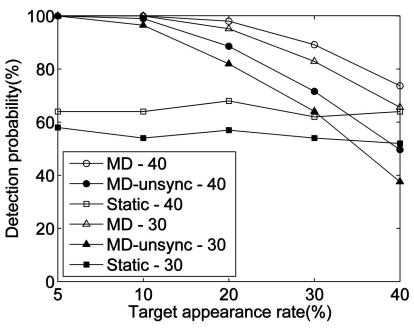

Fig. 16. PD versus target appearance rate.

sensors varies from 2 to 6. PF of all simulations is set to 0.01. We can see that PD increases linearly with the number of detection phases. However, Fig. 14 also shows that the number of mobile sensors plays a more important role in the performance of detection.

To evaluate the performance of the movement scheduling of multiphase detectors, we plot in Fig. 15 the number of sensor moves for detectors with different number of phases when the requested PD varies from 80 to 99 percent. PF is set to be 0.01. Fig. 15 shows that the performance with more detection phases is superior. For example, when the requested PD is 90 percent, sensors move 17 steps when six phases are used, and 23 steps when only three phases are used. Moreover, we can also see that the number of steps is not significantly reduced until the requested PD is above 90 percent. This is because, when the number of moves needed increases significantly, an additional detection phases can effectively filter out more false alarms and reduce unnecessary movements.

We now evaluate the performance of detecting multiple targets at different surveillance locations, as discussed in Section 6.2. The probability that a vehicle may appear at any time instance varies from 5 to 40 percent. Once a vehicle appears on a road, the minimum interval before the next vehicle appears on the same road is set to be 30 seconds. That is, we enforce that only one target may appear within a detection window on any road. However, multiple vehicles may drive on different roads at the same time. We plot PD versus the probability a target appears in Fig. 16. When multiple targets appear in the same time window D, the four cluster heads in the sensing field employ the coordination mechanism discussed in Section 6.2. When a mobile sensor is requested to move by a cluster head, it notifies all other cluster heads, which update their detection thresholds and movement schedules. For performance comparison, we implement a baseline algorithm referred to as MD-unsync, which does not update detection parameters when mobile sensors move away. Fig. 16 shows PD versus different target appearance rates. Each algorithm is labeled by the name and the number of sensors used. We note that all sensors are mobile except for the all-static case. We can see that MD yields considerably higher detection probabilities than MD-unsync. The performance of MD and MD-unsync degrades when more targets appear concurrently. In such a case, moving sensors toward a particular target often lowers the performance of detecting other targets. In contrast, the performance of static deployments remains unaffected. Nevertheless, MD always yields the best performance in all settings.

7.5 Impact of Mobile Sensor Speed

We now evaluate the impact of movement speed on the system detection performance. Fig. 17 shows the probabilities of detecting multiple targets. We can see that the detection performance increases considerably when the speed of movement becomes higher because mobile sensors can move closer to targets within the given delay bound. This result also shows that our movement scheduling algorithm can effectively take advantage of the increase of movement speed. In reality, the speed of a mobile sensor may suffer from variations because of complex terrains or temporal mechanical problems. We now evaluate system PD when the movement speed has different variations. At the beginning of simulations, the average speed of each mobile sensor, v, is randomly chosen between 0.5-$1\ \mathrm{m/s}$. The actual speed of the

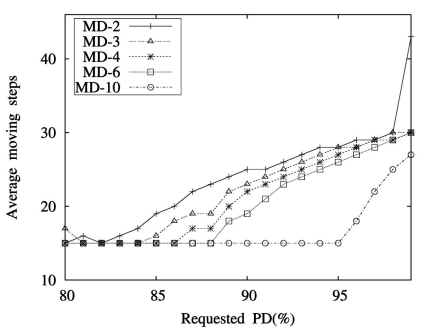

Fig. 15. Average moving steps versus requested PD.

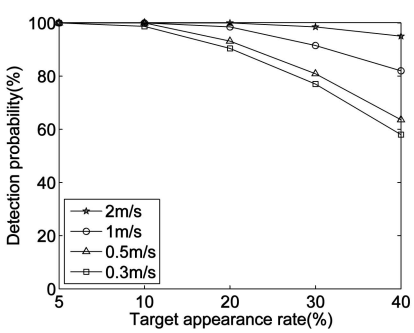

Fig. 17. PD versus speed of movement.

