

Adaptive Tracking in Energy Sensitive Distributed Wireless Sensor Networks

CHUAN FENG, LIZHI YANG AND JERZY W. ROZENBLIT

*Department of Electrical and Computer Engineering,
University of Arizona, Tucson, AZ 85721 USA
E-mail: fengc@email.arizona.edu*

Received: October 3, 2009. Accepted: May 4, 2010

We study the problem of tracking moving targets using distributed Wireless Sensor Networks (WSNs) in which sensors are deployed randomly. Prediction-based techniques are a commonly used strategies to reduce the power usage of energy sensitive wireless sensor networks where sensor nodes are battery-powered. However, due to the uncertainty and unpredictability of real-world targets' motion, the power efficiency of tracking and the accuracy of prediction are reduced. The tracking algorithm must adapt to the real-time changes in velocity and direction of a moving target. In this paper, we proposed a novel energy efficient tracking algorithm called Predict-and-Mesh (PaM) which is suitable for energy sensitive distributed wireless tracking systems. Making use of the PaM algorithm, it is possible to adaptively adjust the sensing frequency for pervasively monitoring various kinds of targets with random movement patterns. In addition, a prediction failure recovery mechanism called "mesh" is proposed to relocate the targets under tracking. Simulation results show that the PaM algorithm is robust against diverse motion changes and has excellent performance.

Keywords: Wireless Sensor Network, Tracking System, Adaptive Tracking, Predict, Mesh, Energy Saving

1 INTRODUCTION

A Wireless Sensor Network (WSN) is an interconnected system of a large set of physically small, low cost, low power sensors that provide ubiquitous sensing and computing capabilities. The sensors have the ability to sense an environment in various modalities (e.g., temperature, sound, light, seismic and

distance), process information, and disseminate data wirelessly. Therefore, the WSN can potentially reduce or eliminate the need for human involvement in information gathering in broad civilian or military applications such as national security, health care, environment protection, energy preservation, food safety, and so on [26].

The design of a WSN is highly application-dependent, i.e., for different applications there are different design and technical issues that need to be addressed. In this paper, we focus our attention on an important application of wireless sensor networks, tracking moving targets [2]. Our goal is to design an efficient tracking approach based on distributed randomly networked sensors in a large scale region. Possible scenarios can be border control, battle field surveillance, traffic flow measurement, animal monitoring, etc.

In developing such a tracking system, tracking algorithm design is a complicated problem since there are several constraints that need to be taken into account. Some are inherent in the nature of wireless sensors, e.g., the sensors may have a limited and irreplaceable power supply, limited sensing ability, limited communication bandwidth or limited computational power. All of these limitations impose constraints on the efficacy of tracking [24]. Therefore, traditional tracking mechanisms are not sufficient for the WSN. For some tracking applications, power efficiency is of great concern, so the tracking algorithm must be designed to expend as little energy as is possible. Moreover, since surveillance and tracking systems are often deployed in critical or hostile environments where functional failures are vital, design priority should be given to both the quality and the reliability of tracking [32]. Arora and et. al. performed a nearly year long experiment on WSN tracking and identified several important issues including network reliability, system scale, and node failures [1].

Given a large surveillance area, the tracking algorithm must be able to accommodate a large number of entities to cover the entire area. Generally, the deployed WSN is distributed. In other words, a fixed superior/subordinate relationship is not effective in such a network. Although a centralized architecture is theoretically optimal and also conceptually simple [20], it is not suitable to a large scale environment because of the limited communication bandwidth and power supplies of wireless sensors.

For certain applications (e.g., military applications), it may be desirable for the sensors to be dropped from an aircraft or by other means into the hostile environment without any further adjustment. Such random deployment strategies may lead to severe coverage problems given the sensors' communication and detection constraints [8, 37]. Thus, the designed algorithm should be adaptable to "blind" areas, i.e., regions that are partially or fully missed by any sensor. In addition, while the position of a sensor is usually fixed after deployment, the network topology may frequently change due to functional failure, physical damage, lack of power or introduction of new sensors to the network. In other words, the WSN for tracking targets may be ad hoc.

Typically, the tracking scheme is fixed when considering the problem of tracking targets in the distributed WSNs. However, in many tracking applications, the motion characteristics of the targets being tracking may vary due to uncertainty and unpredictability in the target motion model [3]. Moreover, different types of targets may have different kinds of motion characteristics. For example, a tank has a much larger maximum velocity than a human soldier. On the other hand, a moving human soldier can perform turns much quicker than a tank. Such variations pose a significant problem when designing a tracking system.

Though several tracking approaches have been proposed in previous literatures (cf. Section 2), few has been designed specifically for adaptability. Thus, they lack the flexibility needed to adapt to the real time changes in direction and velocity of moving targets under tracking. In particular, several existing approaches are based on predictive tracking algorithms [2, 33], in which the accuracy of prediction depends on the frequency of sensing [11, 31]. In this paper, we proposed an energy efficient tracking algorithm called Predict-and-Mesh (PaM) which is suitable for energy sensitive distributed wireless tracking systems. The PaM algorithm adjusts the sensing frequency for pervasively monitoring various kinds of targets with random movement patterns. A prediction failure recovery mechanism is also proposed to relocate the targets under tracking. Compared to some existing work [33, 7, 27], this method requires less computational resources. Therefore, it will not impact the normal activity of a wireless sensor node. We show that the PaM algorithm is robust against diverse motion changes and provides excellent performance.

The organization of the paper is as follows. Section 2 briefly surveys related work on energy efficient tracking in wireless sensor networks. Section 3 presents underlying assumptions and defines preliminary models. In Section 4, we describe the details of the PaM algorithm including prediction models and the “mesh” prediction failure recovery mechanism. In Section 5, we evaluate the performance of our algorithm by comparing PaM against others. We showed the benefits of the proposed approach through simulations. Section 6 concludes the paper and outlines directions for future work.

2 RELATED WORK

A variety of problems in target tracking using wireless sensor networks have been studied. Among these works, [8, 23] and [37] present sensor deployment schemes to ensure adequate coverage of moving targets, while [22] provides a definition of the coverage problem from several points of view and formally defines the best and worst-case coverage in a sensor network. In the area of collaborative sensing, [35] and [19] present tree-based collaboration in tracking. In [36] and [21], an information-driven sensor collaboration mechanism and a dual-space approach to tracking targets which enable selective activation

of sensors are presented, respectively. Some power conservation protocols such as SPAN [6], LEACH [14] and SPEED [13] and data communication approaches (e.g., [5, 14, 17]) which concentrate on efficient data prorogation are proposed for wireless sensor networks.

