Mobility-assisted Spatiotemporal Detection in Wireless Sensor Networks

Guoliang Xing*; Jianping Wang*; Ke Shen*; Qingfeng Huang†; Xiaohua Jia*; Hing Cheung So‡
*Department of Computer Science, City University of Hong Kong
†Palo Alto Research Center (PARC) Inc.;
‡Department of Electronic Engineering, City University of Hong Kong

Abstract

Wireless sensor networks (WSNs) deployed for mission-critical applications face the fundamental challenge of meeting stringent spatiotemporal performance requirements using nodes with limited sensing capacity. Although advance network planning and dense node deployment may initially achieve the required performance, they often fail to adapt to the unpredictability and variability of physical reality. This paper explores efficient use of mobile sensors to address limitations of static WSNs for target detection. We propose a data fusion based detection model that enables static and mobile sensors to effectively collaborate in target detection. An optimal sensor movement scheduling algorithm is developed to minimize the total moving distance of sensors while achieving a set of spatiotemporal performance requirements including high detection probability, low system false alarm rate and bounded detection delay. The effectiveness of our approach is validated by extensive simulations based on real data traces collected by 23 sensor nodes in the SensIT experiments [14].

1 Introduction

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

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

However, several challenges must be addressed in order to take advantage of the mobility of WSNs in target detection. First, due to the higher design complexity and manufacturing cost, the number of mobile nodes available in a network is often limited. Therefore, mobile sensors must effectively collaborate with static sensors to achieve the maximum utility. Second, mobile sensors are only capable of low-speed and short-distance movement in practice due to the high power consumption of locomotion. For instance, the typical speed of several mobile sensor platforms (e.g., Packbot [21], Robomote [11], and XYZ [19]) is only 0.5 ~ 2 m/s. A XYZ mobile sensor powered by two AA batteries can only move about 165 meters [19] before the depletion of batteries. Therefore, the movement of mobile sensors must be efficiently scheduled in order to maximize the amount of target information gathered within a short moving distance.

In this paper, we propose a data-fusion centric target detection model that features effective collaboration between static and mobile sensors. We derive an optimal sensor movement scheduling algorithm that minimizes the total moving distance of sensors under a set of spatiotemporal
performance requirements including (1) bounded detection delay, (2) high target detection probability, and (3) low system false alarm rate. Furthermore, we conduct extensive simulations based on real data traces collected by 23 sensors in the SensIT vehicle detection and classification experiments [14]. Our results show that a small number of mobile sensors can significantly boost the detection performance of a network. Moreover, our algorithm can achieve satisfactory performance in a range of realistic scenarios with single/multiple moving vehicles and high variations in the speed of mobile sensors.

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

2 Related Work

Recent work demonstrated that the sensing performance of WSNs can be improved by integrating mobility. Several projects proposed to eliminate coverage holes in a sensing field by relocating mobile sensors [25, 4, 22, 4]. Although such an approach improves the sensing coverage of a network deployment, it does not dynamically improve the network’s performance after targets of interest appear. Complementary to these projects, we focus on online sensor collaboration and movement scheduling strategies that are used after the appearance of targets.

In our recent work [23], we proposed a decision fusion based detection model in which each mobile sensor makes its own detection decision and locally controls its movement. In this paper, we adopt a value fusion based detection model that significantly simplifies the task of mobile sensors. Specifically, each mobile sensor in a detection process is only required to move a certain distance and send its measurements to its cluster head. Such a model is more suitable for mobile sensors with limited capability of signal processing and motion control. In contrast, a mobile sensor in the algorithm proposed in [23] must be able to locally detect targets and adaptively control their movement. Moreover, this paper studies several important issues that are not addressed in [23] including optimal movement scheduling and multi-target detection.

Several recent studies [17, 6] analyzed the impact of mobility on detection delay and area coverage. These studies are based on random mobility model and do not address the issue of actively controlling the movement of sensors. Bisnik et al. [2] analyzed the performance of detecting stochastic events using mobile sensors. Chin et al. [7] proposed to improve the coverage of a region by patrolling static routes using mobile sensors. Different from Chin’s work, we study efficient sensor collaboration and movement scheduling strategies that achieve specified target detection performance. Mobile sensors that can move reactively are used in a networked robotic sensor architecture [1, 20] to improve the sampling density over a region. However, they did not focus on target detection under spatiotemporal performance constraints.

