Exploiting Reactive Mobility for Collaborative Target Detection in Wireless Sensor Networks

Rui Tan, *Student Member*, *IEEE*, Guoliang Xing, *Member*, *IEEE*,
Jianping Wang, *Member*, *IEEE*, and Hing Cheung So, *Senior Member*, *IEEE*

Abstract—Recent years have witnessed the deployments of wireless sensor networks in a class of mission-critical applications such as object detection and tracking. These applications often impose stringent Quality-of-Service requirements including high detection probability, low false alarm rate, and bounded detection delay. Although a dense all-static network may initially meet these Quality-of-Service requirements, it does not adapt to unpredictable dynamics in network conditions (e.g., coverage holes caused by death of nodes) or physical environments (e.g., changed spatial distribution of events). This paper exploits reactive mobility to improve the target detection performance of wireless sensor networks. In our approach, mobile sensors collaborate with static sensors and move reactively to achieve the required detection performance. Specifically, mobile sensors initially remain stationary and are directed to move toward a possible target only when a detection consensus is reached by a group of sensors. The accuracy of final detection result is then improved as the measurements of mobile sensors have higher Signal-to-Noise Ratios after the movement. We develop a sensor movement scheduling algorithm that achieves *near-optimal* system detection performance under a given detection delay bound. The effectiveness of our approach is validated by extensive simulations using the real data traces collected by 23 sensor nodes.

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

1 Introduction

In recent years, wireless sensor networks have been deployed in a class of mission-critical applications such as target detection [1], object tracking [2], and security surveillance [3]. A fundamental challenge for these wireless sensor networks is to meet stringent Quality-of-Service requirements including high target detection probability, low false alarm rate, and bounded detection delay. However, physical phenomena (e.g., the appearance of intruders) often have unpredictable spatiotemporal distributions. As a result, a large network deployment may require excessive sensor nodes in order to achieve satisfactory sensing performance. Moreover, although dense node deployment may initially achieve the required performance, it does not adapt to dynamic changes of network conditions or physical environments. For instance, death of nodes due to battery depletion or physical attacks can easily cause coverage holes in a monitored battlefield.

In this paper, we exploit reactive mobility to improve the target detection performance of wireless sensor networks. In our approach, sparsely deployed mobile sensors collaborate with static sensors and move in a *reactive* manner to achieve required detection performance. Specifically, mobile sensors remain stationary until a possible target is detected. The accuracy of the final detection decision will be improved after mobile sensors move toward the possible target position and achieve higher Signal-to-Noise Ratios. By taking advantage of such reactive mobility, a network can adapt to irregular and unpredictable spatiotemporal distribution of targets. Moreover, the sensor density required in a network deployment is significantly reduced because the sensing coverage can be reconfigured in an on-demand fashion.

Several challenges must be addressed for utilizing the mobility of sensors in target detection. First, practical mobile sensors are only capable of slow-speed movement, which may lead to long detection delays. The typical speed of mobile sensor systems (e.g., Networked Infomechanical Systems [4], Packbot [5], and Robomote [6]) is about 0.2-2 m/s. Therefore, the movement of sensors must be efficiently scheduled in order to reduce detection latency. Second, the number of mobile sensors available in a network deployment is often much smaller than that of static sensors due to higher manufacturing cost. Hence, mobile sensors must effectively collaborate with static sensors to achieve the maximum utility. At the same time, the coordination among sensors should not introduce high overhead or significant detection delay. Third, the distance that mobile sensors move in a detection process should be minimized. Due to the high power consumption of locomotion, frequent movement will quickly deplete the battery of a mobile node. For instance, a Robomote [6] sensor needs to recharge every 20 minutes when constantly moving. Although mobile sensors may recharge their batteries by moving to locations with wired power supplies, frequent battery recharging causes disruptions to network topologies. Finally, moving sensors lowers the stealthiness of a network, which is not desirable for many applications deployed in hostile environments like battlefields.

In this work, we attempt to address the aforementioned challenges and demonstrate the advantages of reactive

- *R. Tan and J. Wang are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: tanrui2@student.cityu.edu.hk, jianwang@cityu.edu.hk.*
- *G. Xing is with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824-1226. E-mail: glxing@msu.edu.*
- *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 17 Apr. 2008; revised 12 Feb. 2009; accepted 25 June 2009; published online 1 July 2009.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2008-04-0147. Digital Object Identifier no. 10.1109/TMC.2009.125.

mobility in target detection wireless sensor networks. This paper makes the following major contributions:

- We present a new formulation for the problem of target detection based on a novel two-phase detection approach. Our formulation accounts for stringent performance requirements imposed by mission-critical detection application including high detection probability, low false alarm rate, and bounded detection delay. In the two-phase detection approach, mobile sensors initially remain stationary and are directed to move toward a possible target only when a detection consensus is reached by all nearby sensors. Such a strategy allows mobile sensors to avoid unnecessary movement through the collaboration with static sensors.
- We develop a *near-optimal* movement scheduling algorithm based on dynamic programming that minimizes the expected moving distance of mobile sensors. Our scheduling algorithm also enables mobile sensors to locally control their movement and sensing. Thus, both coordination overhead and detection delay are reduced significantly. Although the algorithm is mainly designed for stationary target detection at fixed locations, we also discuss the extensions to more general cases such as detecting moving targets.
- We conduct extensive simulations using real data traces collected by 23 sensors in a real vehicle detection experiment [7]. Our results provide several important insights into the design of target detection systems with mobile sensors. First, we show that a small number of mobile sensors can significantly boost the detection performance of a network. Second, tight detection delays can be achieved by efficiently scheduling slow-moving mobile sensors.

The rest of the paper is organized as follows: Section 2 reviews related work. Sections 3 and 4 introduce the preliminaries and the formulation of our problem. The performance of the proposed two-phase detection approach is studied in Section 5. Section 6 presents a near-optimal movement scheduling algorithm. Several extensions are discussed in Section 7. Section 8 offers simulation results, and Section 9 concludes this paper.

2 Related Work

Recent works demonstrate that the sensing performance of wireless sensor networks can be improved by integrating sensor mobility. Several projects propose to eliminate coverage holes in a sensing field by relocating mobile sensors [8], [9], [10]. Although such an approach improves the sensing coverage of the initial 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 after the appearance of targets.

Several recent studies [11], [12] analyze the impact of mobility on detection delay and area coverage. These studies are based on random mobility models and do not address the issue of actively controlling the movement of sensors. Bisnik et al. [13] analyze the performance of detecting stochastic events using mobile sensors. Chin et al. [14] propose to improve coverage by patrolling fixed routes using mobile sensors. Different from these works, we study efficient sensor collaboration and movement scheduling strategies that achieve specified target detection performance. Reactive mobility is used in a networked robotic sensor architecture [15], [16] to improve the sampling density over a region. However, this project does not focus on target detection under performance constraints. In our recent work [17], reactive mobility is exploited to meet the constraints on target detection performance. Different from [17] which focuses on centralized detection and sensor movement schemes, this work employs distributed schemes that are designed to meet the resource constraints of sensor networks. First, each mobile sensor in our solution controls its movement and makes detection decisions independently. Moreover, this work adopts the decision fusion model that leads to significantly lower communication cost than the value fusion model in [17].

Collaborative target detection in static sensor networks has been extensively studied [1], [18]. The two-phase detection approach proposed in this paper is based on an existing decision fusion model [18]. Several projects study the network deployment strategies that can achieve specified detection performance under collaborative target detection models [19], [20]. Our recent work [21] investigates the fundamental impacts of data fusion on the coverage of wireless sensor networks. Practical network protocols that facilitate target detection/tracking using static or mobile sensors have also been investigated [1], [2], [3], [22], [23], [24]. Complementary to these studies that deal with the mobility of targets, we focus on improving target detection performance by utilizing sensors' mobility.

Several recent studies [25], [26], [27] formulate target detection/tracking in mobile sensor networks as a game problem and propose several motion strategies for mobile sensors. In these studies, the mobile sensors move *actively* to improve the surveillance quality and 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 [28]. We refer interested readers to [29], [30] for comprehensive surveys on this topic. Sensor movement scheduling 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 high target detection probability, low false alarm rate, and bounded detection delay.

3 Preliminaries

In this section, we describe the preliminaries of our work, which include the sensor measurement model (Section 3.1), the detection and multisensor decision fusion models (Section 3.2), and the network and sensor mobility models (Section 3.3).

3.1 Sensor Measurement Model

Sensors perform detection by measuring the energy of signals emitted by the target. The energy of most physical signals (e.g., acoustic and electromagnetic signals) attenuates with the distance from the signal source. Suppose sensor i is x_i meters away from the target that emits a signal of energy S_0, the attenuated signal energy $e_s(x_i)$ at the position of sensor i is given by

$$e_s(x_i) = S_0 \cdot w(x_i), \tag{1}$$

where $w(\cdot)$ is a decreasing function satisfying $w(0) = 1$ and $w(\infty) = 0$. The $w(\cdot)$ is referred to as the *signal decay function*. In this paper, we adopt the two-dimensional polar coordinate system with the target position as the origin. As the signal decay model in (1) is isotropic and the detection scheme adopted in this paper is based on the signal energy, we omit the angular coordinate, and thus, scalar x_i can be referred to as the position of sensor i.

The sensor measurements are contaminated by additive random noise from environment, sensor hardware, and other affecting random phenomena. Depending on the hypothesis that the target is absent (H_0) or present (H_1), the energy measurement of sensor i, denoted by e_i, is given by

$$\begin{aligned} H_0: &\quad e_i = e_n, \\ H_1: &\quad e_i = e_s(x_i) + e_n, \end{aligned}$$

where e_n is the energy of noise experienced by sensor i. In practice, an energy measurement at a sensor is often estimated by the arithmetic average over a number of samples during a *sampling interval* of T seconds [7], [31]. Suppose the number of samples in a sampling interval is K, the noise energy is given by $e_n = \frac{1}{K}\sum_{j=1}^{K} \nu_j^2$, where ν_j is the noise intensity when taking the jth sample. We assume that the noise intensity ν_j is independent and identically distributed. If K is large enough, the noise energy e_n follows the normal distribution according to the Central Limit Theorem. Specifically, $e_n \sim \mathcal{N}(\mathbb{E}[\nu_j^2]/K, \mathrm{Var}[\nu_j^2]/K)$, where $\mathbb{E}[\nu_j^2]$ and $\mathrm{Var}[\nu_j^2]$ are the mean and variance of ν_j^2, respectively. The number of samples K is often large in practice. For instance, acoustic data are recorded at a frequency of 4,960 Hz in the vehicle detection experiments under US Defense Advanced Research Projects Agency (DARPA)/ITO Sensor Information Technology Program [7]. If the sampling interval T is 0.75 s which is adopted in the Sensor Information Technology experiments, K is $4{,}960 \times 0.75 = 3{,}720$. We note that such a sampling scheme does not incur high computation or energy overhead to a sensor. First, our detection model only requires the average value of the samples, which can be computed efficiently. Second, the power consumption of sensors is usually low compared to that of radio communication. We denote $\mu = \mathbb{E}[\nu_j^2]/K$ and $\sigma^2 = \mathrm{Var}[\nu_j^2]/K$ in the remainder of this paper.

The above signal decay and sensor measurement models have been widely assumed in the literature of signal detection [18], [31], [32] and also have been empirically verified [33], [34]. We note that the algorithm proposed in this paper does not depend on the specific form of the signal decay function $w(\cdot)$. In practice, the parameters of the sensor measurement model (i.e., S_0, $w(\cdot)$, μ, and σ^2) can be estimated using training data. In particular, for acoustic sensors, the signal decay function can be expressed as follows [31], [32], [34]:

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

where k is the decay factor and d_0 is a constant determined by target's shape. It is verified in [34] that $k \simeq 2$ using the real data traces collected in the aforementioned Sensor Information Technology experiments [7]. 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 Sensor Information Technology experiments.