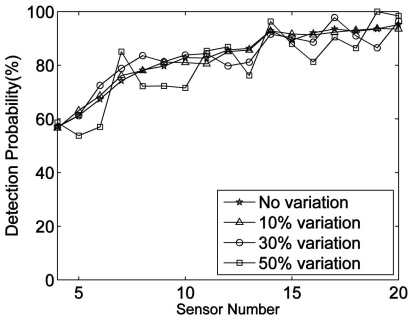

Fig. 18. PD versus variation of movement speed.

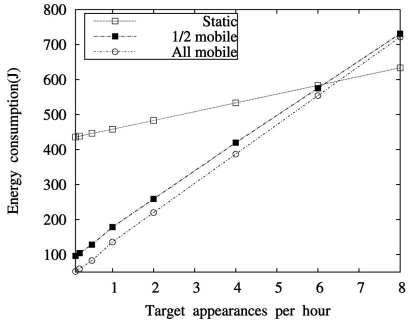

Fig. 20. Energy consumption versus target appearances per hour.

sensor in each move (that takes 0.75 s) is randomly chosen within $[(1-\delta)v, (1+\delta)v]$. As shown in Fig. 18, PD yields a higher variation when δ is larger. However, the average PD remains similar even δ increases to 50 percent. This is because our algorithm always schedules multiple sensors to move in order to achieve the maximum PD. As a result, the variation in each individual sensor's speed does not have a significant impact on the overall system PD.

7.6 Network Costs

In this section, we evaluate the number of sensors and energy consumption, which are important cost metrics of WSNs, required by static, hybrid, and mobile networks. Fig. 19 plots the number of sensors required for achieving the requested PD. The requested PF is 1 percent. We note that for the all mobile and one-half mobile networks, the mobile sensors are scheduled to move as far as possible to maximize the PD. We can see from the figure that the static network needs about seven times more sensors than the all mobile network, and the one-half mobile network needs about twice more sensors than the all mobile network. Therefore, the use of mobile sensors can significantly reduce the number of sensors needed in a deployment.

We also evaluate the corresponding energy consumptions of various networks. In our simulations, we account for the energy consumed in locomotion of mobile sensors, transmission, and idle listening of radios. We assume that the mobile nodes are wheeled robots such as Robomote [14]. The energy consumed in locomotion by a wheeled robot, denoted by $E_M(d)$, can be approximated by $E_M(d) = k \cdot d$ [38], where d is the moving distance and $k = 2\ \mathrm{J/m}$ if the mobile node moves at optimal speed. For typical low-power transceivers such as CC2420, the energy consumed in wireless communication, denoted by $E_C(d)$, can be modeled as $E_C(d) = m \cdot (a + b \cdot d^2)$ [32], [18], where d is the transmission distance, m is the number of bits transmitted, and a and b are constants. In our simulations, a and b are set to be $0.6 \times 10^{-7}\ \mathrm{J/bit}$ and $4 \times 10^{-10}\ \mathrm{J/m^2 \cdot bit}$ according to the real experiments in [32]. The power consumption of an idle node is set to be $21\ \mathrm{mW}$, which is consistent with that of TelosB mote [13]. We ignore the power consumption of a sleeping node, as it is much less than the idle state power consumption. For instance, a TelosB mote consumes $1\ \mu\mathrm{W}$ in sleeping mode [13].

Fig. 20 plots the total energy consumption over 24 hours versus the number of target appearances per hour. The requested detection delay D is set to be $30\ \mathrm{s}$. Therefore, the sensors are woken up in every $30\ \mathrm{s}$ to make measurements. We can see from the figure that all mobile and one-half mobile networks outperform the static network in terms of energy consumption when the target appearance rate is low and the static network becomes superior when target appearance rate is high. This is because the frequent movement cancels out the benefit of energy consumption reduction in wireless communication due to the reduction of sensors.