Collaborative target detection in stationary sensor networks has been extensively studied [24, 10, 3, 16, 9, 13]. Several recent projects also studied network deployment strategies that can achieve specified detection performance [8, 12]. Practical network protocols that facilitate target detection and tracking have also been investigated [15, 18, 5]. Complementary to these studies that deal with the mobility of targets, we focus on improving target detection performance by utilizing the mobility of sensors.

3 Preliminaries

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

3.1 Target and Sensing Model

Sensors detect targets by measuring the energy of signals emitted by targets. We assume a signal model that is widely adopted in the literature [24, 14]. Suppose a target is at location \( u \) and emits a signal of power \( W \). The signal power decays as a function of the distance from the target. If the decay factor is \( k \), the signal power measured by a sensor is given by:

\[
W(d) = \begin{cases} \frac{W}{(d/d_0)^k} & \text{if } d > d_0 \\ W & \text{if } d \leq d_0 \end{cases}
\]  

(1)

where \( d \) is the distance between the sensor and the target, \( d_0 \) is a constant determined by the size of the target and the sensor. \( k \) is typically between 2 to 5. Measurements at a sensor are corrupted by noise modeled as the Gaussian distribution with zero-mean. Let \( N^2(T) \) denote the noise energy measured by sensor \( i \) during \( T \). Suppose sensor \( i \) is \( x_i \) away from a target. The total energy it measures during \( T \) is:

\[
U_i(T) = W(x_i) \cdot T + N^2(T)
\]  

(2)

In practice, the parameters of sensing and noise models are often estimated using a training data set before deployment or during initialization phase of the network.
3.2 Multi-sensor Fusion Model

We assume that the network is organized into clusters. Sensors send their energy measurements to the cluster head, which in turn compares the average of all measurements to a threshold $\eta$. If the average is greater than $\eta$, the cluster head decides that a target is present. Otherwise, it decides there is no target. $\eta$ is referred to as the detection threshold hereafter. Such a value-based data fusion model has been studied in the literature [9].

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

$$P_{F} = P(\frac{1}{n} \sum_{i=1}^{n} N_{i}^{2}(T) > \eta)$$

$$= 1 - P(\sum_{i=1}^{n} N_{i}^{2}(T) \leq n\eta)$$

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

$$P_{F} = 1 - \chi_{n}(n\eta)$$

(4)

The probability of detecting a target is

$$P_{D} = P(\frac{1}{n} \sum_{i=1}^{n} (W(x_{i}) \cdot T + N_{i}^{2}(T)) > \eta)$$

$$= P(\sum_{i=1}^{n} N_{i}^{2}(T) > n\eta - \sum_{i=1}^{n} W(x_{i}) \cdot T)$$

$$= 1 - \chi_{n}(n\eta - \sum_{i=1}^{n} W(x_{i}) \cdot T)$$

(5)

where $x_{i}$ is the distance between sensor $i$ and the target.

4 The Mobility-assisted Spatiotemporal Detection Problem

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

4.1 Overview of the Approach

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

We assume that the surveillance locations are chosen before the deployment or identified by the network autonomously after the deployment. The network is organized into clusters around surveillance locations by running a clustering protocol such as the one proposed in [5]. We employ a simple clustering strategy in which the sensor closest to the surveillance location is elected as the cluster head. Each cluster is composed of the sensors whose SNRs are higher than a threshold. Each static sensor belongs to only one cluster. However, a mobile sensor may belong to multiple clusters because it can contribute to the target detection at different surveillance locations.