In the area of energy efficient tracking, a number of energy conservation approaches have been proposed. A collaborative signal processing (CSP) framework could be used to classify and track different targets [4]. In addition to data communication protocols and signal processing techniques, tradeoffs between the quality and energy efficiency of target tracking are studied in [23] and [10]. In this paper, we propose and study energy efficient tracking approaches in distributed wireless sensor networks, so that the issue of network routing between sensors and servers is not concerned. Our research concentrates on effective and reliable activation schemes to reduce power consumptions in tracking.

Intuitively, turning off unnecessary sensors effectively enhances the lifetime of the entire system since energy consumption increases significantly during the periods of activity [37, 11]. In [26, 31, 34] and [28], prediction based tracking algorithms are presented, which accommodate a sensor hibernation mechanism to conserve power and extend the tracking system's lifetime. Given the past reading history and spatial and temporal relationships of readings from neighbor sensors, the future reading for a sensor can be predicted [9, 32, 30, 18]. In order to facilitate this predictive mechanism, most of today's sensor nodes operate under two modes, namely, active and sleep. These two working modes can be achieved by turning on/off some basic functional units such as Micro-Controller Unit (MCU), radio communication components and sensing related components separately as needed.

However, due to the unpredictable behavior of targets, e.g., random moving speeds and directions, the quality of tracking can be dramatically lowered. In [29], several different prediction based algorithms for tracking mobile targets have been studied. It is proved that tracking efficacy is different when using different prediction strategies. In addition, a detailed quantitative analytical model is proposed in [11]. In that model, the dependency of the accuracy of prediction on tracking interval representing the time length between two consecutive sensing points is represented by a quadratic function. Subsequently, the optimal solution that is subject to the minimum power consumption can be obtained. However, when tracking different kinds of targets (e.g., human and vehicle) with different moving patterns (e.g., acceleration and velocity), the optimal solution will be different. As shown in Fig. 1 and Fig. 2, the optimal sampling frequencies for tracking a human and a vehicle in those cases are about 0.5 second and 7.5 seconds respectively. Moreover, in a distributed environment, it is hard to adjust the sensing frequency when tracking different types of targets. Thus, a more efficient approach is required when tracking diverse targets within the same tracking region.

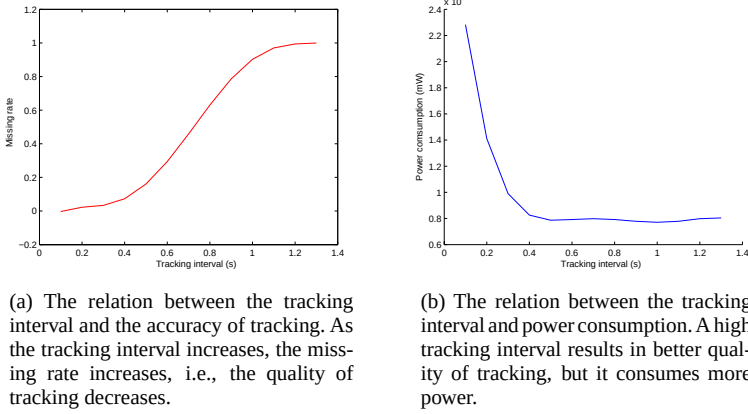


FIGURE 1

Simulation results for tracking a vehicle using the linear prediction tracking algorithm in [11].

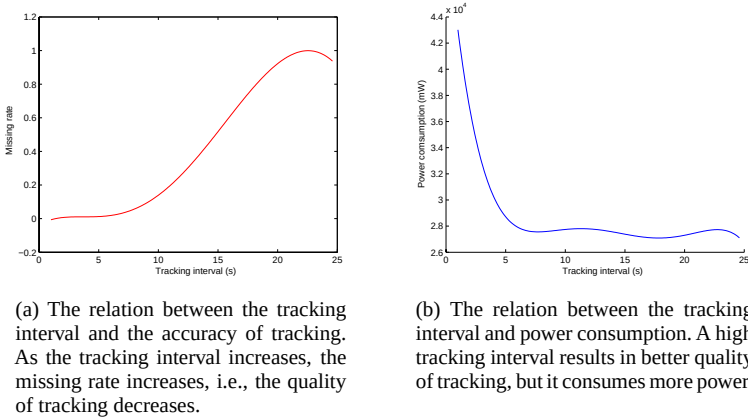


FIGURE 2

Simulation results for tracking a human using the linear prediction tracking algorithm in [11].

3 PRELIMINARY MODELS

In this section, we describe the underlying assumptions and preliminary models which help us better explain our approach in later sections.

3.1 The Power Usage Model

Given n sensors at time t , there are two possible states for each sensor, active and sleep, respectively [16]. For a typical WSN node, the system consumes

19.5 mA of current at peak load and only 10 μ A in the sleep mode [15]. Thus, a sensor node's power consumption is modeled as a structure

$$\mathcal{M} = (\text{State}, P, \delta, \lambda, e) \quad (1)$$

where *State* is the state of the sensor node, *P* is the power of the sensor node, $\delta : \text{State} \rightarrow \text{State}$ is system state transition function, $\lambda : \text{State} \rightarrow P$ represents the power function, and *e* denotes the elapsed time. The interpretation of these elements is shown in the following.

The sensor node system is always in some state, *State*, lasting *e* time. The power level *P* during this time period is determined by the function $\lambda : \text{State} \rightarrow P$. If any internal or external event happens, the system will change its state based on the state transition function $\delta : \text{State} \rightarrow \text{State}$. Therefore, the power level will also be changed based on λ . The power consumption within a time period *T* is shown in the following formula.

$$E = \int_0^T P_t dt = \int_0^T \lambda(\text{State}_t) dt \quad (2)$$

Where *E* is the total power consumption of the sensor node, P_t is the power on time *t*, which is determined by the power function $\lambda(\text{State}_t)$.