7.7 Performance with Realistic Target Mobility

This section presents the simulation results under realistic sensor deployment and target mobility model. In the simulations, 17 sensors are deployed and their initial positions are set according to the deployment in the SensIT vehicle experiments [17]. The targets in our simulation follow the trajectories of the AAV runs in the SensIT experiments. Fig. 23 plots the initial deployment, the trajectory of an AAV run, and the chosen surveillance location.

Fig. 21 shows PD when the number of mobile sensors varies from 1 to 6 under various requested PF. In each setting,

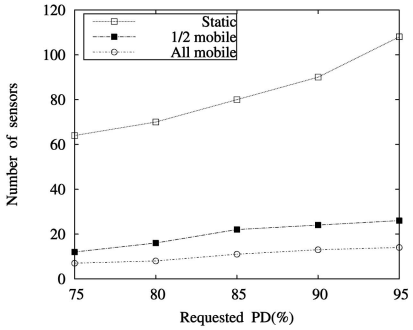

Fig. 19. The number of sensors required versus requested PD. $\mathrm{PF} = 1$ percent.

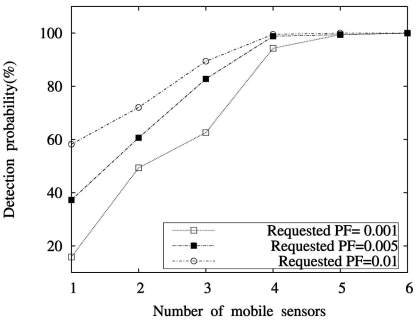

Fig. 21. PD versus number of mobile sensors.

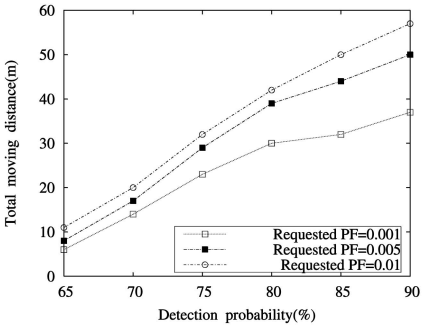

Fig. 22. Total moving distance versus requested PD.

the sensors closest to the surveillance location are chosen to be the mobile sensors and a solution is computed using the algorithm in Fig. 4. We then conduct a number of runs to estimate the PD. We can see that PD is close to 100 percent when only five sensors are mobile. We note that as the requested PF is very low ($\leq 1\%$), the PD is low when few mobile sensors are available. We can conclude from Fig. 21 that a small number of mobile sensors can boost the detection performance of the network.

Fig. 22 plots the total moving distance in the solution computed by our movement scheduling algorithm versus the requested PD under various requested PF. Only three mobile sensors are used. We can see that the average moving distance for a mobile sensor is about 3-12 m if the requested PF is 1 percent. Such a moving distance is acceptable for resource-constrained mobile sensors.

8 Conclusion

This paper explores the use of mobile sensors to address the limitation of static WSNs for target detection. In our approach, mobile sensors initially stationary are triggered to move toward possible target locations by a detection consensus arrived at by all sensors. The fidelity of final detection decision is then improved by a second-phase detection that fuses the measurements of both static and mobile sensors. We develop an *optimal* sensor movement scheduling algorithm that enables mobile sensors to gather the maximum amount of target energy under a given moving distance bound. The effectiveness of our approach is validated by extensive simulations based on real data traces.

Acknowledgments

The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 122307, Project No. CityU 121107). The authors thank Ke Shen for conducting some experiments in the early stage of this work.

References

[1] M.A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G.S. Sukhatme, W.J. Kaiser, M. Hansen, G.J. Pottie, M. Srivastava, and D. Estrin, "Call and Response: Experiments in Sampling the Environment," *Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys),* 2004.

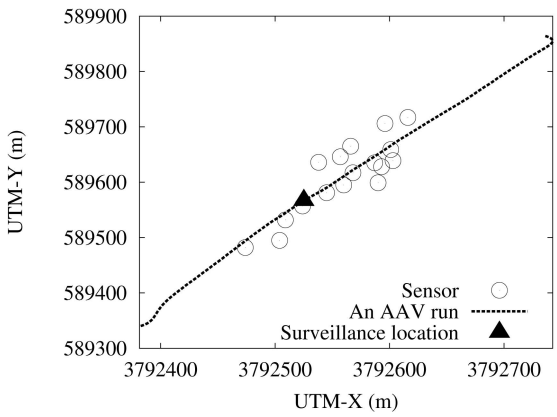

Fig. 23. Sensor deployment in SensIT experiments [17].