We propose a data-fusion centric detection model as follows. Initially, all sensors in a cluster periodically send the measurements to the cluster head that compares the average energy against a threshold $\lambda_{1}$. Once a positive detection decision is made, the cluster head initiates the second phase of detection by sending mobile sensors a movement schedule $S$ that specifies which sensors should move, the time instances to start moving and the distances to move. Mobile sensors then move toward the surveillance location according to the schedule. After a certain delay, all sensors send the cluster head the sum of their energy measurements and the final detection decision is then made by comparing against another threshold $\lambda_{2}$. The detection thresholds, $\lambda_{1}$, $\lambda_{2}$ and the movement schedule $S$ are determined under the constraints that the aggregate delay, PD and PF of the two phases must satisfy the requirements specified by $D$, $\beta$ and $\alpha$, respectively.

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

4.2 A Numerical Example

We now illustrate our approach using a numerical example. To simplify the discussion, suppose there is only one surveillance location, which is monitored by 3 static and 3 mobile sensors. The required PD and PF are 90% and 5%, respectively. The average speed of a mobile sensor is 0.5 m/s. During initialization, the cluster head estimates the parameters of target energy model (see (1)) using a training data set. We use the following parameters: $\mathcal{W} = 0.51$ (after normalization), $d_0 = 2.6m$ and $k = 2$, which are estimated using the data set collected in a vehicle detection experiment [14] (the details are given in Section 7).

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

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

4.3 Assumptions

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

In the first phase of detection, all sensors operate in a synchronous sleep schedule in which they wake up to sample energy at a period of $S$ seconds. We assume the probability that a target may appear at any time instance is uniform. Therefore, the expected detection delay due to sleep scheduling is $S/2$. Suppose $S = 2\gamma D$ where $D$ is the required detection delay bound. Thus the expected delay of the first-phase detection is $S/2 = \gamma D$. A constant chosen according to the desirable trade-off between detection delay and power consumption. For the convenience of discussion, we assume $\gamma = 1/2$ in the rest of discussion. After waking up, each sensor samples energy for $T$ seconds and sends to the cluster head. For instance, the acoustic data is recorded at a frequency of 4960 Hz in every 0.75s in the SensitiveIT experiments [14]. That is, $T$ is 0.75s.

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

We assume that the average speed of a mobile sensor is $v$ unless it is stationary. To simplify the motion control of mobile sensors, we assume the moving distance of a mobile sensor in the second phase is always multiple of $vT$. Furthermore, to simplify our problem formulation, we assume that the distance between a sensor and a surveillance location is also multiple of $vT$. We note that this assumption has little impact on the system detection performance as both $v$
and $T$ are small in practice. For instance, $T$ is $0.75s$ in the experiments in [14] and $v$ is $0.5 \sim 2m/s$ for typical mobile sensor systems [21, 11, 19]. Under such settings, $vT$ is at most $1.5$ meters. Therefore, the assumption that the real sensor locations are multiple of $vT$ does not introduce significant errors.

5. The cluster that monitors location $u$ contains a set of sensors indexed as $1, 2, \cdots, n$. The sensors are initially located at $(x_1^0, \ldots, x_n^0)$.

6. $N_s$ and $N_m$ represent the sets of indices of static and mobile sensors, respectively.

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

$$
(P_u \cdot P_{D_1} + (1 - P_u) \cdot P_{D_2}) \cdot ||S||
$$

subject to

$$
P_{F_1} \cdot P_{F_2} \leq \alpha \quad \text{(7)}
$$

$$
P_{D_1} \cdot P_{D_2} \geq \beta \quad \text{ (8)}
$$

$$
\forall M_i(x, t) \in S,
$$

$$
(i \in N_m) \wedge (vT \leq x_i \leq x_i^0) \wedge (0 \leq t \leq \frac{D}{2} - T) \quad \text{ (9)}
$$

$$
\eta_1 \in \{\eta_{1(1)}, \eta_{1(1)} \cdots \eta_{1(k)}\} \quad \text{(10)}
$$

$$
\eta_2 \in \{\eta_{2(1)}, \eta_{2(1)} \cdots \eta_{2(k)}\} \quad \text{(11)}
$$