3.2 Detection and Decision Fusion Models

Data fusion [18] is a widely used technique for improving the performance of detection systems. There exist two basic data fusion schemes, namely, *value fusion* and *decision fusion*. In value fusion [35], each sensor sends its raw energy measurements to the cluster head, which makes the detection decision based on the received energy measurements. Different from value fusion, decision fusion operates in a distributed manner as follows: Each sensor makes a *local* decision based on its measurements and sends its decision to the cluster head, which makes a *system* decision according to the local decisions. Due to its low overhead, decision fusion is preferred in the bandwidth-constrained wireless sensor networks. Moreover, decision fusion allows mobile sensors to locally control their movement and sensing, as stated in Section 4.1.

Many fusion rules have been proposed in the literature [18] for different detection systems. In this work, we adopt the majority rule due to its simplicity. Specifically, each individual sensor first makes a local detection decision (0 or 1) by comparing the energy measurement against a detection threshold, and reports its local decision to the cluster head. The cluster head makes the system decision by the majority rule, i.e., if more than half of sensors vote 1, the cluster head decides 1; otherwise, it decides 0.

The detection performance is usually characterized by two metrics, namely, the false alarm rate (PF) and detection probability (PD) [18], [32], [35]. PF is the probability of making a positive decision when *no* target is present, and PD is the probability that a present target is correctly detected. The optimal decision rule at sensor i is the Likelihood Ratio Test [18] in which sensor i compares its energy measurement e_i with a detection threshold λ_i. Let I_i denote the local decision of sensor i. If $e_i < \lambda_i$, $I_i = 0$; otherwise, $I_i = 1$. Hence, the *local* false alarm rate and detection probability, denoted by P_F^i and P_D^i, respectively, are given by

$$P_F^i = \Pr(e_i \ge \lambda_i \mid H_0) = Q\left(\frac{\lambda_i - \mu}{\sigma}\right), \tag{3}$$

$$P_D^i = \Pr(e_i \ge \lambda_i \mid H_1) = Q\left(\frac{\lambda_i - \mu - e_s(x_i)}{\sigma}\right), \tag{4}$$

where $Q(\cdot)$ is the complementary Cumulative Distribution Function of the standard normal distribution, i.e., $Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{t^2}{2}) \mathrm{d}t$. Obviously, given the detection threshold, the closer the sensor is from the target, the higher local detection probability it achieves.

Suppose there are total N sensors in a detection cluster, the *system* false alarm rate and detection probability, denoted by P_F and P_D, respectively, are expressed as $P_F = \Pr(Y \geq \frac{N}{2} \mid H_0)$ and $P_D = \Pr(Y \geq \frac{N}{2} \mid H_1)$, where Y represents the total number of positive local decisions, i.e., $Y = \sum_{i=1}^{N} I_i$.

3.3 Network and Sensor Mobility Models

The network is composed of a number of static and mobile sensors. We assume that all sensors are homogeneous. That is, they sense the same type of signal from the target, e.g., acoustic signal. This assumption is relaxed in Section 7.3. Targets appear at a set of known physical locations referred to as *surveillance spot*s with certain probabilities. Surveillance spots are often identified by the network autonomously after the deployment. Therefore, it is impossible to deploy sensors only around surveillance spots. We note that the monitored phenomenon in many applications is spatially distributed. However, the exact spatial distribution is often unknown or complex. In Section 7.4, we briefly discuss how to extend our approach to detect spatially distributed targets.

We assume that each sensor knows its position (through a GPS unit mounted on it or a localization service in the network) and all sensors have synchronized clocks. The network is organized into a cluster-based topology such that each cluster monitors a surveillance spot. All member nodes in a cluster can communicate with the cluster head directly. The clusters can be dynamically formed around the surveillance spots by running a clustering protocol [23] during the network initialization or when the surveillance spots have changed. The above surveillance model is consistent with several previous works [36], [37]. We assume that each static sensor belongs to only one cluster. However, a mobile sensor may belong to multiple clusters because it can contribute to the detection at different surveillance spots.

We now briefly discuss how the above network model can be applied to a target detection application. Suppose a number of static and mobile sensors are randomly deployed (e.g., dropped off from an aircraft) in a battlefield to detect military targets. After operating for a certain amount of time, the network may identify some important locations (e.g., based on detection history) as surveillance spots. A cluster is then formed around each spot to perform the detection.

As each cluster performs detection separately, our discussion focuses on one cluster hereafter. We make the following assumptions:

1. The surveillance spot is at the origin of polar coordinate plane, and the initial position of sensor i is x_i^0.
2. The probability that a target appears at the surveillance spot is P_a, which is known or can be estimated by detection history. The impact of inaccurate P_a and how to estimate P_a when it is not known are discussed in Section 7.1.
3. The target keeps stationary at the surveillance spot and its appearance time is much longer than detection delay. We note that the target is allowed to move slowly if its movement introduces small errors to sensors' signal energy measurements. In Section 7.4, we briefly discuss how to extend our approach to detect moving targets.
4. Mobile sensors are assumed to be able to move continuously in any direction at a constant speed of v, which is referred to as *free mobility model*. Under such a model, the most efficient movement for a mobile sensor to improve detection performance is to move directly toward the surveillance spot. Other mobility models are discussed in Section 7.2.
5. To simplify the motion control of mobile sensors, we assume that the moving distance of a mobile sensor is always multiple of vT.
6. Furthermore, to simplify the problem formulation, we assume that the distance between a sensor and a surveillance spot is also multiple of vT.

We note that the latter two assumptions have little impact on the spatiotemporal precision of detection, as both v and T are small in practice. For instance, T is 0.75 s in the Sensor Information Technology experiments and v is 0.2-2 m/s for typical mobile sensor systems [4], [5], [6]. Under such settings, vT is at most 1.5 m.

Based on the above assumptions, we now formally define the *sensor move* and *movement schedule*.

Definition 1. *A* sensor move, *denoted by $\mathcal{M}_i(x, j)$, is the process in which mobile sensor i moves from position x to $x - vT$ in time duration $[(j-1)T, jT]$, where T is the sampling interval and $j \geq 1$.*

Definition 2. *A* movement schedule, *denoted by $\mathbf{S} = \{\mathcal{M}_i(x, j) | i \in [1, M]\}$, is a collection of sensor moves, where M is the number of mobile sensors in the cluster.*

We note that the movement schedule for a mobile sensor may be temporally inconsecutive, e.g., a mobile sensor moves in the first and third sampling intervals, respectively, while keeping stationary in the second sampling interval. However, we prove in Theorem 2 that the optimal movement schedule for a sensor is temporally consecutive from the beginning of the second phase.

Table 1 summarizes the notation used in this paper.

4 Mobility-Assisted Target Detection with Decision Fusion

This section formulates our problem. A two-phase detection approach is proposed and the problem is formally formulated in Section 4.1. The problem is illustrated with a numerical example in Section 4.2.

4.1 Problem Formulation and Approach Overview

We first formalize *detection performance requirement* as follows:

Definition 3. *The* detection performance requirement *is characterized by a 3-tuple $< \alpha, \beta, D >$. Specifically, for any target that appears at the surveillance spot: 1) the system false*

TABLE 1
Summary of Notation

Symbol	Definition
x_i	position of sensor i in the polar coordinate plane
$e_s(x_i)$	attenuated signal energy at the position of sensor i
T	temporal duration of a sampling interval
μ, σ^2	mean and variance of noise energy, respectively
e_n	noise energy, $e_n \sim \mathcal{N}(\mu, \sigma^2)$
e_i	energy measurement of sensor i, $e_i = e_s(x_i) + e_n$
H_0 / H_1	hypothesis that the target is absent / present
I_i	local detection decision of sensor i, $I_i = 0$ or 1
N	total number of sensors in a detection cluster
M	the number of mobile sensors in a detection cluster
Y	the number of positive local decisions, $Y = \sum_{i=1}^{N} I_i$
P_a	target appearance probability
α / β	upper / lower bound of system PF / PD
D	upper bound of detection delay
λ_1 / λ_2	local detection threshold of the first / second phase
v	movement speed of mobile sensor
$\mathcal{M}_i(x, j)$	a move of mobile sensor i
$\mathbf{S}$	a movement schedule, $\mathbf{S} = \{\mathcal{M}_i(x, j) \mid i \in [1, M]\}$

alarm rate is no higher than α, 2) the system detection probability is no lower than β, and 3) the expected detection delay is no longer than D.

As a static network may not meet a stringent performance requirement, we propose a two-phase detection approach to utilize the mobility of sensors as follows:

1. The target detection is carried out periodically and each *detection cycle* comprises two phases. The length of the detection cycle that can meet the requirement on detection delay is analyzed later in this section.
2. In the first phase, each sensor stays stationary and measures a signal energy for a sampling interval T. It than makes a local decision by comparing against a predefined threshold. Each sensor reports its local decision to the cluster head, which makes a system decision according to the majority rule. If a positive system decision is made, the second phase is initiated; otherwise, the second phase is skipped, and the cluster yields a negative final decision for this cycle.
3. In the second phase, each sensor continuously measures signal energies. Note that each signal energy measurement is gathered for a sampling interval of T. Mobile sensors simultaneously move toward the surveillance spot according to their movement schedules. A sequential fusion-like procedure is adopted at each sensor to make its local decision. Specifically, after each sampling interval, if the sum of signal energies measured by a sensor in this phase exceeds a predefined threshold, the sensor makes a positive local decision and terminates its second-phase detection; otherwise, it continues to sense. When the maximum time duration of the second phase is reached, a sensor makes a negative local decision if its cumulative signal energy is still below the threshold. We note that if a mobile sensor makes a positive local decision, it also terminates its movement no matter whether its movement schedule is completed.

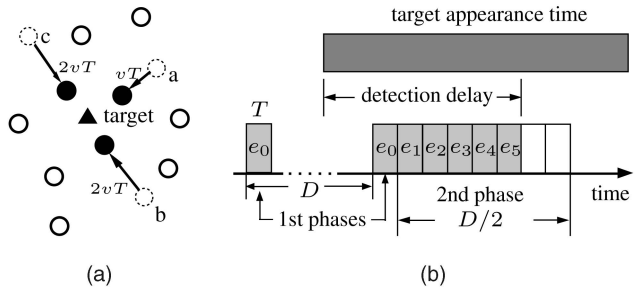


Fig. 1. The illustration of the two-phase detection. (a) Spatial view: void and solid circles represent static and mobile sensors, respectively. The moving distance of a mobile sensor is multiple of vT. (b) Temporal view: the figure draws two detection cycles for a sensor. In the first cycle, the second phase is not initiated as the target is absent. In the second cycle, the sensor terminates its second-phase detection in advance as $\sum_{j=1}^{5} e_j$ exceeds the threshold, although maximum seven sampling intervals are allowed.

4. As soon as enough local decisions for the second-phase detection are received to reach a majority consensus, a positive final detection decision for this cycle is made and the cluster enters the next detection cycle. After the end of the second phase, the mobile sensors shared by multiple clusters may need to move back to their original positions if such movement causes the detection performances of other clusters to be lower than the requirements. Otherwise, these shared mobile sensors stay at the new positions to avoid the energy consumed in moving back.