[2] N. Bisnik, A. Abouzeid, and V. Isler, "Stochastic Event Capture Using Mobile Sensors Subject to a Quality Metric," *Proc. ACM MobiCom,* 2006.

[3] H. Cao, E. Ertin, V. Kulathumani, M. Sridharan, and A. Arora, "Differential Games in Large-Scale Sensor-Actuator Networks," *Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2006.

[4] J.-F. Chamberland and V. Veeravalli, "Decentralized Detection in Sensor Networks," *IEEE Trans. Signal Processing,* vol. 51, no. 2, pp. 407-416, Feb. 2003.

[5] S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying Wireless Sensor Networks under Limited Mobility Constraints," *IEEE Trans. Mobile Computing,* vol. 6, no. 10, pp. 1142-1157, Oct. 2007.

[6] W.-P. Chen, J.C. Hou, and L. Sha, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 3, no. 3, pp. 258-271, July 2004.

[7] J.-C. Chin, Y. Dong, W.-K. Hon, and D. Yau, "On Intelligent Mobile Target Detection in a Mobile Sensor Network," *Proc. Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS),* 2007.

[8] T.-L. Chin, P. Ramanathan, and K.K. Saluja, "Analytic Modeling of Detection Latency in Mobile Sensor Networks," *Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2006.

[9] T.-L. Chin, P. Ramanathan, K.K. Saluja, and K.-C. Wang, "Exposure for Collaborative Detection Using Mobile Sensor Networks," *Proc. Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS),* 2005.

[10] H. Choset, "Coverage for Robotics—A Survey of Recent Results," *Annals of Math. and Artificial Intelligence,* vol. 31, no. 1, pp. 113-126, 2001.

[11] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K.K. Saluja, "Sensor Deployment Strategy for Target Detection," *Proc. Int'l Workshop Wireless Sensor Networks and Applications (WSNA),* Sept. 2002.

[12] T. Clouqueur, K.K. Saluja, and P. Ramanathan, "Fault Tolerance in Collaborative Sensor Networks for Target Detection," *IEEE Trans. Computers,* vol. 53, no. 3, pp. 320-333, Mar. 2004.

[13] Crossbow Technology, Inc., Telosb Datasheet, 2005.

[14] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G.S. Sukhatme, "Robomote: Enabling Mobility in Sensor Networks," *Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2005.

[15] S. Dhillon, K. Chakrabarty, and S.S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise," *Proc. Int'l Conf. Information Fusion (FUSION '02),* 2002.

[16] M. Duarte and Y.-H. Hu, "Distance Based Decision Fusion in a Distributed Wireless Sensor Network," *Telecomm. Systems,* vol. 26, nos. 2-4, pp. 339-350, 2004.

[17] M.F. Duarte and Y.H. Hu, "Vehicle Classification in Distributed Sensor Networks," *J. Parallel and Distributed Computing,* vol. 64, no. 7, pp. 826-838, July 2004.

[18] F. El-Moukaddem, E. Torng, G. Xing, and S. Kulkarni, "Mobile Relay Configuration in Data-Intensive Wireless Sensor Networks," *Proc. Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS),* 2009.