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

$$
P_{F_1} = 1 - X_a(n\eta_1) \quad \text{(12)}
$$

$$
P_{F_2} = 1 - X_a(nm\eta_2) \quad \text{(13)}
$$

$$
P_{D_1} = 1 - X_n \left( m\eta_1 - \sum_{i=1}^{n} W(x_i^0) \cdot T \right) \quad \text{(14)}
$$

$$
P_{D_2} = 1 - X_n \left( mn\eta_2 - \sum_{i=1}^{m} \sum_{j=0}^{m-1} E_i(j, S) \right) \quad \text{(15)}
$$

$$
m = \frac{D}{2T} \quad \text{(16)}
$$

$E_i(j, S)$ is the energy sampled by sensor $i$ during interval $[jT, (j+1)T]$ under the movement schedule $S$:

$$
E_i(j, S) = \begin{cases} 
\int_{jT}^{(j+1)T} W(x - vt)dt & \text{if } M_i(x, jT) \in S; \\
W(x) \cdot T & \text{if } M_i(x, jT) \notin S; \\
W(x) \cdot T & \text{if } x = \max(x', jT), x = X_m(x', t), \\
t < jT, \text{ otherwise } x = x_i^0
\end{cases}
$$

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

4.4 Problem Formulation

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

1. $P_u$ denotes the probability that a target appears at location $u \in A$ during time $D$, which is known or can be estimated by the history of detection.

2. $x_i$ represents the distance between sensor $i$ and location $u$. We assume $u$ is the origin and hence $x_i$ also represents sensor $i$’s location$^1$.

3. A sensor move, denoted by $M_i(x, t)$, is the process in which sensor $i$ moves from location $x$ to $x - vT$ in time interval $[t, t + T]$ where $T$ is the sampling interval (see Section 4.3).

4. A movement schedule, denoted by $S = \{M_i(x, t)\}$, is a list of moves. $||S||$ represents the cardinality of $S$, i.e., the total number of moves in the schedule.

$^1$As the detection performance of a sensor only depends on its distance to the target, the sign of $x_i$ is insignificant.
(7) and (8) require that the joint PF and PD of the two phases must meet the constraints specified by the application. (9) specifies the spatial and temporal constraints of sensor movement. Each mobile sensor must move between its initial location and the target location, and the movement must complete within $D/2$. At the end of $D/2$, all sensors send their energy measurements to the cluster head which then makes the final detection decision.

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

5 Optimal Solution of the MSD Problem

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

5.1 Structure of the Optimal Solution

The formulation in Section 4.4 shows that the MSD problem is a nonlinear optimization problem with as many as $n D/2T + 2$ variables ($\eta_1$, $\eta_2$ and the movement schedule $S$ composed of at most $n D/2T$ moves). An exhaustive search of all possible values of these variables incurs exponential complexity. In this section, we first analyze the structure of the MSD problem, which allows us to develop an optimal solution that has a polynomial time complexity.

A MSD solution $< \eta_1, \eta_2, S >$ is valid if all constraints can be satisfied. A valid solution is optimal if it minimizes the cost function among all valid solutions. We note that when the movement schedule $S$ is known, unique values of $\eta_1$ and $\eta_2$ can be found. According to $S$, the total sampled energy can be computed by (17) and hence constraints (7) to (11) can be evaluated. An exhaustive search in the domains of $\eta_1$ and $\eta_2$ can find the values that minimize the cost function (6) under the constraints. $S$ is said to be valid/optimal, if the solution constructed by $S$, and $\eta_1$ and $\eta_2$ (that are found by the exhaustive search) is valid/optimal. We now focus on finding the optimal movement schedule.

The search of $\eta_1$ and $\eta_2$ for a given movement schedule is discussed in Section 5.3.

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

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

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

$$c(\eta_1, S) = (P_u \cdot P_{D_1}(\eta_1) + (1 - P_u) \cdot P_{P_1}(\eta_1)) \|S\| \leq (P_u \cdot P_{D_1}(\eta_1') + (1 - P_u) \cdot P_{P_1}(\eta_1')) \|S\| = (P_u \cdot P_{D_1}(\eta_1') + (1 - P_u) \cdot P_{P_1}(\eta_1')) \|S'\| = c(\eta_1', S')$$

\[ \square \]

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

5.2 Optimal Sensor Movement Scheduling

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

![Diagram of sensor movements]