Such a two-phase approach has several advantages: 1) Unnecessary movement of mobile sensors is avoided, as mobile sensors start to move only after the first-phase detection produces a positive decision. 2) The sequential detection strategy allows each mobile sensor to locally control its sensing and moving according to its movement schedule, which avoids internode coordination overhead. Therefore, only the communication between the cluster head and each member sensor is required. 3) Moreover, as a sensor can terminate its detection and movement schedule in advance if it has enough cumulative signal energy to make a positive decision, the delay of reaching a consensus and the locomotion energy consumption can be reduced.

We now analyze the delay of the two-phase approach to meet the requirement of detection. To ensure a detection delay of D, the period of the detection cycle must be no longer than D. To simplify our discussion, we let the period equal to D. Hence, the expected delay of the first phase is $\frac{D}{2}$ if the time instance at which the target appears is uniformly distributed within D. Accordingly, the time duration of the second phase must be no longer than $\frac{D}{2}$ in order to bound the total expected detection delay within D. We assume that $\frac{D}{2}$ is multiple of T. Therefore, there can be maximum $\frac{D}{2T}$ sampling intervals in the second phase. Consequently, the requirement on detection delay is satisfied. The spatial and temporal views of the two-phase detection are illustrated in Fig. 1.

We define the following notation:

1. λ_1 and λ_2 represent the local detection thresholds for the first-phase detection and the second-phase detection, respectively. Note that all sensors use the same local detection threshold in each phase.

2. P_{F_1} (P_{F_2}) and P_{D_1} (P_{D_2}) represent the system false alarm rate and detection probability in the first (second)-phase detection, respectively, which are derived in Section 5.
3. As a mobile sensor may terminate its movement schedule in advance, the actual number of moves in a detection is a random variable which depends on the detection threshold in the second phase and movement schedule. Let $\mathcal{L}_0(\lambda_2, \mathbf{S})$ and $\mathcal{L}_1(\lambda_2, \mathbf{S})$ represent the total expected numbers of moves of all mobile sensors when the target is absent and present, respectively, which are derived in Section 5.2.

Our objective is to find the detection thresholds λ_1, λ_2, and the movement schedule $\mathbf{S}$, such that the total expected distance that the mobile sensors move away from their original positions is minimized, subject to the performance requirement $< \alpha, \beta, D >$. As discussed in Section 1, the objective of minimizing the total moving distance is motivated by several practical considerations including high power consumption of locomotion and disruptions to the network topology caused by sensor movement. The problem is formally formulated as follows:

Problem 1. *To find a solution* $< \lambda_1, \lambda_2, \mathbf{S} >$ *such that the following cost function is minimized:*

$$c(\lambda_1, \lambda_2, \mathbf{S}) = (1 - P_a) \cdot P_{F_1} \cdot \mathcal{L}_0(\lambda_2, \mathbf{S}) + P_a \cdot P_{D_1} \cdot \mathcal{L}_1(\lambda_2, \mathbf{S}), \tag{5}$$

subject to the following constraints:

$$P_{F_1} \cdot P_{F_2} \leq \alpha, \tag{6}$$

$$P_{D_1} \cdot P_{D_2} \geq \beta, \tag{7}$$

$$\lambda_1 \in \Lambda_1 = \{\lambda_{1(1)}, \lambda_{1(2)}, \ldots, \lambda_{1(q)}\}, \tag{8}$$

$$\lambda_2 \in \Lambda_2 = \{\lambda_{2(1)}, \lambda_{2(2)}, \ldots, \lambda_{2(q)}\}, \tag{9}$$

$$\forall \mathcal{M}_i(x, j) \in \mathbf{S}, \left(vT \leq x \leq x_i^0\right) \wedge \left(1 \leq j \leq \frac{D}{2T}\right). \tag{10}$$

In the cost function given by (5), the second-phase detection is initiated with the probability of $P_a \cdot P_{D_1}$ if the target is present, and $(1 - P_a) \cdot P_{F_1}$ if the target is absent. Therefore, the cost function measures the total expected moving distance of mobile sensors. Equations (6) and (7) are the detection performance required by user. As the decisions of two phases are mutually independent, the joint false alarm rate and detection probability are the product of two phases' false alarm rates and detection probabilities, respectively. Equations (8) and (9) specify discrete values of the two detection thresholds. We assume that the number of options for each threshold is q. In practice, the achievable precision of sensors is low-bounded. Equation (10) specifies the spatial and temporal constraints of sensor movement in the second phase. Each mobile sensor must move between its initial position (i.e., $x \leq x_i^0$) and the surveillance spot (i.e., $x \geq vT$). Furthermore, the movement must complete within $\frac{D}{2T}$ sampling intervals (i.e., $1 \leq j \leq \frac{D}{2T}$), which ensures the detection delay bound.

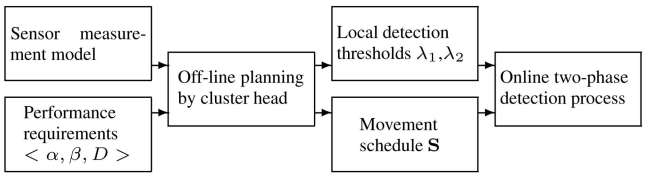


Fig. 2. The overview of the approach.

We now provide a high-level overview of our solution to Problem 1. In Section 5, we derive the closed-form expressions of the false alarm rates and detection probabilities in the two-phase detection. These probabilities will be used to find the solution. In Section 6.1, we analyze the structure of optimal solution. In Section 6.2, we first develop a near-optimal movement scheduling algorithm for given detection thresholds, and then, search the detection thresholds to construct a near-optimal solution.

The overview of our approach is illustrated in Fig. 2. Initially, each cluster head receives the sensor measurement model and performance requirements from the user. The sensor measurement model stated in Section 3.1 is often estimated from training data in practice. The performance requirements specified by user are $< \alpha, \beta, D >$. Each cluster head then solves the offline optimization problem formulated above and yields a solution $< \lambda_1, \lambda_2, \mathbf{S} >$. Finally, each cluster head sends the local detection thresholds and movement schedule to each sensor, which starts the online two-phase detection.

4.2 A Numerical Example

We now illustrate our problem and the basic approach using a numerical example. To simplify the discussion, we assume that there is only one surveillance spot. We also assume that after a possible target appears, a decision consensus is always reached in the first phase of detection, which triggers all mobile sensors to move toward the surveillance spot. The required false alarm rate and detection probability (i.e., α and β) are 5 and 75 percent, respectively. The minimum movement speed of mobile sensors is $1\ \mathrm{m/s}$. During initialization, the cluster head estimates the parameters of sensor measurement model using a real data set obtained from [7] (the details of experimental settings are given in Section 8).

We now discuss three different cases: 1) if all sensors are static, 14 sensors will be needed to achieve the required detection performance within a delay of 2 seconds, as shown in Fig. 3a, 2) if the allowable detection delay is 7 seconds, 10 mobile sensors will be needed as they can move closer to the target resulting in higher Signal-to-Noise Ratios, and 3) if a detection delay of 15 seconds is allowed, only seven mobile sensors are needed as illustrated in Fig. 3c. This is because these sensors are able to move longer distances toward the surveillance spot than in case 2.

Two important observations can be made from this example. First, the detection performance can be significantly improved by taking the advantage of mobility of sensors. Second, scheduling more mobile sensors to move toward a possible target results in a shorter delay. This observation is particularly important as most mobile sensor platforms have low movement speeds.

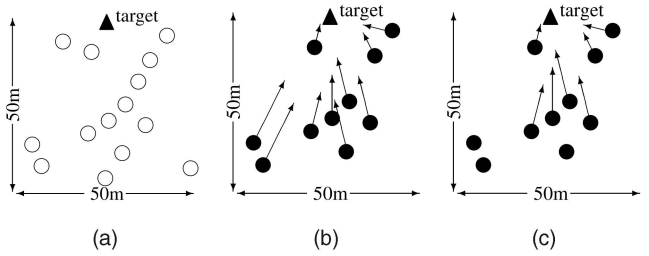


Fig. 3. A numerical example of target detection using static or mobile sensors. (a) 14 static sensors; $P_F = 5\%$, $P_D = 75\%$, and delay is 2 s. (b) 10 mobile sensors; $P_F = 5\%$, $P_D = 75\%$, and delay is 7 s. (c) 7 mobile sensors; $P_F = 5\%$, $P_D = 75\%$, and delay is 15 s.

5 Performance Modeling of Two-Phase Detection

We now derive the false alarm rates and the detection probabilities in the two phases of detection which are used in Section 6 to find the solution of our problem.

5.1 First-Phase Detection

Since the local false alarm rate does not depend on sensor's position, according to (3), all sensors have the same local false alarm rate, denoted by α_1, which is calculated by $\alpha_1 = Q(\frac{\lambda_1-\mu}{\sigma})$. Hence, in the absence of target, the number of positive local decisions follows the Binomial distribution, i.e., $Y \mid H_0 \sim \mathrm{Bin}(N, \alpha_1)$. According to the de Moivre-Laplace Theorem [32], the Binomial distribution $\mathrm{Bin}(N, \alpha_1)$ can be approximated by the normal distribution $\mathcal{N}(N\alpha_1, N\alpha_1 - N\alpha_1^2)$ if $N \geq 10$ [38]. This condition can be met in many moderate- to large-scale network deployments. Therefore, the system false alarm rate in the first phase can be approximated by

$$P_{F_1} = \Pr\left(Y \geq \frac{N}{2} \middle| H_0\right) \simeq Q\left(\frac{\frac{N}{2} - N\alpha_1}{\sqrt{N\alpha_1 - N\alpha_1^2}}\right). \tag{11}$$

We now derive the system detection probability in the first-phase detection. In the presence of target, the local decision at sensor i, $I_i|H_1$, follows the Bernoulli distribution with $\beta_{1,i}$ as the probability of success, where $\beta_{1,i}$ is the local detection probability of sensor i at its original position x_i^0 in the first phase. According to (4), $\beta_{1,i}$ is calculated by $\beta_{1,i} = Q(\frac{\lambda_1-\mu-e_s(x_i^0)}{\sigma})$. As $I_1|H_1, \ldots, I_N|H_1$ are mutually independent, the mean and variance of $Y|H_1$ are given by

$$\mathbb{E}[Y \mid H_1] = \sum_{i=1}^{N} \mathbb{E}[I_i \mid H_1] = \sum_{i=1}^{N} \beta_{1,i}$$

and

$$\mathrm{Var}[Y \mid H_1] = \sum_{i=1}^{N} \mathrm{Var}[I_i \mid H_1] = \sum_{i=1}^{N} \beta_{1,i} - \sum_{i=1}^{N} \beta_{1,i}^2,$$

respectively. However, $I_1|H_1, \ldots, I_N|H_1$ are not identically distributed, as $\beta_{1,i}$ depends on sensor's original position. The following lemma proves the condition for the Lyapunov's Central Limit Theorem [39]. The proof is in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2009.125.

Lemma 1. *Let $\{I_i : i = 1, \ldots, N\}$ be a sequence of mutually independent Bernoulli random variables, the Lyapunov condition holds for this sequence.*

According to Lemma 1 and the Lyapunov's Central Limit Theorem, $Y|H_1$ follows the normal distribution when N is large, i.e., $Y|H_1 \sim \mathcal{N}(\sum_{i=1}^{N} \beta_{1,i}, \sum_{i=1}^{N} \beta_{1,i} - \sum_{i=1}^{N} \beta_{1,i}^2)$. Hence, the system detection probability in the first phase can be calculated by

$$P_{D_1} \simeq Q\left(\frac{\frac{N}{2} - \sum_{i=1}^{N} \beta_{1,i}}{\sqrt{\sum_{i=1}^{N} \beta_{1,i} - \sum_{i=1}^{N} \beta_{1,i}^2}}\right). \tag{12}$$

5.2 Second-Phase Detection

In this section, we analyze the performance of the sequential detection process in the second phase, which includes the false alarm rate, the detection probability, and the expected number of moves under a given movement schedule.