For each sensor, power consumption will vary under various states. Typically, there are four power usage modes for each active sensor, namely, *idle*, *sensing*, *transmitting* and *receiving*. Otherwise, the sensor node will be in the *power down* mode as sleeping. “*Power down*” indicates a working mode shutting off everything but the necessary circuits for waking up. Thus, any event in the sensing field may result in state transition for a sensor node as

$$\delta : \text{State} \rightarrow \text{State} = \{\text{State}_\iota, \text{State}_\sigma, \text{State}_\tau, \text{State}_\rho, \text{State}_\Delta\} \quad (3)$$

where δ denotes the state transition function, State_v represents a sensor in state *v*, which can be ι , σ , τ , ρ , and Δ . Here ι represents the sensor in the idle mode, σ represents the sensor in the sensing mode, δ indicates that the sensor is in the power down mode, τ and ρ denote the sensor in the transmitting mode and the receiving mode, respectively. The sensor power consumption modes are summarized in Table. 1.

	Idle	Sensing	Receiving	Transmitting	Power down
Active	Y	Y	Y	Y	N
Sleep	N	N	N	N	Y

TABLE 1
Power usage modes.

Moreover, the power function is shown in equation 4.

$$\lambda : State \rightarrow P = \{P_\iota, P_\sigma, P_\tau, P_\rho, P_\Delta\} \quad (4)$$

where λ denotes the state transition function, P_v denotes the power of a sensor under different working states $State_v$. As above, v can be $\iota, \sigma, \tau, \rho$, and Δ .

According to equations 2 and 4, power consumption at any given period T for a sensor is given by

$$E = P_i \cdot t_i + P_e \cdot t_e + B_r \cdot P_r \cdot t_r + B_t \cdot P_t \cdot t_t + P_d \cdot t_d \quad (5)$$

$$T = t_i + t_e + t_r + t_t + t_d \quad (6)$$

where E is the power consumption of the sensor, T is the given period, t_i is the idle period of the sensor, t_e denotes the time required for the sensor to optimally estimate the position of an target, t_d is the power down period of the sensor, t_t is the time required for the sensor to send out a unit packet (transmission bit rate), t_r is the time required for the sensor to receive a unit packet, B_r and B_t represent the size of packets that are received and transmitted by the sensor, respectively. In reality, sensor detection is imprecise, therefore appropriate signal processing approaches such as Kalman filter are often employed to estimate the position, velocity and acceleration of a target [3, 30]. Once a sensor node is activated, it will be in the idle status and the radio components will only be used when data communications are needed. Given a WSN with m identical sensors, the power consumption E can be represented by

$$E = \begin{pmatrix} t_{1i} & t_{1e} & B_r \cdot t_{1r} & B_t \cdot t_{1t} & t_{1d} \\ t_{2i} & t_{2e} & B_r \cdot t_{2r} & B_t \cdot t_{2t} & t_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{mi} & t_{me} & B_r \cdot t_{mr} & B_t \cdot t_{mt} & t_{md} \end{pmatrix} \times \begin{pmatrix} P_i \\ P_r \\ P_e \\ P_t \\ P_d \end{pmatrix} \quad (7)$$

where E is a vector of which each entry represents the power consumption of each sensor in the network.

Thus, assuming an target under tracking has been in the sensing field for a time interval T , the overall power consumed by the WSN for tracking this target can be formulated as

$$E_o = \sum_{i=1}^m E_i \quad (8)$$

where E_o is the overall power consumption, $E_i, i = 1, 2, \dots, n$, denotes entries in the vector E . Note that P_i, P_r, P_e, P_t and P_d in equation 7 highly depend

on the hardware platform of the sensor nodes. For instance, given a specific hardware platform, e.g., Berkeley MICA motes [15], one second of the sleep mode can save enough power for sending more than 70 packets, or performing 70K operations [21]. Therefore, power consumption can be conserved by applying an efficient tracking algorithm in which most of sensors will be in the sleep status.

3.2 The Sensing and Communication Model

We assume that the sensors used to detect targets have a 360° degree sensing scope r . Examples of this kind of sensor founded in are acoustic, seismic, and electromagnetic sensor arrays. In general, these sensors are physically small and equipped with limited power supplies, thus r is constrained to be small. Within the detectable distance r , an active sensor node S_i is able to estimate its distance D and orientation θ (θ is within $[0, 360^\circ]$) to an target [21]. In order to simplify the problem, we assume that the distance and orientation information can be accurately estimated in t_e seconds. This can be implemented by a sensor array such as accurately mounting several cheap sonar sensors onto one frame. The size of the sensor can be relatively small (sonar sensor array) or big (radar station) upon to different requirements. Thus, the relative position vector R of target to sensor can be calculated using the following function

$$R = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} D \cdot \cos \theta \\ D \cdot \sin \theta \end{pmatrix}. \quad (9)$$

As the target moves out of the sensing scope of S_i , another sensor node S_j needs to be activated to catch the target. In order to conserve power usage, S_i will be switched to the sleep state after it gets an acknowledgement from sensor S_j . Thus, the tracking task is carried out by sequentially activating and shutting the sensors located along the track of the moving target. Note that the sensing system is distributed, thus there are information exchanges among sensors to enable them to collaborate in tracking the target. Typically, each sensor has a wireless communication scope R which is relatively larger than r . Thus, the sensor node S_i can directly communicate with any other sensor node S_j within the circle area with radius R , centered by S_i . In order to accurately activate a sensor S_j in a specific position, the sensing node S_i has to know its own position and the position of S_j . Since the position of each sensor node S_i is not predetermined in a randomly deployed sensor network, a network localization method is needed. We assume that the sensing system consists of a supernode subset in which all of the sensor nodes are equipped with a precise position system which have knowledge of their own locations [12, 16]. Therefore, theoretically each sensor node in the sensor network is able to obtain a location list of all other sensors within its communication range R .