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

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

$$E_1 = \int_0^{2} \frac{dt}{(x - 1 - t)^2} + \frac{W - 2}{(x - 3)^2} + \frac{W}{x^2}$$

$$= \frac{1}{x - 3} - \frac{1}{x - 1} + \frac{W - 2}{(x - 3)^2} + \frac{W}{x^2}$$

$$E_2 = \int_0^{1} \frac{dt}{(x - 2)^2} + \frac{D - 1}{(x - 2)^2} + \int_0^{1} \frac{dt}{(x - 1 - t)^2} + \frac{W - 1}{(x - 1)^2}$$

$$= \frac{1}{x - 2} - \frac{1}{x - 1} + \frac{W - 1}{(x - 2)^2} + \frac{W - 1}{(x - 1)^2}$$

(18)

(19)

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

**Theorem 2.** Suppose an optimal schedule has total $L$ moves, and $n$ sensors move $l_i$ steps, respectively. $L = \sum_{1\leq i\leq n} l_i$. For each sensor $i$, the $l_i$ moves occur consecutively from time zero.

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

$$e_i = \int_0^{l_i T} W(x_i^0 - vt)dt + t_1 W(x_i^0)$$

$$+ \sum_{2 \leq j \leq K} t_j W(x_i^0 - v T) \sum_{1 \leq m \leq j - 1} l_i^m$$

$$+ \frac{D}{2} - \sum_{2 \leq j \leq K} t_j - l_i(t_j) \cdot W(x_i^0 - v l_i T)$$

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

$$e_i = \int_0^{l_i T} W(x_i^0 - vt)dt + \left(\frac{D}{2} - l_i(t_i)\right) \cdot W(x_i^0)$$

$$+ t_1(W(x_i^1) - W(x_i^0 - v l_i T))$$

$$+ \sum_{2 \leq j \leq K} t_j(W(x_i^j) - W(x_i^0 - v l_i T))$$

As $l_i \geq \sum_{1 \leq m \leq j - 1} l_i^m$ and $W(\cdot)$ is a decreasing function, the above equation is maximized when $t_i = 0$ ($1 \leq i \leq K$).

That is, all the moves of sensor $i$ are continuous and start at time zero. \[\square\]
According to Theorem 2, the number of possible move combinations in the optimal schedule is significantly reduced. We now present a dynamic programming algorithm that finds the optimal schedule for a given number of sensor moves.

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

\[
e_i(h_i) = \int_0^{h_i T} W(x_i^0 - vt) dt + \frac{D}{2} - h_i T \cdot W(x_i^0 - vh_i T)
\]

(20)

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

\[
E(j, h) = \max_{0 \leq h_j \leq H_j} \{ E(j - 1, h - h_j) + e_j(h_j) \} \quad (21)
\]

\[
H_j = \min \left\{ \frac{D}{2}, \frac{x_j^0}{v T} \right\} \quad (22)
\]

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

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

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

\[
S(j, h) = S(j - 1, h - h_x) \cup \{ M_i(x, v x T) | 0 \leq x \leq h_x - 1 \}
\]

\[
h_x = \arg \max_{0 \leq h_j \leq H_j} E(j - 1, h - h_j) + e_j(h_j)
\]

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

5.3 The Procedure of Solving the MSD Problem

We now present the procedure of solving the MSD problem. For each possible number of moves, \( l \), we first compute \( E(n, l) \) and the movement schedule \( S(n, l) \) using the scheduling algorithm described earlier. Then the values of \( \eta_1 \) and \( \eta_2 \) are searched to minimize the expected sensor moving distance under the constraints. The maximum number of moves is given by \( H = \sum_{1 \leq i \leq n} H_i \) where \( H_i \) is given by (22). The optimal movement schedule and \( \eta_1 \) and \( \eta_2 \) can then be found in \( H \) iterations. Fig. 4 shows the pseudo code of the procedure.