We assume that all sensors in the second-phase detection have the same detection threshold of λ_2. In the absence of target, the local false alarm rate, denoted by α_2, is given by

$$\alpha_2 = 1 - \Pr\left(\bigcap_{j=1}^{\frac{D}{2T}} E_{i,j} < \lambda_2 \middle| H_0\right), \tag{13}$$

where $E_{i,j} = \sum_{k=1}^{j} e_{i,k}$ and $e_{i,j}$ is the energy received during the jth sampling interval at sensor i. As $e_{i,j}$ are independent and identically distributed when target is absent, i.e., $e_{i,j} \mid H_0 \sim \mathcal{N}(\mu, \sigma^2)$, all sensors share the same local false alarm rate. The joint probability in (13) can be calculated by Monte Carlo method. The details of the Monte Carlo method can be found in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2009.125. Similar to (11), the system false alarm rate in the second phase can be calculated by

$$P_{F_2} \simeq Q\left(\frac{\frac{N}{2} - N\alpha_2}{\sqrt{N\alpha_2 - N\alpha_2^2}}\right). \tag{14}$$

Similar to (13), the local detection probability of sensor i is

$$\beta_{2,i} = 1 - \Pr\left(\bigcap_{j=1}^{\frac{D}{2T}} E_{i,j} < \lambda_2 \middle| H_1\right). \tag{15}$$

For a static sensor, the energies received in different sampling intervals are independent and identically distributed. However, for a mobile sensor, the energies received in different sampling intervals have different mean values, which depend on the movement schedule. Specifically,

$$e_{i,j} \mid H_1, \text{ static } \sim \mathcal{N}\big(\mu + e_s\big(x_i^0\big), \sigma^2\big), \quad \forall j, \tag{16}$$

$$e_{i,j} \mid H_1, \text{ mobile } \sim \mathcal{N}(\mu + \xi_{i,j}(\mathbf{S}), \sigma^2), \quad \forall j, \tag{17}$$

where $\xi_{i,j}(\mathbf{S})$ is the energy received by mobile sensor i in the jth sampling interval of the second phase under movement schedule $\mathbf{S}$. Suppose L_i is the number of total moves of

sensor i in movement schedule $\mathbf{S}$. As we will prove in Theorem 2, the system detection performance is maximized when the L_i moves are consecutive from the beginning of the second phase. In such a case, $\xi_{i,j}(\mathbf{S})$ can be calculated as

$$\xi_{i,j}(\mathbf{S}) = \begin{cases} \frac{1}{vT}\int_{x_i^0-jvT}^{x_i^0-(j-1)vT} e_s(x_i)\mathrm{d}x_i, & 1 \le j \le L_i; \\ e_s(x_i^0 - L_i vT), & j > L_i. \end{cases} \quad (18)$$

In (18), when mobile sensor i is moving from $x_i^0 - jvT$ to $x_i^0 - (j-1)vT$ during the jth sampling interval, the received energy is the arithmetic average over K samples, which can be approximated by the mean value of the function $e_s(x_i)$ defined by (1); when mobile sensor i stops moving and remains at the position of $x_i^0 - L_i vT$ after L_i sampling intervals, the received energy in one sampling interval is a constant, i.e., $e_s(x_i^0 - L_i vT)$. Similar to (12), the system detection probability in the second phase can be calculated by

$$P_{D_2} \simeq Q\left(\frac{\frac{N}{2} - \sum_{i=1}^N \beta_{2,i}}{\sqrt{\sum_{i=1}^N \beta_{2,i} - \sum_{i=1}^N \beta_{2,i}^2}}\right). \quad (19)$$

Denote l_i as the random variable of the actual moves of mobile sensor i in the second phase. We now derive the expected values of l_i when the target is absent and present, which are denoted by $\mathcal{E}_0^i(L_i)$ and $\mathcal{E}_1^i(L_i)$, respectively.

When the target is absent, mobile sensor i terminates the second phase after k sampling intervals with a probability of

$$\begin{cases} \Pr\left(\bigcap_{j=1}^{k-1} E_{i,j} < \lambda_2 \bigcap E_{i,k} \ge \lambda_2 | H_0\right), & k < \frac{D}{2T}; \\ \Pr\left(\bigcap_{j=1}^{\frac{D}{2T}-1} E_{i,j} < \lambda_2 | H_0\right), & k = \frac{D}{2T}. \end{cases}$$

Accordingly, if the target is absent, the expected number of moves $\mathcal{E}_0^i(L_i)$ is given by

$$\begin{aligned} \mathcal{E}_0^i(L_i) &= \mathbb{E}[l_i|H_0] = 1 \cdot \Pr(E_{i,1} \ge \lambda_2 | H_0) \\ &\quad + \sum_{k=2}^{L_i-1} k \cdot \Pr\left(\bigcap_{j=1}^{k-1} E_{i,j} < \lambda_2 \bigcap E_{i,k} \ge \lambda_2 \middle| H_0\right) \\ &\quad + L_i \cdot \Pr\left(\bigcap_{j=1}^{L_i-1} E_{i,j} < \lambda_2 \middle| H_0\right) \\ &= 1 - \Pr(E_{i,1} < \lambda_2 | H_0) \\ &\quad + \sum_{k=2}^{L_i-1} k \cdot \left(\Pr\left(\bigcap_{j=1}^{k-1} E_{i,j} < \lambda_2 \middle| H_0\right)\right. \\ &\quad \left. - \Pr\left(\bigcap_{j=1}^{k} E_{i,j} < \lambda_2 \middle| H_0\right)\right) \\ &\quad + L_i \cdot \Pr\left(\bigcap_{j=1}^{L_i-1} E_{i,j} < \lambda_2 \middle| H_0\right) \\ &= 1 + \sum_{k=1}^{L_i-1} \Pr\left(\bigcap_{j=1}^{k} E_{i,j} < \lambda_2 \middle| H_0\right). \end{aligned} \quad (20)$$

Similarly, if the target is present, the expected number of moves $\mathcal{E}_1^i(L_i)$ can be derived as

$$\mathcal{E}_1^i(L_i) = \mathbb{E}[l_i \mid H_1] = 1 + \sum_{k=1}^{L_i-1} \Pr\left(\bigcap_{j=1}^{k} E_{i,j} < \lambda_2 \middle| H_1\right). \quad (21)$$

Suppose there are M mobile sensors in the cluster. The total expected numbers of moves, $\mathcal{L}_0(\lambda_2, \mathbf{S})$ and $\mathcal{L}_1(\lambda_2, \mathbf{S})$, which are needed to compute the cost defined by (5), are given by $\mathcal{L}_r(\lambda_2, \mathbf{S}) = \sum_{i=1}^M \mathcal{E}_r^i(L_i)$ for $r = 0, 1$. We note that the joint probabilities in (13), (15), (20), and (21) can be calculated by Monte Carlo method. The details of the Monte Carlo method can be found in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2009.125.

6 Near-Optimal Sensor Movement Scheduling

In this section, we first analyze the structure of the optimal solution in Section 6.1. A dynamic-programming-based near-optimal movement scheduling algorithm is proposed in Section 6.2. The procedure of finding the detection thresholds is described in Section 6.3.

6.1 The Structure of Optimal Solution

A naive method to solve Problem 1 that is formulated in Section 4.1 is to exhaustively search all possible combinations of λ_1, λ_2, and $\mathbf{S}$. Since both λ_1 and λ_2 have q possible values and each mobile sensor has maximum $2^{\frac{D}{2T}}$ possible movement schedules, the complexity is $\mathcal{O}(q^2 \cdot 2^{\frac{D}{2T}M})$. Such an exponential complexity with respect to the number of mobile sensors is not practical. In this section, we analyze the structure of optimal solution to the problem, which leads to the development of a polynomial-time near-optimal movement scheduling algorithm in Section 6.2.

A solution $< \lambda_1, \lambda_2, \mathbf{S} >$ is said to be *valid* if all constraints can be satisfied. In other words, given a movement schedule $\mathbf{S}$, if λ_1 and λ_2 can be found to satisfy the constraints (6) and (7), $< \lambda_1, \lambda_2, \mathbf{S} >$ is a valid solution. A valid solution is *optimal* if it minimizes the cost function (5).

For a movement schedule $\mathbf{X}$, we define $C(\lambda_2, \mathbf{X})$ as the inverse function of P_{D_2} given by (19):

$$C(\lambda_2, \mathbf{X}) = Q^{-1}(P_{D_2}) = \frac{\frac{N}{2} - \sum_{i=1}^N \beta_{2,i}}{\sqrt{\sum_{i=1}^N \beta_{2,i} - \sum_{i=1}^N \beta_{2,i}^2}}. \quad (22)$$

As the local detection probabilities $\beta_{2,i}$ depend on the detection threshold of the second phase as well as the movement schedule, C is a function of λ_2 and $\mathbf{X}$. The following theorem shows a property of the optimal solution:

Theorem 1. *Suppose* $\mathbf{S}$ *and* $\mathbf{S}'$ *are two valid movement schedules. For a certain* λ_2*, if* $\mathcal{L}_0(\lambda_2, \mathbf{S}) = \mathcal{L}_0(\lambda_2, \mathbf{S}')$*,* $\mathcal{L}_1(\lambda_2, \mathbf{S}) = \mathcal{L}_1(\lambda_2, \mathbf{S}')$ *and* $C(\lambda_2, \mathbf{S}) \le C(\lambda_2, \mathbf{S}')$*, there must exist* λ_1 *and* λ_1' *such that* $c(\lambda_1, \lambda_2, \mathbf{S}) \le c(\lambda_1', \lambda_2, \mathbf{S}')$.

Proof. Suppose $< \lambda_1, \lambda_2, \mathbf{S} >$ and $< \lambda_1', \lambda_2, \mathbf{S}' >$ minimize the cost function among all valid solutions with schedules $\mathbf{S}$ and $\mathbf{S}'$ for a certain λ_2, respectively. As $\mathbf{S}$, $\mathbf{S}'$, and λ_2 are known, such solutions can be found by the exhaustive search of values of λ_1 in polynomial time. We construct a new solution $< \lambda_1', \lambda_2, \mathbf{S} >$. We now show

that it is a valid solution. Compared with $< \lambda_1', \lambda_2, \mathbf{S}' >$, this new solution only changes P_{D_2} in all constraints. As P_{D_2} always decreases with $C(\lambda_2, \mathbf{X})$ and $C(\lambda_2, \mathbf{S}) \leq C(\lambda_2, \mathbf{S}')$, we have $P_{D_2}(\lambda_2, \mathbf{S}) \geq P_{D_2}(\lambda_2, \mathbf{S}')$. Therefore, constraint (7) can be met and $< \lambda_1', \lambda_2, \mathbf{S} >$ is a valid solution. Since λ_1 minimizes the cost function among all valid solutions with $\mathbf{S}$, hence,

$$\begin{aligned} c(\lambda_1, \lambda_2, \mathbf{S}) &\leq c(\lambda_1', \lambda_2, \mathbf{S}) \\ &= (1 - P_a) P_{F_1}(\lambda_1') \mathcal{L}_0(\lambda_2, \mathbf{S}) + P_a P_{D_1}(\lambda_1') \mathcal{L}_1(\lambda_2, \mathbf{S}) \\ &= (1 - P_a) P_{F_1}(\lambda_1') \mathcal{L}_0(\lambda_2, \mathbf{S}') + P_a P_{D_1}(\lambda_1') \mathcal{L}_1(\lambda_2, \mathbf{S}') \\ &= c(\lambda_1', \lambda_2, \mathbf{S}'). \end{aligned}$$

□

Theorem 1 shows that for given total expected numbers of moves (i.e., $\mathcal{L}_0$ and $\mathcal{L}_1$) and detection threshold for the second-phase detection (i.e., λ_2), the cost function increases with $C(\lambda_2, \mathbf{X})$. Therefore, the optimal solution must yield the minimum $C(\lambda_2, \mathbf{X})$ among all solutions that have the same total expected numbers of moves. Moreover, the maximum total expected numbers of moves for all mobile sensors are bounded. Therefore, for given $\mathcal{L}_0$, $\mathcal{L}_1$, and λ_2, if there only exists a polynomial number of valid movement schedules, the optimal schedule can be found as the one that minimizes the value of $C(\lambda_2, \mathbf{X})$ in polynomial time. However, according to (20)-(22), λ_2, $\mathbf{X}$, and $\beta_{2,i}$ have a complex nonlinear relationship, which suggests that there may exist an exponential number of movement schedules for given $\mathcal{L}_0$ and $\mathcal{L}_1$. In the following, we describe a linear approximation of $C(\lambda_2, \mathbf{X})$, which is the key to find the near-optimal movement schedule in polynomial time.