[19] M. Hata, "Empirical Formula for Propagation Loss in Land Mobile Radio Services," *IEEE Trans. Vehicular Technology,* vol. VT-29, no. 3, pp. 317-325, Aug. 1980.
[20] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-Efficient Surveillance System Using Wireless Sensor Networks," *Proc. ACM Mobisys,* 2004.
[21] Y. Hwang and N. Ahuja, "Gross Motion Planning: A Survey," *ACM Computing Surveys,* vol. 24, no. 3, pp. 219-291, 1992.
[22] J. Latombe, *Robot Motion Planning.* Kluwer Academic Publishers, 1991.
[23] H. Lau, S. Huang, and G. Dissanayake, "Probabilistic Search for a Moving Target in an Indoor Environment," *Proc. Int'l Conf. Intelligent Robots and Systems,* 2006.
[24] D. Li and Y.H. Hu, "Energy Based Collaborative Source Localization Using Acoustic Micro-Sensor Array," *J. EUROSIP on Applied Signal Processing,* vol. 4, pp. 321-337, 2003.
[25] D. Li, K. Wong, Y.H. Hu, and A. Sayeed, "Detection, Classification and Tracking of Targets in Distributed Sensor Networks," *IEEE Signal Processing Magazine,* vol. 19, no. 2, pp. 17-29, Mar. 2002.
[26] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility Improves Coverage of Sensor Networks," *Proc. ACM MobiHoc,* 2005.
[27] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2003.
[28] D. Lymberopoulos and A. Savvides, "Xyz: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications," *Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2005.
[29] M. Rahimi, M. Hansen, W.J. Kaiser, G.S. Sukhatme, and D. Estrin, "Adaptive Sampling for Environmental Field Estimation Using Robotic Sensors," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS),* 2005.
[30] N. Roy and C. Earnest, "Dynamic Action Spaces for Information Gain Maximization in Search and Exploration," *Proc. Am. Control Conf. (ACC),* 2006.
[31] L. Schenato, S. Oh, S. Sastry, and P. Bose, "Swarm Coordination for Pursuit Evasion Games Using Sensor Networks," *Proc. Int'l Conf. Robotics and Automation,* 2005.
[32] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang, "C-Mac: Model-Driven Concurrent Medium Access Control for Wireless Sensor Networks," *Proc. IEEE INFOCOM,* 2009.
[33] A.A. Somasundara, A. Ramamoorthy, and M.B. Srivastava, "Mobile Element Scheduling with Dynamic Deadlines," *IEEE Trans. Mobile Computing,* vol. 6, no. 4, pp. 395-410, Apr. 2007.
[34] R. Tan, G. Xing, J. Wang, and H.C. So, "Collaborative Target Detection in Wireless Sensor Networks with Reactive Mobility," *Proc. Int'l Workshop Quality of Service (IWQoS),* 2008.
[35] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous Localization, Calibration, and Tracking in an Ad Hoc Sensor Network," *Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2006.
[36] P. Varshney, *Distributed Detection and Data Fusion.* Spinger-Verlag, 1996.
[37] G. Wang, G. Cao, and T.L. Porta, "Movement-Assisted Sensor Deployment," *IEEE Trans. Mobile Computing,* vol. 5, no. 6, pp. 640-652, June 2006.
[38] G. Wang, M. Irwin, P. Berman, H. Fu, and T. La Porta, "Optimizing Sensor Movement Planning for Energy Efficiency," *Proc. Int'l Symp. Low Power Electronics and Design,* pp. 215-220, 2005.
[39] W. Wang, V. Srinivasan, and K.-C. Chua, "Trade-Offs between Mobility and Density for Coverage in Wireless Sensor Networks," *Proc. ACM MobiCom,* 2007.
[40] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C.-W. Yi, "Data Fusion Improves the Coverage of Wireless Sensor Networks," *Proc. ACM MobiCom,* 2009.
[41] Z. Yuan, R. Tan, G. Xing, C. Lu, Y. Chen, and J. Wang, "Fast Sensor Placement Algorithms for Fusion-Based Target Detection," *Proc. Real-Time Systems Symp. (RTSS),* 2008.
[42] Y. Zou and K. Chakrabarty, "Distributed Mobility Management for Target Tracking in Mobile Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 6, no. 8, pp. 872-887, Aug. 2007.

Guoliang Xing received the BS degree in electrical engineering and the MS degree in computer science from Xi'an Jiao Tong University, China, in 1998 and 2001, respectively, and the MS and DSc degrees in computer science and engineering from Washington University, St. Louis, in 2003 and 2006, respectively. He is an assistant professor in the Department of Computer Science and Engineering at Michigan State University. From 2006 to 2008, he was an assistant professor of computer science at City University of Hong Kong. He served on a number of technical program committees and held several workshop organization positions including the program cochairs of the First ACM International Workshop on Heterogeneous Sensor and Actor Networks (HeterSanet) 2008 and the Workshop on Wireless Ad hoc and Sensor Networks (WWASN) 2008 and 2009. His research interests include power management and controlled mobility in wireless sensor networks, data-fusion-based network design and analysis, and cyberphysical systems. He is a member of the IEEE.