For each value of \( \eta_1, P_{D_j} \), and \( P_{F_k} \), can be computed according to (14) and (12). Furthermore, unique \( P_{F_2} \) can be determined as the minimum value that satisfies constraint (7). This is because that a higher \( P_{F_2} \) leads to a higher \( P_{F_2} \), which may cause constraint (7) to be violated. Then \( \eta_2 \) can be solved from \( P_{F_2} \) according to (13). So far, constraints (7), (9), (10) and (11) have been satisfied. For instance, constraint (7) is enforced in solving \( \eta_2 \). It remains to check if constraint (8) is met. A new cost is computed according to (6) if 8 is met. A zero cost may occur when all constraints are satisfied without moving the sensors toward the target. If the new cost is lower than the current cost, the current movement schedule and detection thresholds are recorded. As \( E(n, l) \) and \( S(n, l) \) can be pre-computed using the scheduling algorithm, the complexity of the procedure is \( \mathcal{O}(H \cdot k) \).

6 Extensions

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

6.1 Multi-phase MSD Problem

In the two-phase model, a false alarm in the first phase inevitably results in the movement of sensors. We address this issue by a multi-phase detector in which measurements are fused periodically and mobile sensor stop moving once a negative decision is made. The M-phase MSD problem is to minimize:

\[
\sum_{i=1}^{M-1} \left[ \left( P_u \prod_{j=1}^{l} P_{D_j} + (1 - P_u) \prod_{k=1}^{l} P_{F_k} \right) \cdot \|S\| \right]
\]

subject to
Input: \( D, \{ E(n,j) \mid 0 \leq j \leq H \}, P_{u,v} \) (\( x_1^0, x_2^0, \ldots, x_n^0 \)), \( \{ \eta_{1(0)}, \eta_{1(k)} \}, \{ \eta_{2(0)}, \eta_{2(k)} \} \)

/*output movement schedule and two detection thresholds*/

Output: \( S, \eta_1, \eta_2 \)

1. cost = \( \infty \);
2. for \( l = 0 \to H \) 
3. \hspace{1cm} for \( n_1 = [\eta_{1(0)} : \eta_{1(k)}] \) 
4. \hspace{2cm} Compute \( P_{D_1} \) and \( P_{F_1} \) using (14) and (12); 
5. \hspace{2cm} Find the minimum \( n_2 \to \{ \eta_{2(0)} : \eta_{2(k)} \} \) using (7); 
6. \hspace{2cm} Compute \( P_{D_2} \) using \( E(n,l) \) according to (15); 
7. \hspace{2cm} if ((8) holds) 
8. \hspace{3cm} Compute current cost \( C \) using (6); 
9. \hspace{3cm} if \( (C = 0) \) exit; \( \mathrm{fi} \); 
10. \hspace{3cm} if \( (C < \mathrm{cost}) \) 
11. \hspace{4cm} \( \mathrm{cost} = C; S = S(n,l); \eta_1 = n_1; \eta_2 = n_2 \) 
12. \hspace{3cm} \( \mathrm{fi} \) 
13. \hspace{2cm} \( \mathrm{fi} \) 
14. end 
15. end

**Figure 4. The procedure of solving the MSD problem.**

\[
\prod_{i=1}^{M} P_{F_i} \leq \alpha \quad (24)
\]

\[
\prod_{i=1}^{M} P_{D_i} \geq \beta \quad (25)
\]

\[\forall i \in [1,M], \eta_i \in \{ \eta_{1(0)}, \eta_{1(1)}, \ldots, \eta_{1(k)} \} \quad (26)\]

Constraint (9) is also applicable and not shown. The cost function (23) is the total expected moving distance in all M phases. Note that sensors move in the \( l^{th} \) phase only if the decisions of all the previous phases are correct or false alarms. The M-phase MSD problem can be solved similarly as the two-phase MSD problem. However, the time complexity increases to \( \mathcal{O}(H \cdot k^{M-1}) \).

### 6.2 Detection of Multiple Targets

When multiple concurrent targets are close to each other, efficient coordination among different clusters is needed as the movement of sensors toward a target may affect the detection performance of other targets.

We now discuss an on-demand coordination strategy to handle multiple targets. Suppose \( A_u \) and \( A_v \), are two cluster heads which monitor locations \( u \) and \( v \) and share sensor \( i \).