4 PAM TRACKING

4.1 The Background

As we discussed in Section 2, several prediction based tracking algorithms have been proposed in the previous literature. Some are designed for a centralized network architecture. Thus, they are not suitable for large scale tracking applications, which are much more complicated. Sensors may be dropped randomly with a uniform distribution to cover a large scale region to be monitored. Example of such a surveillance zone can be a sensing field is 100 meters by 100 meters and a total of 1000 sensors are uniformly placed. The sensors need to collaborate to track targets. The idea of the proposed prediction based tracking algorithms is that the lifetime of such a sensing system can be extended by using a set of prediction based activation mechanisms, i.e., control activities of the sensors (active/sleep) for tracking purposes. The word “predict” denotes that the system is able to predict the next location of the target by using the time series trajectory of its path (Fig. 3). The predicted position can be calculated by

$$P_2 = P_1 + \vec{v}t \quad (10)$$

where P_1 is the history target coordinate (x_1, y_1) , P_2 is the predicted position, t is the sensing time interval, and $\vec{v}$ is the estimated target velocity. Where $\vec{v}$ can be given by

$$\vec{v} = V \times \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \quad (11)$$

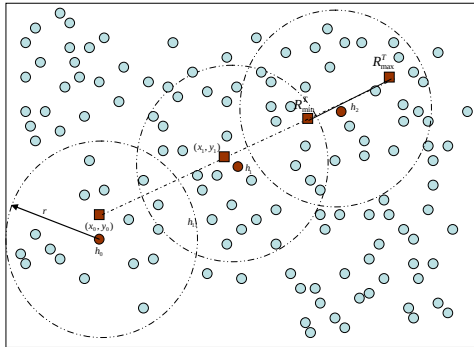


FIGURE 3

The generic prediction model. For instance, when the target is at the position (x_1, y_1) , the speed and orientation of the moving targets can be calculated based on history information (x_1, y_1) and (x_0, y_0) . Thus, the next position can be predicted based on this model.

where the direction α (α is within $[0, 360^\circ]$) can be given by

$$\begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} = \begin{pmatrix} \frac{x_1 - x_0}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}} \\ \frac{y_1 - y_0}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}} \end{pmatrix}, \quad (12)$$

and the speed V can be obtained by

$$V = \frac{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}}{t}. \quad (13)$$

Where (x_0, y_0) and (x_1, y_1) provide historical position information, t represents the sensing period, i.e., the tracking interval which denotes the time length between two consecutive sensing points. In [29], some heuristic approaches that can be used to estimate the speed and orientation of the moving target are proposed.

Given the predicted location, appropriate sensors in that region will be activated to sense the target. Using this mechanism, initially, most of the sensor nodes will be in sleep status in order to conserve energy until activity is triggered by mobile targets. Once a target is detected, the active nodes, which are not only sensing centers but also coordinators (SCs), will activate another set of appropriate sensors as the next SCs. Subsequently, the previous SCs will then change to the sleep status (non-sensing and power down). It is clear that the fewer nodes that are active, the less power is consumed. Intuitively, the node selection criteria can be defined by the shortest distance between the predicted position and the position of the sensors.

4.2 The PaM Prediction Model

Our tracking prediction mechanism is different from previous works. Assume that there are N wireless sensors (e.g., acoustic sensors), which are initially in the sleep mode in the sensing field. Some special sentinel sensors always in active status are also deployed at the boundary of the field. A mobile target traverses through this sensing field with a speed rate V . The target is first detected by an active sensor S_i . Then, after Δt seconds, the target may move out of the detection range r of S_i . Since the target moves randomly, it is necessary to estimate the possibility of this scenario to avoid losing the target. Because the sensor has limited power, the goal of the estimation is to activate as few sensors as possible to save power. Another energy factor in the power consumption used in data communication. It is desirable to eliminate unnecessary communication between sensors as much as possible. Obviously, a good prediction mechanism should also maintain high quality and resolution of tracking as required in various applications.

However, the prediction scheme design is complex due to the unpredictable behaviors of the target. Based on the sensing and communicating model, the prediction model may vary given a target with a different speed rate V and acceleration a . The speed V is recursively measured by active sensor nodes. For example, a target M_i traveling at speed V_i is detected by a sensor S_j at time t at position (x_1, y_1) with the distance $D(M_i, S_j)$ ($D(M_i, S_j) \leq r$) and the orientation $\theta((M_i, S_j))$ to S_j . As shown in Fig. 4, intuitively, the fastest way for the target leaving the sensing area is bearing off along the radius direction, which means the velocity direction α is same as the target orientation θ . It will take at least t_o seconds for M_i to move out of the sensing range r of S_j . In order to ease the computation in sensors, we denote t_o as an *escape period* given by

$$t_o = (r - D(M_i, S_j))/V_i \quad (14)$$

It is clear that the target can be detected again by sensor S_i within time t_o if V_i is not increased. Otherwise, V_i has to be updated as V'_i as the target moves out of range of S_j and the sensor S_k has to be activated to detect this target. The V'_i can be estimated by

$$V'_i = V_i + \int_0^{t_o} \alpha_i(t) dt \quad (15)$$

where α_i denotes the possible acceleration of a specific target which uniformly falls in $[0, \alpha_{max}]$. Thus, V'_i falls in a speed interval $[V_i, V_i + \alpha_{max} \times t_o]$. Notice

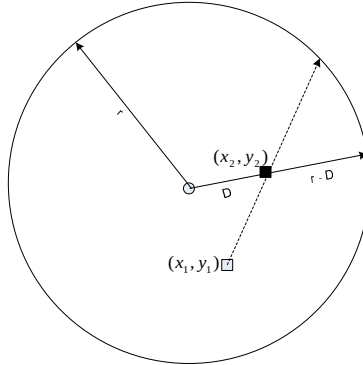


FIGURE 4

The diagram of the escape period. Assume a target M_i travelling at a speed V_i is detected by a sensor S_i at time t at position (x_2, y_2) with the distance $D(M_i, S_i)$ ($D(M_i, S_i) \leq r$) and the orientation $\theta(M_i, S_i)$ to S_i . Intuitively, it will take at least t_o seconds for M_i to move out of the sensing range r of S_i .

that $(V_i + \alpha_{max} \times t_o) \leq V_{max}$, where V_{max} denotes the maximum speed of the target.

The maximum speed V_{max} and maximum acceleration α_{max} are highly related to specific application requirements. For instance, if the WSN is deployed to do border control, then the typical targets will be human beings or vehicles. Therefore, we can set $V_{max} = 40m/s$ and $\alpha_{max} = 10m/s^2$.

Assume that at time $t + t_o$ the sensor S_j detects the target at position (x_2, y_2) so that there are two pieces of position information with respect to target M_i in the knowledge base of S_j . The speed V_i can be accurately calculated by

$$\vec{V}_i = \frac{\vec{P}_2 - \vec{P}_1}{t_o}. \quad (16)$$

Where $\vec{V}_i$ is the speed vector with speed rate V_i . $\vec{P}_n$ is the position vector of (x_n, y_n) . Note that the sensor S_j will keep updating V_i until the target moves out of its sensing range.