Denote $\mathbf{b} = [\beta_{2,1}, \ldots, \beta_{2,N}]^{\mathrm{T}}$ and $\mathbf{1} = [1, \ldots, 1]^{\mathrm{T}}$, we have

$$C(\lambda_2, \mathbf{X}) = f(\mathbf{b}) = \frac{\frac{N}{2} - \mathbf{b}^{\mathrm{T}}\mathbf{1}}{\sqrt{\mathbf{b}^{\mathrm{T}}\mathbf{1} - \mathbf{b}^{\mathrm{T}}\mathbf{b}}}.$$

The first order Taylor expansion of $f(\mathbf{b})$ at $\mathbf{b}_0$ is $f(\mathbf{b}) = f(\mathbf{b}_0) + \nabla f(\mathbf{b}_0)^{\mathrm{T}}(\mathbf{b} - \mathbf{b}_0) + R_1$, where $\nabla f(\mathbf{b}) = [\frac{\partial f}{\partial \beta_{2,1}}, \ldots, \frac{\partial f}{\partial \beta_{2,N}}]^{\mathrm{T}}$, R_1 is the remainder, and

$$\frac{\partial f}{\partial \beta_{2,i}} = -\frac{1 + \frac{1}{2}(\frac{n}{2} - \mathbf{b}^{\mathrm{T}}\mathbf{1})(\mathbf{b}^{\mathrm{T}}\mathbf{1} - \mathbf{b}^{\mathrm{T}}\mathbf{b})^{-1}(1 - 2\beta_{2,i})}{\sqrt{\mathbf{b}^{\mathrm{T}}\mathbf{1} - \mathbf{b}^{\mathrm{T}}\mathbf{b}}}.$$

If we expand $f(\mathbf{b})$ at $\mathbf{b}_0 = \frac{1}{2} \cdot \mathbf{1}$, we have $\frac{\partial f}{\partial \beta_{2,i}}|_{\mathbf{b}_0} = -\frac{2}{\sqrt{N}}$ and the Taylor expansion becomes

$$C(\lambda_2, \mathbf{X}) = -\frac{2}{\sqrt{N}} \sum_{i=1}^{N} \beta_{2,i} + \sqrt{N} + R_1. \tag{23}$$

The order of the remainder term R_1 is $\mathcal{O}(\|\mathbf{b} - \mathbf{b}_0\|^2)$ [40], where $\| \cdot \|$ represents the euclidean distance function.

Equation (23) shows that $C(\lambda_2, \mathbf{X})$ monotonically decreases with $\sum_{i=1}^{N} \beta_{2,i}$ if R_1 is independent of $\mathbf{b}$. Our numerical simulation in Appendix C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2009.125, shows that this monotonicity holds with a high probability (>98%). In practice, $C(\lambda_2, \mathbf{X})$ can be minimized by maximizing $\sum_{i=1}^{N} \beta_{2,i}$. Since the local detection probabilities of static sensors are independent of the movement schedule, if the sum of detection probabilities of all mobile sensors is maximized, $\sum_{i=1}^{N} \beta_{2,i}$ is also maximized.

We now show another property of the optimal solution that further reduces the problem complexity.

Theorem 2. *Suppose mobile sensor i is scheduled with L_i moves in an optimal schedule. In order to maximize the sum of local detection probabilities in the second-phase detection (i.e., $\sum_{i=1}^{N} \beta_{2,i}$), the L_i moves must be consecutive from the beginning of the second phase.*

Proof. Suppose schedule set $\mathcal{S}$ is composed of all schedules in which mobile sensor i is scheduled with arbitrary L_i moves. As the sum of Gaussian variables still follows normal distribution, the sum of energy received by mobile sensor i during j sampling intervals from the beginning of the second phase follows a normal distribution, i.e., $E_{i,j} \sim \mathcal{N}(j\mu + \sum_{k=1}^{j} \xi_{i,k}(\mathbf{S}), j\sigma^2)$ for any schedule $\mathbf{S} \in \mathcal{S}$. Obviously, for each $k = 1, \ldots, \frac{D}{2T}$, if mobile sensor i moves consecutively from the beginning of the second phase, the energy received in the kth sampling interval, i.e., $\xi_{i,k}(\mathbf{S})$ will have the maximum value given by (18). Accordingly, at mobile sensor i, for each $j = 1, \ldots, \frac{D}{2T}$, $E_{i,j}$ has the maximum mean value. Hence, the joint probability $\Pr(\bigcap_{j=1}^{\frac{D}{2T}} E_{i,j} < \lambda_2 \mid H_1)$ is minimized. Therefore, the local detection probability of mobile sensor i, $\beta_{2,i}$, which is given by (15), is maximized. If all mobile sensors move in parallel and consecutively from the beginning of the second-phase detection, the sum of local detection probabilities in the second-phase detection, $\sum_{i=1}^{N} \beta_{2,i}$, is maximized. □

6.2 A Near-Optimal Movement Scheduling Algorithm

Based on the analysis on the structure of optimal solution, we develop the following strategy to solve Problem 1 that is formulated in Section 4.1. First, for given total expected numbers of moves, $\mathcal{L}_0$ and $\mathcal{L}_1$, we employ a dynamic programming algorithm to find the schedule that minimizes $C(\lambda_2, \mathbf{S})$ defined by (22) in polynomial time, which is presented in this section. Then, we search the detection thresholds of two phases λ_1 and λ_2 to find the near-optimal solution in polynomial time, which is presented in Section 6.3. We note that the solution $< \lambda_1, \lambda_2, \mathbf{S} >$ found this way is optimal if $C(\lambda_2, \mathbf{X})$ strictly decreases with $\sum_{i=1}^{N} \beta_{2,i}$ in (23).

Let $P(i, \mathcal{L}_0^i, \mathcal{L}_1^i)$ denote the maximum $\sum_{j=1}^{i} \beta_{2,j}$ for mobile sensors of index from 1 to i with total expected moves no more than $\mathcal{L}_0^i$ and $\mathcal{L}_1^i$ when the target is absent and present, respectively. We have a dynamic programming recursion:

$$\begin{aligned} &P\big(i, \mathcal{L}_0^i, \mathcal{L}_1^i\big) = \\ &\max_{0 \leq L_i \leq H_i} \big\{ P\big(i-1, \mathcal{L}_0^i - \mathcal{E}_0^i(L_i), \mathcal{L}_1^i - \mathcal{E}_1^i(L_i)\big) + \beta_{2,i}(L_i) \big\}, \end{aligned} \tag{24}$$

where H_i is the maximum number of moves of sensor i. Note that $H_i = \min\{\frac{D}{2T}, \frac{x_i^0}{vT}\}$, as the sensor will stop moving if it reaches the surveillance spot or the required delay bound is reached. $\beta_{2,i}(L_i)$ (given by (15)) is the local detection probability of sensor i which is scheduled with consecutive

L_i moves in the second phase. The initial state of the above recursion is $P(0,\cdot,\cdot)=0$.

According to (24), at the ith iteration, the optimal value of $P(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$ is computed as the maximum value of H_i cases which have been calculated in previous iterations of the recursion. Specifically, for the case where sensor i is scheduled with L_i moves, the sum of local detection probabilities can be computed as $P(i-1,\mathcal{L}_0^i-\mathcal{E}_0^i(L_i),\mathcal{L}_1^i-\mathcal{E}_1^i(L_i))+\beta_{2,i}(L_i)$ where the first addend is the maximum sum of local detection probabilities for sensor $1,\ldots,i-1$ given the total expected numbers of moves of $\mathcal{L}_0^i-\mathcal{E}_0^i(L_i)$ and $\mathcal{L}_1^i-\mathcal{E}_1^i(L_i)$. According to Theorem 2, sensor i's moves are consecutive from the beginning of the second phase. Therefore, at most H_i cases need to be considered when computing $P(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$. As there are total M mobile sensors in the detection cluster, the maximum sum of local detection probabilities for these mobile sensors is given by $P(M,\mathcal{L}_0^M,\mathcal{L}_1^M)$.

In order to calculate P_{D_2} using (19), the square sum of all local detection probabilities is also needed. For each $P(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$, let $Q(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$ denote the corresponding square sum of local detection probabilities. The initial value is $Q(0,\cdot,\cdot)=0$. A schedule $\mathbf{S}(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$ is also defined for each $P(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$ and initialized to be empty. $Q(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$ and $\mathbf{S}(i,\mathcal{L}_0^i,\mathcal{L}_1^i)$ are computed accordingly in each iteration as follows:

$$
\begin{aligned}
L_i^* &= \underset{0\le L_i\le H_i}{\operatorname{argmax}} \big\{P\big(i-1,\mathcal{L}_0^i-\mathcal{E}_0^i(L_i),\mathcal{L}_1^i-\mathcal{E}_1^i(L_i)\big)\\
&\quad +\beta_{2,i}(L_i)\big\},\\
Q\big(i,\mathcal{L}_0^i,\mathcal{L}_1^i\big) &= Q\big(i-1,\mathcal{L}_0^i-\mathcal{E}_0^i(L_i^*),\mathcal{L}_1^i-\mathcal{E}_1^i(L_i^*)\big)+\beta_{2,i}^2(L_i^*),\\
\mathbf{S}\big(i,\mathcal{L}_0^i,\mathcal{L}_1^i\big) &= \mathbf{S}\big(i-1,\mathcal{L}_0^i-\mathcal{E}_0^i(L_i^*),\mathcal{L}_1^i-\mathcal{E}_1^i(L_i^*)\big)\\
&\quad \cup\big\{\mathcal{M}_i\big(x_i^0-(j-1)vT,j\big)\mid 1\le j\le L_i^*\big\}.
\end{aligned}
$$

Note that both $\mathcal{L}_0$ and $\mathcal{L}_1$ are the expected numbers of moves, and hence, are real numbers. Their values are discretized in the dynamic programming procedure, i.e., $\mathcal{L}_r=\{0,\Delta,2\Delta,\ldots,\mathcal{L}_{r,\max}\}, r=0,1$, where Δ is the granularity. The maximum value $\mathcal{L}_{r,\max}$ can be set to be $\max_{\lambda_2\in\Lambda_2}\sum_{i=1}^{M}\mathcal{E}_r^i(H_i)$, $r=0,1$, respectively. The complexity of the dynamic programming procedure is $\mathcal{O}((\frac{MD}{\Delta})^2)$.