We assume that a sensor can correctly sense the target signal with the highest SNR when multiple targets appear in its detection range. When \( A_u \) detects a possible target and requests \( i \) to move toward \( u \), sensor \( i \) sends \( A_v \) its distance and direction to \( u \). \( A_v \) then adjusts its detection thresholds and sensor movement schedules as follows. First, \( A_v \) labels sensor \( i \)'s initial position by the time instance it is available, which can be calculated according to sensor \( i \)'s round-trip time. For instance, it takes sensor \( i \) 30 seconds to move to the location of target \( u \) and then returns to its original position. Therefore, \( i \) can be scheduled to move toward target \( v \) only after 30 seconds. In the movement scheduling algorithm, the total energy measurement of a sensor (equation (20)) is then calculated according to its available time intervals. The coordination among different clusters maximizes the utility of mobile sensors by taking into account their available times when detecting multiple targets.

### 7 Simulations

In this section, we present simulation results based on the real data traces collected by Duarte et al. [14]. In their experiments, 75 WINS NG 2.0 nodes are deployed to detect Assault Amphibian Vehicles (AAVs) driving through several intersected roads. The data set used in our simulations includes the time series recorded by 23 nodes at the frequency of 4960Hz. Received energy is calculated every 0.75s. We refer to [14] for more detailed setup of the experiments.

#### 7.1 Methodology and Simulation Settings

The simulation code is written in C++. We use the AAV3 data set in [14] as the training data for estimating the energy attenuation model defined by (1). Our estimated parameters are: \( S_0 = 0.51 \) (after normalization), \( d_0 = 2.6m \) and \( k = 2 \). The estimated energy model is used by cluster heads to run the algorithm shown in Fig. 4 that computes the detection thresholds and the movement schedule of sensors. In each run of simulations, when a sensor makes a measurement, the energy is set to be the real measurement gathered by a sensor at a similar distance to target in the data trace.

Sensors in our simulations are randomly distributed in a sensing field of 50 x 50m² surrounded by four roads, as illustrated in Fig. 8. Vehicles drive along the roads at a constant speed of 2.5m/s, which is similar to the average speed of the AAVs in the data traces. The simulation time of each run is \( 10^5 \) seconds. In the first set of simulations, we evaluate the performance of detecting one target. The probability that a vehicle appears at any time instance (in the unit of seconds) is 5%. Once a vehicle appears on a road, the minimum interval before the next vehicle appears on the
same road is set to be 30 seconds. In Section 7.2.2, we increase the probability of target appearance and evaluate the performance of detecting multiple targets. The detection delay requirement, $D$, is set to be $16s$. The requested false alarm rate ($\alpha$) and detection probability ($\beta$) are set to be 0.01 and 0.9 unless otherwise specified. In each run of simulations, the speed of mobile sensors is randomly chosen within $0.5 \sim 1m/s$.

We note that our simulation settings account for several realistic factors. First, there exists considerable deviation between the measurements of sensors in our simulations and the training data used to estimate the target signal model. This deviation is due to various reasons including the difference between vehicles and the changing noise level caused by wind. Moreover, our movement scheduling algorithm assumes that targets remain stationary at each surveillance site before disappearance. However, each AAV in our simulations drives along a road. As a result, the actual SNRs received by sensors are considerably lower than those used in the movement scheduling algorithm. The performance of our solution can be improved if the mobility of targets is explicitly taken into consideration, e.g., by integrating with target tracking algorithms [5].

7.2 Simulation Results

This section presents the results of our simulations. We conduct three sets of simulations that evaluate the impact of the number of mobile sensors on system detection performance, the effectiveness of our movement scheduling algorithms, and the impact of the speed of mobile sensors, respectively.

7.2.1 Impact of the Number of Mobile Sensors

In the first set of simulations, targets only appear at one surveillance site and remain stationary. The performance of detecting moving targets is evaluated in the rest of this section. To evaluate the impact of mobility on system detection performance, we plot four receiver operating characteristic (ROC) curves under different numbers of mobile nodes in Fig. 5. ROC curves characterize a detection system’s achievable trade-off between PD and PF. In Fig. 5, Static refers to the deployment in which all sensors remain stationary. Total 6 sensors are deployed. We can see that the system detection performance increases significantly with the number of mobile sensors. In particular, when all 6 sensors are mobile, the improvement of detection probability is about $20 \sim 40\%$.