Making use of equations (14) and (16), the sensing system is capable of refreshing the knowledge of the target under tracking to preserve the quality of tracking as well as conserving the power usage. However, in reality, t_o in (14) may increase to infinity or decrease to zero. Thus, appropriate upper and lower bounds have to be placed on t_o .

The worst situation is that the target leaves along the radius direction with maximum acceleration. Intuitively, the upper bound can be given by

$$t_u = \frac{-V_i + \sqrt{V_i^2 + 2\alpha_{max}(r - D)}}{\alpha_{max}}. \quad (17)$$

The upper bound t_u determines the largest time interval for sensing. It indicates that the target under tracking is not able to move out of the communication range of S_j within time t_u . It seems that $t_u = 0$ when $r = D$ in the equation (17). This means the target is very close to the sensing edge. A continually sensing is not possible for a real sensor, so that a lower bound is introduced. The lower bound t_l can be defined based on the time cycle for a sensor to finish the process of measuring. Therefore t_l is usually hardware dependent. Thus, t_o can be expressed as

$$t_o = \begin{cases} t_u & t_o \geq t_u \\ t_l & t_o \leq t_l \\ (r - D(M_i, S_j))/V_i & \text{others} \end{cases}. \quad (18)$$

In reality, the $r - D(M_i, S_j)$ might be quite small when the target is near the sensing limit of the sensor, so that the t_l is relative large. Under such a scenario, the possibility of losing the target will be high. To enhance the

quality of tracking, S_i should be able to activate another sensor node with a smaller distance to the target. This evaluation function can be expressed by

$$S_i = \begin{cases} S_i & \text{if } D(M_j, S_i) \leq D(M_j, S_k) \\ S_k & \text{if } D(M_j, S_i) > D(M_j, S_k) \end{cases} \quad (19)$$

Based on the sensor hardware platform and the need of the application, the heuristic function represented by (18) and (19) can be employed.

4.3 The Mesh

If the selected sensor can not detect the target, we call it prediction failure. As discussed in Section I, prediction failure cannot be ignored due to blind areas in the sensing field, the functional failure of sensors, or the unpredictable behaviors of the target. We have developed a mesh approach to help the system recover efficiently from prediction failure, as shown in Fig. 5. Assume that the target is not detected, according to our best knowledge of the nature of the moving target, given the speed V_i at time t , it is not possible for the target to move out of the region C called the *mesh region*, which is the large circle area in Fig. 5, within time period t_o . Notice that C is a closed area with radius μ which is given by

$$\mu = D + L_{max} \quad (20)$$

where D is the spatial distance between sensor S_i and the target under tracking; and L_{max} is given by

$$L_{max} = V_{max} \times t_o \quad (21)$$

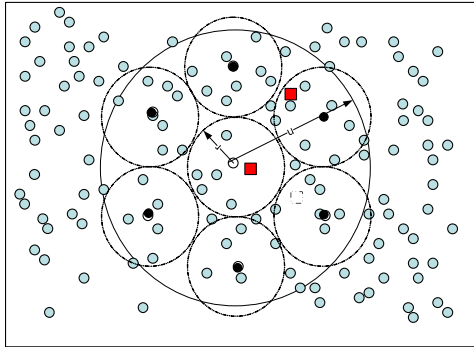


FIGURE 5

The mesh process. The black dots represent the virtual sensor nodes and the SC node activates the closest nodes to these virtual nodes to reallocate the target.

Therefore, if S_i activates several sensors so that the entire area of C is almost fully covered, the target will be detected. This is called the *mesh* process. In order to save power during this process, we introduce a concept of *virtual sensor nodes* which are indicated by the black dots in Fig.5. Each of these virtual sensor nodes has the same sensing scope r as the real sensor node S_i . Thus, we can conclude that the mesh area C can be almost fully covered if there are real active nodes which overlap with these virtual sensor nodes. However, since the real sensor nodes are randomly deployed, it is possible that there will be no physical sensor at the position of the virtual node. Therefore, it is intuitively reasonable to activate the sensors which are closest to the virtual nodes. While some areas may not be covered, the efficacy of the mesh process can be enhanced by activating more virtual nodes until the target gets detected.

Obviously, the mesh method requires much more power consumption than the prediction method. Therefore, it is used as a backup to the prediction method when the WSN loses the target.

5 SIMULATION

In this section, we present the results of several simulations to evaluate the performance of the PaM algorithm. These simulations have been done using Matlab. Specific emphasis is placed on the following aspects of the algorithm:

- *Power consumption per time unit.* We denote P_s as the criteria to evaluate the power efficacy of the PaM algorithm for each sensor. P_s is given by

$$P_s = \frac{E_o}{t_s - t_e} \quad (22)$$

where E_o can be obtained from Eq.(8), t_s is the start time, t_e is the end time.

- *Quality of tracking.* The quality of tracking is represented by the missing rate which is the ratio of the number of failures of locating an target based on prediction. Obviously, the smaller the missing rate is, the better the quality of tracking is. If we have to use the “mesh” method to recover a lost target, we consider it as a miss of the prediction method.
- *Adaptability of the PaM algorithm.* We evaluate the adaptability of the PaM algorithm for diverse targets with various random motion patterns.

5.1 The Simulation Setup

In our simulations, the sensing field was 1000 meters by 1000 meters. For all simulation results presented in this section, distances were measured in units of meters. Special sentinel sensors were placed at the edge of the field to activate the nearest regular sensors. The energy cost of sentinel sensors were

not counted. A total of 2500 regular sensors were uniformly and randomly placed in the sensing field. Each sensor had a detection radius of 30 meters and a communication radius of 90 meters [15]. The power levels of each working state of the sensors are defined in Table. 2 and the sensors were assumed to be operating at 3.0 volts [15]. Human beings and vehicles were the two kinds of targets studied in our simulations. And it was assumed that they moved randomly in this two-dimensional sensing field. The moving speed V varied between 0 and V_{max} with an acceleration α varying between 0 and α_{max} . The acceleration direction change varied between 0 and 360° . The values of these parameters are shown in Table 3.