6.3 Finding Detection Thresholds

This section presents the procedure of finding the two detection thresholds and the movement schedule to construct the near-optimal solution to Problem 1. Once the clusters are formed after deployment, $P(M,\mathcal{L}_0,\mathcal{L}_1)$, $Q(M,\mathcal{L}_0,\mathcal{L}_1)$, and $\mathbf{S}(M,\mathcal{L}_0,\mathcal{L}_1)$ are precomputed for each possible combination of λ_2, $\mathcal{L}_0$, and $\mathcal{L}_1$ using the above movement scheduling algorithm:

$$
\begin{aligned}
&\{P_{\lambda_2}(M,\mathcal{L}_0,\mathcal{L}_1)\mid \lambda_2\in\Lambda_2,\mathcal{L}_0\in[0,\mathcal{L}_{0,\max}],\mathcal{L}_1\in[0,\mathcal{L}_{1,\max}]\},\\
&\{Q_{\lambda_2}(M,\mathcal{L}_0,\mathcal{L}_1)\mid \lambda_2\in\Lambda_2,\mathcal{L}_0\in[0,\mathcal{L}_{0,\max}],\mathcal{L}_1\in[0,\mathcal{L}_{1,\max}]\},\\
&\{\mathbf{S}_{\lambda_2}(M,\mathcal{L}_0,\mathcal{L}_1)\mid \lambda_2\in\Lambda_2,\mathcal{L}_0\in[0,\mathcal{L}_{0,\max}],\mathcal{L}_1\in[0,\mathcal{L}_{1,\max}]\}.
\end{aligned}
$$

Algorithm 1 shows the pseudocode of the solving procedure. For each possible total expected number of moves $\mathcal{L}_0$ and $\mathcal{L}_1$, the values of λ_1 and λ_2 are searched to minimize the cost defined by (5) under the constraints. A zero cost may occur when all constraints are satisfied without moving the sensors toward the surveillance spot (line 12 in Algorithm 1). We note that the algorithm may not find any valid solution when the performance requirements exceed the maximum detection capability of the cluster. For instance, the constraint on the system detection probability may not be satisfied even when all mobile sensors have been scheduled with the maximum number of moves under the delay bound. The complexity of Algorithm 1 is $\mathcal{O}((\frac{MD}{\Delta})^2\cdot q^2)$, which is dependent on the granularities Δ and q. The smaller Δ or greater q yields better solution at the price of higher computation and storage overhead. Therefore, in practice, we can balance the solution quality and the overhead with respect to the capability of cluster head by choosing proper granularities.

Algorithm 1. The procedure of finding the detection thresholds

Input: $\Lambda_1,\Lambda_2,\{P_{\lambda_2}(M,\mathcal{L}_0,\mathcal{L}_1)|\lambda_2\in\Lambda_2,\mathcal{L}_0\in[0,\mathcal{L}_{0,\max}],\mathcal{L}_1\in[0,\mathcal{L}_{1,\max}]\},\{Q_{\lambda_2}(M,\mathcal{L}_0,\mathcal{L}_1)|\lambda_2\in\Lambda_2,\mathcal{L}_0\in[0,\mathcal{L}_{0,\max}],\mathcal{L}_1\in[0,\mathcal{L}_{1,\max}]\},\{\mathbf{S}_{\lambda_2}(M,\mathcal{L}_0,\mathcal{L}_1)|\lambda_2\in\Lambda_2,\mathcal{L}_0\in[0,\mathcal{L}_{0,\max}],\mathcal{L}_1\in[0,\mathcal{L}_{1,\max}]\}$

Output: near-optimal solution $<\lambda_1^*,\lambda_2^*,\mathbf{S}^*>$

```
1: cost = +∞
2: for L0 = [0, Δ, 2Δ, ..., L0,max] do
3:   for L1 = [0, Δ, 2Δ, ..., L1,max] do
4:     for λ1 = [λ1(1), λ1(2), ..., λ1(q)] do
5:       compute P_F1 and P_D1 using (11) and (12)
6:       for λ2 = [λ2(1), λ2(2), ..., λ2(q)] do
7:         compute P_F2 using (14)
8:         if (6) holds then
9:           compute P_D2 using P_λ2(M,L0,L1), Q_λ2(M,L0,L1)
             according to (19)
10:          if (7) holds then
11:            compute current cost c using (5)
12:            if c = 0 then
13:              exit
14:            else if c < cost then
15:              cost = c, λ1* = λ1, λ2* = λ2, S* = S_λ2(M,L0,L1)
16:            end if
17:          end if
18:        end for
19:      end for
20:    end for
21:  end for
22: end for
```

7 Extensions

We now discuss several open issues that have not been addressed in previous sections and investigate their impacts on the performance of the proposed algorithm. We also discuss how to extend our approach to address them.

7.1 Estimating Target Appearance Probability

We assume that the probability that a target appears at the surveillance spot is P_a in the problem formulation. From the structure of the problem formulation, P_a only affects the cost function (5). Hence, when the estimated P_a is inaccurate, the solution found can still satisfy the

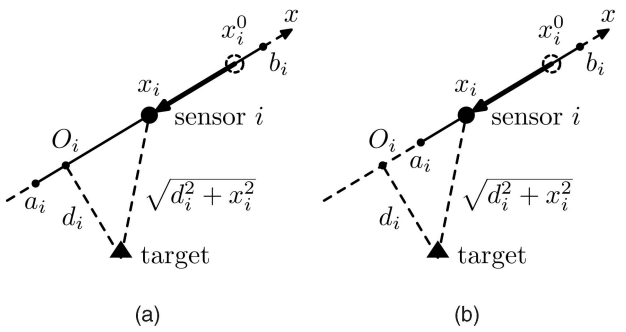


Fig. 4. Two cases of the straight trail mobility model. (a) Case 1: $a_i \leq 0$. (b) Case 2: $a_i > 0$.

constraints (6)-(10) at the expense of longer movement distance of mobile sensors. In other words, inaccurate P_a still yields *valid* solution.

We now discuss how to estimate P_a when it is not known a priori. As many monitored physical phenomena (e.g., moving vehicle) are often spatially correlated, one approach is to obtain the P_a from a closely located neighbor cluster which has accurate P_a. Another approach is to estimate P_a based on detection history. The basic idea is as follows: The initial P_a can be set to be the best guess. As inaccurate P_a still yields valid solution, i.e., any target can be detected with a probability of β, while the false alarm rate is α, after n detections, the number of positive final decisions (denoted by n^+) is $n^+ \simeq \beta p + \alpha(n-p)$, where p is the true number of target appearances which is unknown. Note that βp is the number of correct detections, and $\alpha(n-p)$ is the number of false alarms. Therefore, the P_a can be estimated as $P_a = \frac{p}{n} \simeq \frac{n^+ - n\alpha}{n(\beta-\alpha)}$. We note that this method can also be applied to update P_a periodically so that the detection cluster can adapt to the change of target appearance probability.

7.2 Incorporating Different Sensor Mobility Models

In previous sections, we assume the free mobility model for mobile sensors, i.e., the mobile sensors are able to move continuously in any direction. We now discuss how our approach can be applied to other mobility models.

In the *hop-based mobility model*, a sensor's movement is restricted to hops powered by fuel propellers. Several sensor systems are based on this mobility model [41]. Our approach can be easily extended to such a model by setting the unit of sensor moving distance to be the hop distance. In another mobility model, sensors move along fixed trails in order to improve the reliability of the system. For instance, the climate sensors deployed in James Reserve only move along fixed cables between trees [4], [15]. In such a case, we can find the partial path on a sensor's trail that leads closer to a surveillance spot and apply the movement scheduling algorithm only on the partial path.

We now discuss a typical case of the trail mobility model, which is referred to as the *straight trail mobility model*. In this mobility model, each mobile sensor can move along a finite straight trail, as illustrated in Fig. 4. For sensor i, we set up a coordinate system with the trail as the x-axis and the foot of the perpendicular across the surveillance spot as the origin O_i. Without loss of generality, we assume that sensor i can only move in $[a_i, b_i]$ and $b_i > 0$. Accordingly, there are two possible cases, i.e., $a_i \leq 0$ and $a_i > 0$. Denote x_i^0 as the original position of sensor i and d_i as the shortest distance of the surveillance spot to the x-axis. As shown in Fig. 4, when $a_i \leq 0$, the partial path is $[O_i, x_i^0]$ and the sensor can move from x_i^0 to O_i because it achieves the highest Signal-to-Noise Ratio at O_i. In the case of $a_i > 0$, the partial path is $[a_i, x_i^0]$. Therefore, the spatial and temporal constraint of sensor movement under the straight trail mobility model is

$$\forall \mathcal{M}_i(x,j) \in \mathbf{S}, \left(vT + \max\{0, a_i\} \leq x \leq x_i^0\right) \wedge \left(1 \leq j \leq \frac{D}{2T}\right).$$

That is, the position of sensor i at the beginning of every move lies within $[vT + \max\{0, a_i\}, x_i^0]$. The above equation corresponds to the constraint (10) under the free mobility model. Under the coordinate system of sensor i, the distance between the target and sensor i is $\sqrt{d_i^2 + x_i^2}$, where x_i is the position of sensor i. Accordingly, under the coordinate system of sensor i, the attenuated signal energy at the position of sensor i is given by $e_s'(x_i \mid d_i) = e_s(\sqrt{x_i^2 + d_i^2})$, where $e_s(\cdot)$ is given by (1). Note that all sensors share the same formula (given by (1)) to compute the attenuated signal energy under the free mobility model. However, under the straight trail mobility model, different mobile sensors have different formulas defined in their own coordinate systems. Nevertheless, the structure of the optimal solution does not change. Specifically, by replacing $e_s(x_i)$ in (16) and (18) with $e_s'(x_i \mid d_i)$, the near-optimal movement scheduling algorithm also works under the straight trail mobility model.

7.3 Handling Heterogeneous Sensors

In this section, we discuss the basic idea of extending our approach to the networks that are composed of several types of sensors (e.g., acoustic and seismic sensors). Suppose there are total m types of sensors and each sensing modality has a specific measurement model as stated in Section 3.1. We summarize the major extensions to our approach as follows:

1. As different sensing modalities should have different detection thresholds, the solution of our problem becomes $< \vec{\lambda_1}, \vec{\lambda_2}, \mathbf{S} >$, where $\vec{\lambda_k} = [\lambda_{k,1}, \ldots, \lambda_{k,m}]^{\mathrm{T}}$ and $\lambda_{k,j}$ is the detection threshold for modality j in the kth phase.
2. In the absence of target, the measurements of different modalities are not identically distributed. However, as a number of sensors take part in the decision fusion, the sum of positive decisions $Y|H_0$ approximately follows the normal distribution according to the Lyapunov's Central Limit Theorem. As a result, P_{F_1} and P_{F_2} have similar expressions as (12) and (19). Specifically, P_{F_1} and P_{F_2} can be obtained by replacing $\beta_{1,i}$ in (12) with $\alpha_{1,i}$ and $\beta_{2,i}$ in (19) with $\alpha_{2,i}$, respectively, where $\alpha_{k,i}$ is the local false alarm rate of sensor i in the kth phase. Hence, the system false alarm rate and detection probability can be computed with closed-form expressions.
3. Theorems 1 and 2 still hold after replacing λ_1 and λ_2 by $\vec{\lambda_1}$ and $\vec{\lambda_2}$, respectively, as these two theorems do

not depend on the algebraic representation (scalar or vector) of the detection thresholds.