Jianping Wang received the BS and MS degrees in computer science from Nankai University, China, in 1996 and 1999, respectively, and the PhD degree from the University of Texas at Dallas in 2003. She is currently an assistant professor of computer science at City University of Hong Kong. Prior to joining City University of Hong Kong, she worked at the University of Mississippi and Georgia Southern University. Her research interests are optical networks, multicast, dependable networking, wireless networks, and the integration of optical networks and wireless networks. She is a member of the IEEE.

Zhaohui Yuan received the BS degree in computer science from Huazhong Normal University, China, in 2004, and the PhD degree in computer science from Wuhan University, China, in 2009. Currently, he is a research member at the Next Generation Networks Research Center, City University of Hong Kong. His research interests include wireless sensor networks, real-time and embedded systems, and mobile computing.

Rui Tan received the BS and MS degrees in automation from Shanghai Jiao Tong University, China, in 2004 and 2007, respectively. He received the PhD degree from the Department of Computer Science, City University of Hong Kong in 2010. He is currently a postdoctoral researcher at Michigan State University. His research interests include data fusion, controlled mobility, and sensing coverage in wireless sensor networks. He is a student member of the IEEE.

Limin Sun received the MS and PhD degrees from the National University of Defense Technology, Changsha, China, in 1995 and 1998, respectively. From June 1998 to June 2000, he was a postdoctoral fellow in the Institute of Software, Chinese Academy of Sciences, Beijing, China. He is currently a professor and the director for Wireless Ad Hoc Network Lab of the Institute of Software. His current research interests include wireless sensor networks, vehicular ad hoc networks, and wireless broadband access networks.

Qingfeng Huang received the DSc degree in computer science from Washington University in St. Louis in 2003, and the AM degree in physics. He has been a member of the technical staff at C8 MediSensors, Inc., since November 2008. Before that, he had been a research scientist at the Palo Alto Research Center (PARC) since 2003. He has 29 patents filed and granted. He has also published more than 40 academic papers in areas including software engineering, context-aware and mobile computing, sensor networks, intelligent transportation systems, neuroscience, and quantum physics. His current R&D interests include medical sensors, algorithms and software for robotic networks, transportation networks and sensor actuator networks, tools enabling knowledge sharing and harvesting in healthcare and education, noninvasive brain-interface technology, and mechanisms that facilitate innovation. He is a member of the IEEE.

Xiaohua Jia received the BSc and MEng degrees from the University of Science and Technology of China, in 1984 and 1987, respectively, and the DSc degree in information science from the University of Tokyo in 1991. He is currently the chair professor in the Department of Computer Science at City University of Hong Kong. His research interests include distributed systems, computer networks, wireless sensor networks, and mobile wireless networks. He is an editor of the *IEEE Transaction on Parallel and Distributed Systems*, *Wireless Networks*, *Journal of World Wide Web*, *Journal of Combinatorial Optimization*, etc. He is the general chair of the ACM MobiHoc 2008, the TPC cochair of the IEEE MASS 2009, and the area chair of the IEEE INFOCOM 2010. He is a senior member of the IEEE.

Hing Cheung So received the BEng degree in electronic engineering from City University of Hong Kong in 1990, and the PhD degree in electronic engineering from The Chinese University of Hong Kong in 1995. From 1990 to 1991, he was an electronic engineer at the Research & Development Division of Everex Systems Engineering, Ltd., Hong Kong. From 1995 to 1996, he worked as a postdoctoral fellow at The Chinese University of Hong Kong. From 1996 to 1999, he was a research assistant professor in the Department of Electronic Engineering, City University of Hong Kong. Currently, he is an associate professor in the Department of Electronic Engineering at City University of Hong Kong. His research interests include fast and adaptive filter, signal detection, parameter estimation, and source localization. He is a senior member of the IEEE.