5.2 Simulation Procedures

Two targets (a human and a vehicle) sequentially entered the sensing field from the position (100,0) with initial moving direction $\theta = 0$. When an target enters the sensing area, some sensors at the edge of the sensing field work as guard sensors (in active status) to get the first information of the target (position and velocity). With this information, the simulator is initialized. The remaining sensors inside the boundary region were kept in the sleep mode. Fig.6 and Fig.7 are the snapshots of the simulations for tracking a vehicle and a human being, respectively. These figures show the layout of the sensing field. The dot points represent the sensors and the line represents the moving track of the target. The motion dynamics can be found in Table 3.

5.3 Simulation Results

Power Consumption and Tracking Quality

In order to evaluate the power consumption performance of the PaM algorithm, we ran two simulations, tracking a human and a vehicle, respectively.

	Idle	Sensing	Receiving	Sending	PowerDown
Active	600 μA	4 mA	7 mA	9 mA	0
Sleep	0	0	0	0	10 μA

TABLE 2
Energy consumption parameters of the sensor nodes

	Object 1: Human beings	Object 2: Vehicle
V_{max} (m/s)	2.0	40.0
α_{max} (m/s ²)	6.0	10.0

TABLE 3
Moving parameters of the tracked targets

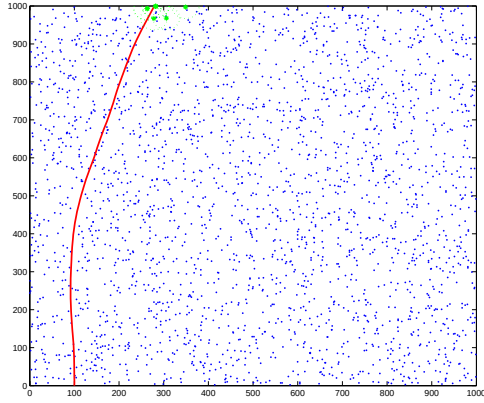


FIGURE 6
Snapshots of the simulation for tracking a vehicle.

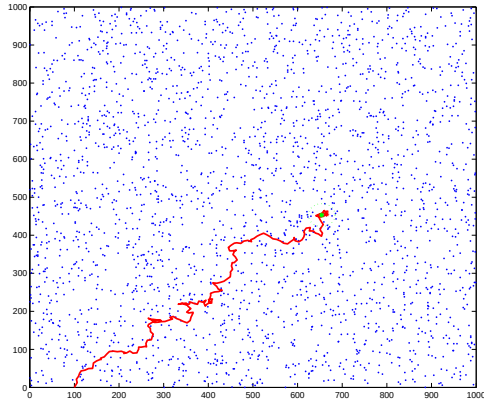


FIGURE 7
Snapshots of the simulation for tracking a person.

In the first simulation, we assumed that all sensors were in the idle status until an event triggered them for sensing. Then the total energy consumed per time unit was calculated based on Eq. (7). In the second simulation, we implemented the PaM algorithm. All sensors were initially in the power down status until they received the activation message. The comparison results are shown in Table 4.

	Human	Vehicle
Power consumption in simulation 1 (mW)	621.7040	6125.8000
Missing rate in simulation 1	0.0000	0.0000
Power consumption in simulation 2 (mW)	11.1403	27.0637
Missing rate in simulation 2	0.0370	0.0244

TABLE 4
Power Usage and QoS Evaluation

Observe that while the PaM algorithm may result in a temporary loss of the target, the overall quality of tracking is still high. As we mentioned before, this target losing will be recovered by the “mesh” method. Moreover, making use of the PaM algorithm dramatically reduces power usage. Thus, we can conclude that the PaM algorithm can preserve the quality of tracking and enhance the lifetime of the sensing system as well.

Adaptability

In [29], several prediction based tracking algorithms were proposed. However, the prediction time interval was fixed subject to the moving speed of the target. Thus, the quality of tracking and power consumption will vary in diverse scenarios of tracking different kinds of targets. To illustrate the excellent performance of the PaM algorithm, we ran another two simulations to evaluate its adaptability. The simulations were run for tracking a human and a vehicle using the algorithm proposed in [29] with a parameter Δt which denotes the prediction time interval. In both simulations, the power consumption and the missing rate were recorded. Here the missing rate is used to express the accuracy of the prediction. If the prediction fails, some recovery mechanisms will be performed to retrieve target. the The larger the missing rate, the worse the quality of tracking.

The results are shown in Table 6 and Table 5. Fig.8 and Fig.9 illustrate the performance evaluation of tracking a human. When tracking a human, power consumption using the PaM algorithm almost equals the optimal solution of using the nonadaptive prediction tracking algorithm while the PaM algorithm provides better tracking quality. Thus, the overall performance of the PaM algorithm is much better than the non-adaptive prediction algorithm with respect to power consumption and the quality of tracking. Experiments with respect to velocity (e.g. vehicles) should have been evaluated. Fig.10 and Fig.11 illustrate the performance evaluation of tracking a vehicle. When tracking a vehicle, the power efficacy performance of the PaM algorithm is better than the nonadaptive tracking algorithm as the prediction interval in the non-adaptive approach is set to a value between 0.1 second and 1.4 second.

Tracking interval (second)	Missing rate	Power consumption (mW)
1.0000	0.0000	18.2400
2.0000	0.0000	14.1200
4.2000	0.0291	12.2118
6.4000	0.0537	11.3340
8.4000	0.0720	11.0999
10.8000	0.4377	11.5620
12.8000	0.3667	10.9919
14.8000	0.5044	10.9763
16.8000	0.7117	11.2660
19.4000	0.8691	11.1477
22.2500	0.9778	10.8872
24.6000	1.0000	11.1807
Adaptive	0.0244	11.1403

TABLE 5
The Performance Evaluation of Tracking A Human

Tracking interval (second)	Missing rate	Power consumption (mW)
0.1183	0.0028	92.4000
0.2650	0.0181	51.5640
0.3750	0.0699	35.4953
0.4850	0.1255	31.2325
0.5950	0.2211	30.3723
0.7050	0.5315	28.5106
0.8333	0.6182	32.7797
0.9433	0.8168	30.7463
1.0533	0.8556	32.5254
1.1633	1.0000	31.6527
1.2733	0.9861	32.4714
1.3833	1.0000	30.9975
Adaptive	0.0370	27.0637