The complexity of Algorithm 1 will increase to $\mathcal{O}((\frac{MD}{\Delta})^2 q^{2m})$, as iterating each $\vec{\lambda}_1$ and $\vec{\lambda}_2$ has a complexity of $\mathcal{O}(q^{2m})$. Note that the decisions of heterogeneous sensors are fused according to the majority rule. More efficient fusion rules that account for the difference in fidelity characteristics of sensors might be incorporated into our approach, e.g., weighted decision fusion. However, we omit the investigation of the optimal fusion rules for heterogeneous sensors due to space limit.

7.4 Detecting Moving Targets

In previous sections, we assume that the target remains stationary at the surveillance spot. In many applications, the target is mobile. In this section, we briefly discuss how to extend our approach to address the problem of detecting moving targets. Note that the approach presented below can also be extended to the more general case where the target is spatially distributed, as long as the spatial distribution of the target can be estimated.

We face several challenges in detecting moving targets. First, the accurate position of the moving target is often unknown in practice. Moreover, the signal attenuation characteristic of the moving target varies over time. Therefore, it is difficult to find the optimal solution that achieves the specified detection performance requirement. Our basic idea to address this issue is to treat the moving target as a stationary target with conservative source energy estimate. For a cluster, we consider the performance of detecting the moving target with source energy of S_0 in a region A that is around the surveillance spot. We assume the time that the target is in A is longer than the required detection delay D. Denote $d_{i,\max}$ as the maximum distance from sensor i to any point in A. Hence, the minimum energy received by sensor i when the target is in A, denoted by $s_{i,\min}$, is $s_{i,\min} = S_0 w(d_{i,\max})$. If a stationary target with source energy of S_0' is at the surveillance spot, the signal energy received by sensor i is $s_i' = S_0' w(x_i^0)$. If we let $S_0' = \min_i\{\frac{S_0 w(d_{i,\max})}{w(x_i^0)}\}$, we have $s_{i,\min} \geq s_i'$ for any sensor i in the cluster. As a result, the detection performance for the moving target with source energy of S_0 will not be worse than that for the stationary target at the surveillance spot with source energy of S_0'. Therefore, the solution found by our algorithm with target source energy of S_0' is a *valid* solution for detecting the moving target with source energy of S_0. Note that the region A can be chosen according to the mobility model of the target. For instance, if the target path is known, the region A can be reduced to the path section around the surveillance spot. The above scheme will achieve better performance when it is integrated with a target tracking protocol that can determine the trajectory of a moving target.

8 Performance Evaluation

8.1 Simulation Methodology and Settings

We conduct extensive simulations using the real data traces collected in the Sensor Information Technology vehicle detection experiments [7]. In the experiment, 75 sensors are deployed to detect military vehicles driving through several intersected roads. The data set used in our simulations includes the acoustic time series recorded by 23 nodes at the frequency of 4,960 Hz and ground truth. Received energy is calculated every 0.75 s. Each run is named after the vehicle type and the number of run, e.g., AAV3 stands for the third run when an Assault Amphibian Vehicle (AAV) drives through the road. We refer to [7] for more detailed setup of the experiment. In our simulations, the acoustic data of AAV3-AAV11 are used.

As the data are collected by fixed sensors, they cannot be directly used in our simulations. We generate data for our simulations as follows: For each energy measurement collected by a sensor, we compute the distance between the sensor and the vehicle from the ground truth data. When a sensor makes a measurement in our simulations, the energy is set to be the real measurement gathered at a similar distance to target.

While the sensor measurements are directly taken from real data traces, we use a sensor measurement model estimated from a training data set in our movement scheduling algorithm. Such a methodology accounts for several realistic factors. First, there exists considerable deviation between the measurements of sensors in our simulations and the training data. This deviation is due to various reasons including sound reverberation, the difference between vehicles, and the changing noise levels caused by wind. Moreover, our movement scheduling algorithm assumes that targets remain stationary at each surveillance spot before disappearance. However, vehicles in several runs of our simulations drive along a road. As a result, the actual Signal-to-Noise Ratios 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 existing target tracking algorithms [1], [2].

The simulation code is written in C++. As in [31], we estimate the sensor measurement model using the AAV3 track as the training data set. Our estimated parameters in (1) and (2) are: $S_0 = 0.51$ (after normalization), $d_0 = 2.6$ m, $k = 2$, $\mu = 10^{-4}$, and $\sigma^2 = 2\mu$. In our simulations, sensors are randomly distributed in a field of $50 \times 50\ \mathrm{m}^2$ surrounded by four road sections, and the length of each road section is 75 m, as illustrated in Fig. 5. A surveillance spot is located at the center of each road section. In real scenarios, surveillance spots would be identified by the network after the deployment.

The total simulation time is 3×10^7 seconds, and each target appearance lasts for 15 seconds. The probability that the target appears at the beginning of a sampling interval is set to be 5 percent. Each sensor in the deployment region belongs to four clusters. A sensor is randomly selected as the cluster head for each surveillance spot. Based on the local decisions of all sensors in the first-phase detection, a cluster head can determine if a possible target appears at the

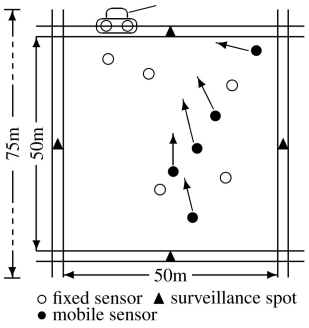


Fig. 5. The deployment region of sensors.

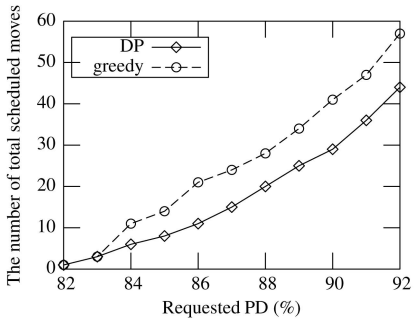


Fig. 7. The number of total scheduled moves *versus* requested PD.

surveillance spot it monitors. The maximum false alarm rate α is set to be 5 percent, except in Fig. 6. The moving speed of mobile sensors is set to be 1 m/s except in Section 8.3. Sampling interval T is set to be 0.75 s, which is consistent with the setting in the Sensor Information Technology experiments [7]. The search granularity in the near-optimal movement scheduling algorithm Δ is set to be 0.1.

The expected detection delay D is set to be 15 s. We assume that a target remains stationary at a surveillance spot for 15 seconds before it disappears in the first several sets of simulations. In Section 8.5, we evaluate the performance of our algorithm when vehicles drive along the roads at 5 m/s, which is the average speed of the vehicles in the Sensor Information Technology experiments. It takes a vehicle 15 s to pass a road section, as shown in Fig. 5.

8.2 System Detection Performance

Our first set of simulations evaluates the basic performance of the mobility-assisted detection model and the effectiveness of our movement scheduling algorithm.

Fig. 6 shows the Receiver Operating Characteristic (ROC) curves for different numbers of mobile sensors. Under each false alarm rate bound, the movement schedule of mobile sensors is computed to maximize the system detection probability. Total 12 sensors are deployed. In the figure, *static* refers to the deployment in which all sensors remain stationary, *1/4 mobile* refers to three mobile sensors and nine static sensors, and so on. We can see that the system detection performance increases significantly with the number of mobile sensors. In particular, six mobile sensors can improve the detection performance by 10-35 percent.

In the second set of simulations, we evaluate the effectiveness of our dynamic programming (DP)-based movement scheduling algorithm. Total 10 mobile sensors are deployed. We employ a *greedy* scheduling algorithm as the baseline, in which the cluster head always chooses the mobile sensor closest to the surveillance spot and schedules it with one move until the required detection performance is achieved. Fig. 7 shows the total number of moves in the schedules found by different algorithms when the requested detection probability varies from 82 to 92 percent. As shown in Fig. 7, our algorithm schedules about 10 fewer moves than the greedy algorithm.

8.3 Impact of Mobile Sensor Speed

In this set of simulations, we evaluate the impact of mobile sensor speed on the system detection performance. Total 10 mobile sensors are deployed.

Fig. 8 plots the actual detection probability versus the requested detection probability if mobile sensor speed changes from 0.2 to 1.0 m/s. For each mobile sensor speed, the achievable detection probability yields a saturation point, which occurs when all mobile sensors have moved the maximum distance within the detection delay bound. However, even when the mobile sensors move as low as 0.2 m/s, a detection probability of 86 percent is achieved. When the mobile sensor speed is higher, the detection probability increases considerably. This result shows that our movement scheduling algorithm can effectively improve system detection performance by taking advantage of the increase of speed.

Fig. 9 plots the number of moves versus the requested detection probability. For each speed, the number of moves increases with the requested detection probability. Moreover, for a certain requested detection probability, the total number of moves decreases with sensor speed. It shows that

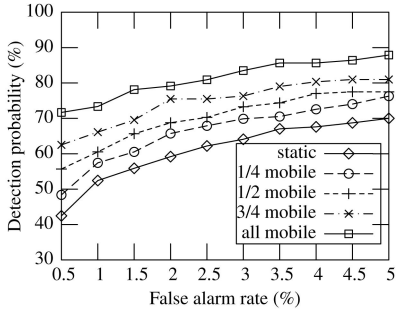


Fig. 6. Receiver operating characteristic.

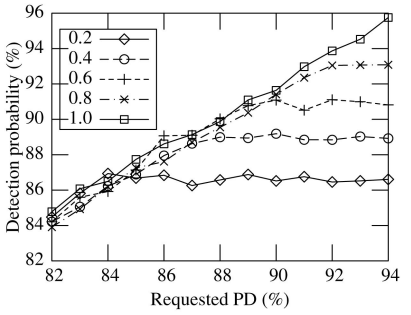


Fig. 8. Actual PD *versus* requested PD with different sensor speeds.

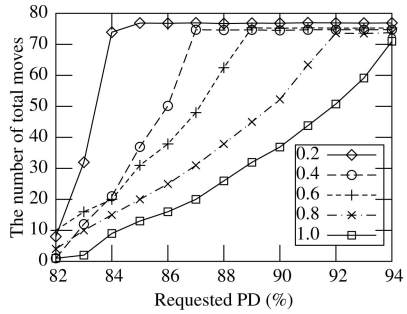


Fig. 9. The number of total moves *versus* requested PD with different sensor speeds.

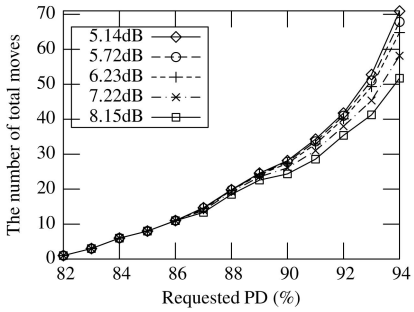


Fig. 11. The number of total moves *versus* requested PD with different Signal-to-Noise Ratio.

our movement scheduling algorithm can mitigate the impact of low sensor speed on detection performance by scheduling more sensors to move longer distances toward the target.

8.4 Impact of Inaccurate Signal-to-Noise Ratios

In this set of simulations, we evaluate the impact of inaccurate Signal-to-Noise Ratios on the system detection performance. Total 10 mobile sensors are deployed. The movement scheduling algorithm uses an average peak Signal-to-Noise Ratio of $6.23\,\mathrm{dB}$, while the actual signal energy received by each sensor is amplified or suppressed by certain ratio. Such a setting evaluates the performance of our algorithm when the estimated target energy model is not accurate.