Figure 6 shows the detection probability when the number of sensors varies from 4 to 20. In each setting, the detection threshold is computed to maximize the system PD under a PF of 0.05. We can see that PD reaches about 81% when only four sensors are mobile. In contrast, PD is only about 42% if all sensors are static. When the total number of sensors increases, the system performance under different settings becomes similar because a near 100% PD can be achieved without moving sensors. Fig. 6 also shows that the use of mobile sensors can significantly reduce the
density of sensors needed in a deployment. For example, 8 mobile sensors achieve a similar detection performance as 20 static sensors.

### 7.2.2 Performance of Movement Scheduling

We now evaluate the performance of our movement scheduling algorithm. To distinguish from baseline algorithms, we refer to our two-phase detection algorithm as the *mobility-assisted detector (MD)*. Besides MD, we also simulate a three-phase detection algorithm (as described in Section 6) referred to as MD-3. We compare MD and MD-3 against two baseline algorithms. *MD-random1* is a variant of MD that employs a random movement scheduling algorithm. At each scheduling step, a cluster head always randomly chooses a sensor to move until the required detection performance is achieved. *MD-random2* is another variant of MD that uses a node-based random scheduling strategy. A cluster head randomly chooses a sensor and continues moving it until it reaches the location of target or the required detection performance is achieved.

Fig. 7 shows the average number of moves of 10 mobile sensors when the requested PD varies from 0.8 to 0.95. The PF is set to be 0.01. MD and MD-3 significantly outperform the two baseline algorithms, which demonstrates the effectiveness of our optimal movement scheduling algorithm. MD-3 is superior to MD when the requested PD is above 85%. This is because, when the number of moves needed increases significantly, an additional detection phase can filter out more false alarms and reduce unnecessary moves.

We now evaluate the performance of detecting multiple targets. The probability that a vehicle may appear at any time instance varies from 5% to 40%. Once a vehicle appears on a road, the minimum interval before the next vehicle appears on the same road is set to be 30 seconds. That is, we enforce that only one target may appear within a detection window on any road. However, multiple vehicles may drive on different roads at the same time. We plot PD versus the probability a target appears in Fig. 9. When multiple targets appear in the same time window $D$, the four cluster heads in the sensing field employ the coordination mechanism discussed in Section 6.2. When a mobile sensor is requested to move by a cluster head, it notifies all other cluster heads, which update their detection thresholds and movement schedules. For performance comparison, we implement a baseline algorithm referred to as MD-unsync, which does not update detection parameters when mobile sensors move away. Fig. 9 shows PD versus different target appearance rates. Each algorithm is labeled by the name and the number of sensors used. We can see that MD yields considerably higher detection probabilities than MD-unsync. The performance of MD and MD-unsync degrades when more targets appear concurrently. In such a case, moving sensors toward a particular target often lowers the performance of detecting other targets. In contrast, the performance of static deployments remain unaffected. Nevertheless, MD always yields the best performance in all settings.

### 7.2.3 Impact of Mobile Sensor Speed

We now evaluate the impact of movement speed on the system detection performance. Fig. 10 shows the probabilities of detecting multiple targets. We can see that the detection performance increases considerably when the speed of movement becomes higher because mobile sensors can move closer to targets within the given delay bound. This result also shows that our movement scheduling algorithm can effectively take advantage of the increase of movement speed. In reality, the speed of a mobile sensor may suffer from variations because of complex terrains or temporal mechanical problems. We now evaluate system PD when the movement speed has different variations. At the beginning of simulations, the average speed of each mobile sensor, $v$, is randomly chosen between $0.5 \sim 1 \text{ m/s}$. The actual speed of the sensor in each move (that takes 0.75 s)
is randomly chosen within $[(1 - \delta)v, (1 + \delta)v]$. As shown in Fig. 11, PD yields a higher variation when $\delta$ is larger. However, the average PD remains similar even $\delta$ increases to 50%. This is because our algorithm always schedule multiple sensors to move in order to achieve the maximum PD. As a result, the variation in each individual sensor’s speed does not have a significant impact on the overall system PD.

8 Conclusion

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

References