TABLE 6
The Performance Evaluation of Tracking A Vehicle

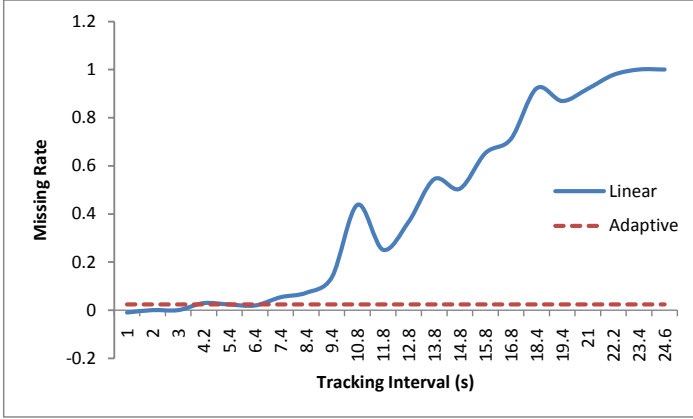


FIGURE 8
The missing rate of tracking a human being.

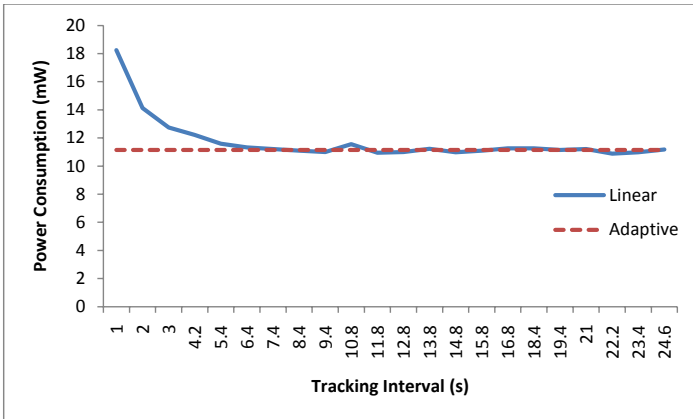


FIGURE 9
The power consumption of tracking a human being.

The PaM algorithm has much better quality of tracking as well. In particular in the case of humans, the savings are approximately 8.7%. Savings are increased to 37.8% in the case of vehicles. The reason is the conventional methods lost the high speed targets more often due to the un-adaptive natural. The simulation results shows that the PaM algorithm is suitable for tracking diverse kinds of targets with diverse moving patterns.

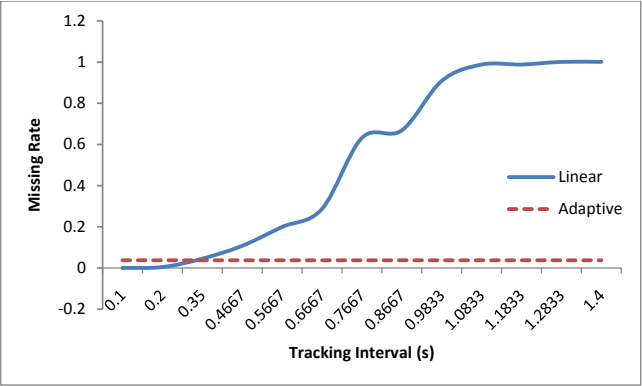


FIGURE 10
The missing rate of tracking a vehicle.

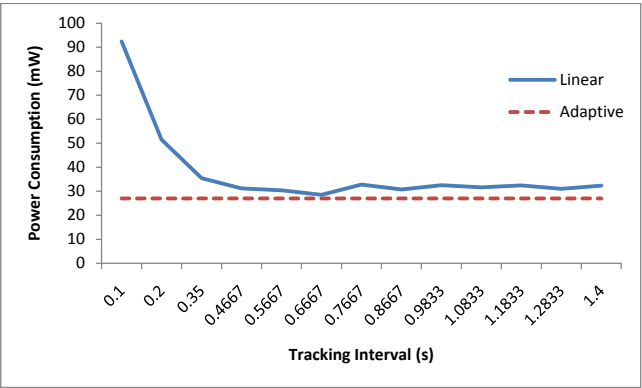


FIGURE 11
The power consumption of tracking a vehicle.

6 CONCLUSIONS

In this paper the PaM algorithm is proposed as a practical approach for large scale surveillance applications. The PaM algorithm uses a novel prediction model and a mesh process to monitor the movement of the targets. As an application layer algorithm, this method is more suitable to a wireless sensor node than existing works [33, 25] due to the adaptation, fast convergence and light weight. It has proven successful in several ways. First, it provides more space for implementation since it is built based on an randomly deployed distributed network architecture, thereby ensuring flexibility. Second, making

use of the PaM algorithm, the quality of tracking is preserved and the life time of the sensing system is dramatically enhanced as well. Third, the PaM algorithm proved to be adaptive for tracking diverse kinds of targets with various random movement patterns.

Our future work will focus on building motion models of various targets. The PaM algorithm can be made more efficient if it is provided with a precise motion model for the specific target. In other words, the accuracy of prediction can be further improved. Another direction to look at is to incorporate the identification approach, which is used to classify multiple targets in the sensing field into the PaM algorithm. Thus, multiple levels of fidelity can be provided according to the needs of applications.