Fig. 10 plots the achieved detection probability versus the requested detection probability with different Signal-to-Noise Ratios. The label of each curve indicates the actual Signal-to-Noise Ratio value. We can see that the detection probability meets the requested value when the estimated Signal-to-Noise Ratio is accurate (the curve of "$6.23\,\mathrm{dB}$"). When the actual Signal-to-Noise Ratio is higher than $6.23\,\mathrm{dB}$, the detection probability is higher than the requested value. For instance, the detection probability is almost one when the actual Signal-to-Noise Ratio is $8.15\,\mathrm{dB}$. When the estimated Signal-to-Noise Ratio is lower than the actual value, the detection probability decreases as expected. When the actual Signal-to-Noise Ratio is lower than the estimation by more than $1\,\mathrm{dB}$ (the curve of "$5.14\,\mathrm{dB}$"), the achieved detection probability is lower than the requested value by only about 5 percent.

Fig. 11 plots the number of total moves in the presence of target versus the requested detection probability. In this figure, we can see that more sensor moves are needed to meet a higher requested detection probability. When the Signal-to-Noise Ratio is higher than the estimated value, the number of sensor moves can be reduced because mobile sensors can terminate their second-phase detection in advance once targets are detected. Fig. 11 also shows that the impact of inaccurate Signal-to-Noise Ratios is not obvious when the requested detection probability is smaller than 90 percent.

8.5 Moving Target Detection

In this set of simulations, we evaluate the impact of target's movement and Signal-to-Noise Ratio on the detection performance. The target speed is $5\,\mathrm{m/s}$. We note that our algorithm does not explicitly take into account the mobility of targets. In particular, the movement scheduling algorithm assumes that a target remains stationary at the center of a road section (as shown in Fig. 5) before it disappears. Therefore, when a vehicle drives along the road section, the actual Signal-to-Noise Ratio is lower than the one used in movement scheduling, which results in lower detection probabilities. Our algorithm would yield a better performance when the mobility of targets is explicitly considered, e.g., by integrating with existing object tracking algorithms [1], [2].

Fig. 12 plots the probabilities of correctly detecting moving AAV3 by static, hybrid, and mobile networks when the number of sensors varies from 4 to 18. We can see that the system detection probability significantly increases with the total number of sensors. And obviously, detection performance is increased by introducing reactive mobility. Fig. 13 plots the corresponding average detection delay of the second-phase detection in percentage (100 percent represents the delay time of $\frac{D}{2}$). We can see that the shorter detection delay can be achieved by deploying more mobile

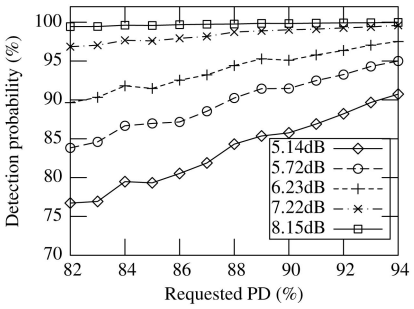


Fig. 10. Actual PD *versus* requested PD with different Signal-to-Noise Ratios.

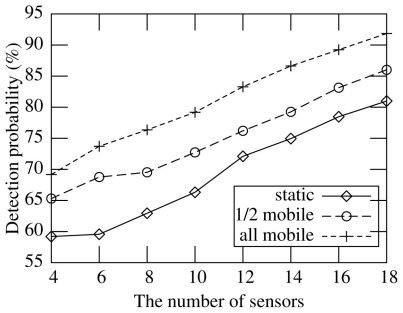


Fig. 12. PD *versus* the number of sensors when detecting moving targets.

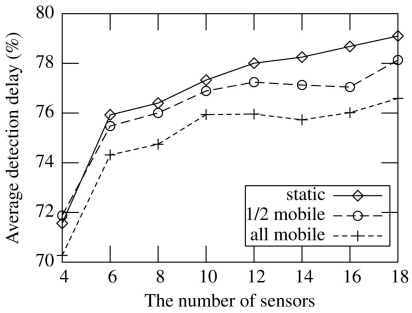


Fig. 13. Detection delay *versus* the number of sensors.

sensors. We also observe that the detection delay does not change a lot when the total number of sensors exceeds 10.

In practice, the energy model of targets may not be accurately estimated. In this set of simulations, the movement schedule of sensors is computed based on the energy model estimated from the data in AAV3 run, while different runs are used for detection. Total 10 mobile sensors are deployed, and the requested detection probability is 92 percent. In Figs. 14 and 15, the AAV runs are ordered by average peak Signal-to-Noise Ratio among all sensors. Fig. 14 shows that the AAV runs with higher Signal-to-Noise Ratios are detected with higher probabilities. Fig. 15 shows that significantly fewer number of moves are used when targets have higher Signal-to-Noise Ratios. This is because mobile sensors terminate their second-phase detection in advance after a positive local decision is made in the sequential detection. This result demonstrates that our algorithm can adaptively reduce cost (i.e., the moving distance) in the presence of higher Signal-to-Noise Ratios because of the local movement control and decision making.

9 Conclusion

This paper exploits reactive mobility to improve the detection performance of wireless sensor networks. We propose a two-phase detection approach in which mobile sensors collaborate with static sensors and move reactively to achieve the required detection performance. We develop a near-optimal sensor movement scheduling algorithm that minimizes the expected moving distance of mobile sensors. Our extensive simulations based on real data traces show that a small number of mobile sensors can significantly improve the system detection performance. Moreover, our movement scheduling algorithm achieves satisfactory performance under a range of realistic settings including slow speed of sensors (as low as $0.2\,\mathrm{m/s}$) and inaccurate

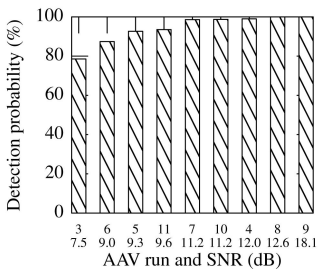


Fig. 14. PD *versus* Signal-to-Noise Ratio.

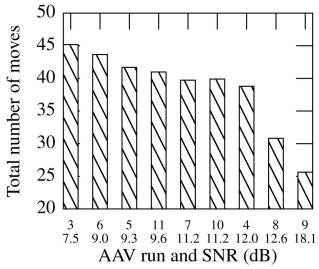


Fig. 15. The number of total moves *versus* Signal-to-Noise Ratio.

knowledge about the target. Our approach is mainly focused on stationary target detection at fixed surveillance spots. In this paper, we have outlined several possible extensions to our approach for more general cases such as detecting moving targets. In our future work, we plan to develop complete solutions for these general cases and evaluate the proposed algorithm through the experiments on real sensor network systems.

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 and Project No. CityU 121107).

References

[1] 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.

[2] F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration for Tracking Applications," *IEEE Signal Processing Magazine,* vol. 19, no. 2, pp. 61-72, Mar. 2002.

[3] 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. MobiSys,* 2004.

[4] R. Pon, M.A. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W.J. Kaiser, M. Srivastava, G. Sukhatme, and D. Estrin, "Networked Infomechanical Systems: A Mobile Embedded Networked Sensor Platform," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* 2005.

[5] 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.

[6] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G.S. Sukhatme, "Robomote: Enabling Mobility in Sensor Networks," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* 2005.

[7] 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, 2004.

[8] 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.

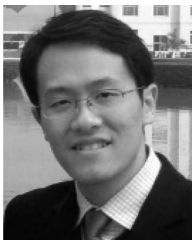
[9] S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying Wireless Sensor Networks under Limited Mobility Constraints," *IEEE Trans. Mobile Computing,* vol. 6, no. 10, pp. 1142-1157, Oct. 2007.

[10] W. Wang, V. Srinivasan, and K.-C. Chua, "Trade Offs between Mobility and Density for Coverage in Wireless Sensor Networks," *Proc. ACM MobiCom,* 2007.

[11] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility Improves Coverage of Sensor Networks," *Proc. ACM MobiHoc,* 2005.

[12] T.-L. Chin, P. Ramanathan, and K.K. Saluja, "Analytic Modeling of Detection Latency in Mobile Sensor Networks," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '06),* 2006.

[13] N. Bisnik, A. Abouzeid, and V. Isler, "Stochastic Event Capture Using Mobile Sensors Subject to a Quality Metric," *Proc. ACM MobiCom,* 2006.
[14] T.-L. Chin, P. Ramanathan, K.K. Saluja, and K.-C. Wang, "Exposure for Collaborative Detection Using Mobile Sensor Networks," *Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems (MASS '05),* 2005.
[15] 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. Conf. Embedded Networked Sensor Systems (SenSys '04),* 2004.
[16] 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 '05),* 2005.
[17] G. Xing, J. Wang, K. Shen, Q. Huang, X. Jia, and H. So, "Mobility-Assisted Spatiotemporal Detection in Wireless Sensor Networks," *Proc. Int'l Conf. Distributed Computing Systems (ICDCS '08),* 2008.
[18] P. Varshney, *Distributed Detection and Data Fusion.* Springer-Verlag, 1996.
[19] S. Dhillon, K. Chakrabarty, and S.S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise," *Proc. Int'l Conf. Information Fusion (FUSION),* 2002.
[20] 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 '08),* 2008.
[21] 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.
[22] J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '03),* 2003.
[23] 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.
[24] 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.
[25] 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.
[26] H. Cao, E. Ertin, V. Kulathumani, M. Sridharan, and A. Arora, "Differential Games in Large-Scale Sensor-Actuator Networks," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '06),* 2006.
[27] J.-C. Chin, Y. Dong, W.-K. Hon, and D. Yau, "On Intelligent Mobile Target Detection in a Mobile Sensor Network," *Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems (MASS '07),* 2007.
[28] J. Latombe, *Robot Motion Planning.* Kluwer Academic Publishers, 1991.
[29] Y. Hwang and N. Ahuja, "Gross Motion Planning: A Survey," *ACM Computing Surveys,* vol. 24, no. 3, pp. 219-292, 1992.
[30] H. Choset, "Coverage for Robotics—A Survey of Recent Results," *Annals of Math. and Artificial Intelligence,* vol. 31, no. 1, pp. 113-126, 2001.
[31] X. Sheng and Y.-H. Hu, "Maximum Likelihood Multiple-Source Localization Using Acoustic Energy Measurements with Wireless Sensor Networks," *IEEE Trans. Signal Processing,* vol. 53, no. 1, pp. 44-53, Jan. 2005.
[32] R. Niu, P.K. Varshney, M. Moore, and D. Klamer, "Decision Fusion in a Wireless Sensor Network with a Large Number of Sensors," *Proc. Int'l Conf. Information Fusion (FUSION),* 2004.
[33] 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.
[34] D. Li and Y.H. Hu, "Energy Based Collaborative Source Localization Using Acoustic Micro-Sensor Array," *J. EUROSIP Applied Signal Processing,* vol. 2003, pp. 321-337, 2003.
[35] 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.
[36] H. Liu, X. Jia, P. Wan, C. Yi, S. Makki, and N. Pissinou, "Maximizing Lifetime of Sensor Surveillance Systems," *IEEE/ACM Trans. Networking,* vol. 15, no. 2, pp. 334-345, Apr. 2007.
[37] M. Cardei, M.T. Thai, Y. Li, and W. Wu, "Energy-Efficient Target Coverage in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* 2005.
[38] I. Olkin, L.J. Gleser, and C. Derman, *Probability Models and Applications.* Macmillan Publishing, 1980.
[39] R.B. Ash and C.A. Doléans-Dade, *Probability & Measure Theory,* second ed. A Harcourt Science and Technology Company, 1999.
[40] J.E. Marsden and A.J. Tromba, *Vector Calculus.* W.H. Freeman Company, 1996.
[41] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility Limited Flip-Based Sensor Networks Deployment," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 2, pp. 199-211, Feb. 2007.



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

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 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 cyber-physical 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.

Hing Cheung So received the BEng degree from City University of Hong Kong in 1990, and the PhD degree from The Chinese University of Hong Kong in 1995, both in electronic engineering. 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.