REFERENCES

- [1] A. Arora, P. Dutta, and et. al. (2004). A line in the sand: a wireless sensor network for target detection classification and tracking. *Computer Networks*, 46(5):605–634.
- [2] S. Bhatti and J. Xu. (2009). Survey of target tracking protocols using wireless sensor network. In *Fifth International Conference on Wireless and Mobile Communications*, pages 110–115.
- [3] W. Blair and T. Bar-Shalom. (January 1996). Tracking maneuvering targets with multiple sensors: Does more data always mean better estimates? *IEEE Trans. on Aerospace and Electronic Systems*, 32(1):450–456.
- [4] R.R. Brooks, P. Ramanathan, and A. M. Sayeed. (2003). Distributed target classification and tracking in sensor network. *Proc. of the IEEE*, 91(8):1163–1171.
- [5] I. Chatzigiannakis, T. Dimitriou, S. Nikolettseas, and P. Spirakis. (2004). A probabilistic forwarding protocol for efficient data propagation in sensor networks. In *Proceedings of 5th European Wireless Conference on Mobile and Wireless Systems beyond 3G (EW 2004)*, pages 344–350.
- [6] B. Chen, K. Jamieson, and H. Balakrishnan. (September 2004). Span: An energy efficient coordination algorithm for topology maintenance in ad hoc wireless network. *IEEE Trans. on Wireless Communications*, 4(5):1689–1701.
- [7] J. Chen, K. Cao, K. Li, and Y. Sun. (2009). Distributed sensor activation algorithm for target tracking with binary sensor networks. *Cluster Computing*.
- [8] T. Clouqueur, V. Phipatanasuphorn, and K. K. Saluja. (2002). Sensor deployment strategy of target detection. In *Proceedings of AACM WSNA'02*, pages 42–48.
- [9] S. Goel and T. Imielinski. (2001). Prediction-based monitoring in sensor networks taking lessons from mpeg. *ACM SIGCOMM Computer Communication Review*, 31(5):82–98.
- [10] C. Gui and P. Mohapatra. (2004). Power conservation and quality of surveillance in target tracking sensor networks. In *Proceedings of The 9th International Conference on Mobile Computing and Networking (MobiCom) 2004*, pages 129–143.
- [11] Z. Guo, M. Zhou, and L. Zakrevski. (2005). Optimal tracking interval for predictive tracking in wireless sensor network. *IEEE Communications Letters*, 9(9):805–807.
- [12] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. (2003). Range-free localization schemes in large scale sensor networks. In *Proceedings of The 9th International Conference on Mobile Computing and Networking (MobiCom) 2003*, pages 81–95.
- [13] T. He, J. A. Stankovic, C. Liu, and T. F. Abdelzaher. (2003). Speed: A stateless protocol for real-time communication in sensor networks. In *Proceedings of 23rd International Conference on Distributed Computing Systems (ICDCS 2003)*, pages 46–55.

- [14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of 33rd IEEE Hawaii International Conference on System Sciences (HICSS 2000)*, page 8020.
- [15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. (2000). System architecture directions for network sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, pages 93–104.
- [16] L. Hu and D. Evans. (2004). Localization for mobile sensor networks. In *Proceedings of The Ninth International Conference on Mobile Computing and Networking (MobiCom) 2004*, pages 45–57.
- [17] C. Intanagonwiwat, R. Govindan, and D. Estrin. (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of 6th ACM/IEEE Annual International Conference on Mobile Computing (MOBICOM 2000)*, pages 56–67.
- [18] X. Kuang, R. Feng, and H. Shao. (2008). A lightweight target-tracking scheme using wireless sensor network. *Measurement Science and Technology*, 19:1–7.
- [19] H. T. Kung and D. Vlah. (2003). Efficient location tracking using sensor networks. In *Proceedings of 2003 IEEE Wireless Communications and Networking Conference*, pages 1954–1961.
- [20] M. E. Liggins, C. Chong, and et. al. (1997). Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107.
- [21] J. Liu, P. Cheung, L. Guibas, and F. Zhao. (2002). A dual-space approach to tracking and sensor management in wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks*, pages 131–139.
- [22] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. (2005). Worst and best-case coverage in sensor networks. *IEEE Trans. on Mobile Computing*, 4(1):84–92.
- [23] S. Pattem, S. Poduri, and B. Krishnamachari. (2003). Energy-quality tradeoff for target tracking in wireless sensor networks. In *Proceedings of 2nd Workshop on Information Processing in Sensor Networks, ISPN*, pages 32–46.
- [24] J. Peng, L. Yang, F. Momen, and J. W. Rozenblit. (2005). Level-of-detail based modeling of object movement in tracking and surveillance system. In *Proceedings of 2005 Winter Simulation Conference*, pages 461–466.
- [25] H. Rad, B. Abolhassani, and M. Abdizadeh. (2009). A new adaptive prediction-based tracking scheme for wireless sensor networks. In *Annual Conference on Communication Networks and Services Research*, pages 335–341.
- [26] D. Vaidya, J. Peng, L. Yang, and J. W. Rozenblit. (2005). A framework for sensor management in wireless and heterogeneous sensor network. In *Proceedings of 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, pages 155–162.
- [27] W. Xiao, S. Zhang, J. Lin, and C. Tham. (2010). Energy-efficient adaptive sensor scheduling for target tracking in wireless sensor networks. *Journal of Control Theory and Applications*, 8(1):86–92.
- [28] Y. Xu and W. Lee. (2003). On localized prediction for power efficient object tracking in sensor networks. In *Proceedings of the 1st International Workshop on Mobile Distributed Computing (MDC)*, pages 434–439.
- [29] Y. Xu, J. Winter, and W. Lee. (2004). Prediction-based strategies for energy saving in object tracking sensor networks. In *Proceedings of IEEE International Conference on Mobile Data Management, 2004*, pages 346–357.
- [30] Z. Xu and L. Rui. (2009). A self-organizing bearings-only target tracking algorithm in wireless sensor network. *Chinese Journal of Aeronautics*, 22(6):627–636.

- [31] H. Yang and B. Sikdar. (2003). A protocol for tracking mobile targets using sensor networks. In *Proceedings of the 1st IEEE Workshop on Sensor Network Protocols and Applications*, pages 71–81.
- [32] L. Yang, J. W. Rozenblit, C. Feng, and H. Qiao. (2006). A multi-modality framework for energy efficient tracking in large scale wireless sensor networks. In *Proceedings of IEEE International Conference on Networking, System and Control, 2006*, pages 916–921.
- [33] J. Yick, B. Mukherjee, and D. Ghosal. (2005). Analysis of a prediction-based mobility adaptive tracking algorithm. In *Proceedings of the 2nd International Conference on Broadband Networks*, pages 753–760.
- [34] X. Yu, K. Niyogi, S. Nikolettseas, S. Mehrotra, and N. Venkatasubramanian. (2004). Adaptive target tracking in sensor networks. In *Proceedings of 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDs'04)*, pages 344–350.
- [35] W. Zhang and G. Gao. (2004). Dynamic convoy tree-based collaboration for target tracking in sensor networks. *IEEE Trans. on Wireless Communications*, 4(5):1689–1701.
- [36] F. Zhao, M. Kung, and D. Vlah. (2002). Information-driven dynamic sensor collaboration for tracking applications. *Signal Processing Magazine*, 19(2):61–72.
- [37] Y. Zou and K. Chakrabarty. (2004). Sensor deployment and target localization in distributed sensor networks. *ACM Trans. on Embedded Computing Systems*, 3(1):61–91.