Adaptive Tracking in Distributed Wireless Sensor Networks

Lizhi Yang, Chuan Feng, Jerzy W. Rozenblit, Haiyan Qiao
The University of Arizona
Electrical and Computer Engineering Department
Tucson, AZ 85721-0104, USA
{lyang, fengc, jr, haiyanq}@ece.arizona.edu

Abstract

We study the problem of tracking moving objects using distributed Wireless Sensor Networks (WSNs) in which sensors are deployed randomly. Due to the uncertainty and unpredictability of real-world objects' motion, the tracking algorithm is needed to adapt to real-time changes of velocities and directions of a moving target. Moreover, the energy consumption of the tracking algorithm has to be considered because of the inherent limitations of wireless sensors. In this paper, we proposed an energy efficient tracking algorithm, called Predict-and-Mesh (PaM) that is well suited for pervasively monitoring various kinds of objects with random movement patterns. PaM is a distributed algorithm consisting of two prediction models: n-step prediction and collaborative prediction, and a predication failure recovery process called mesh. The simulation results show that the PaM algorithm is robust against diverse motion changes and has the excellent performance.

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 the environment in various modalities, process the information, and disseminate data wirelessly. Therefore, if the ability of the WSN is suitably harnessed, it is envisioned that the WSN can 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 [11].

The design of the 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 the application of tracking real-world objects, e.g., tracking moving targets using huge numbers of sensors with a small detection range. Typically, sensors are distributed randomly, in a large scale region to be monitored. The possible scenarios can be border control, battle field surveillance, traffic flow measuring, or animal monitoring, etc. In developing such a network, tracking algorithm design, however, is a complicated problem due to sensors' limitations in computation, sensing, memory, battery life and network bandwidth [10]. All of these limitations impose the constraints on the efficacy of tracking. Therefore, the classical tracking mechanisms are not sufficient for the WSN by themselves.

In general, the tracking scheme is fixed when considering the problem of tracking a target using the distributed WSN. However, in many tracking applications, the motion characteristics of the object under tracking may vary with time due to the uncertainty and unpredictability in the target motion model [1]. Moveover, different types of objects may have different kinds of motion characteristics. For example, a tank has a much larger maximum velocity than a human solider. On the other hand, a human solider, given a moving velocity, can perform turn much quicker than a tank. Such variabilities pose a significant problem when designing a tracking algorithm. Thus, the design priority will be given to both the efficiency and adaptability of the network utilization.

In this paper, we propose an energy efficient tracking algorithm called Predict-and-Mesh (PaM) that is well suitable for pervasively monitoring movements of various kinds of objects with random movement patterns. PaM is a distributed algorithm consisting of two prediction models: n-step prediction and collaborative prediction, and a predication failure recovery process called *mesh*. We show that the PaM algorithm is robust against diverse motion changes of a target under tracking.

Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06)
0-7695-2546-6/06 $20.00 © 2006 IEEE

The organization of the rest of the paper is as follows. In Section 2, the tracking algorithm design challenges are discussed with respect to the efficacy and adaptability. Section 3 defines several useful preliminary models and gives details of the PaM algorithm. In Section 4, we present the simulation and its results. Section 5 concludes the paper and outlines directions for future work.

2. Tracking Algorithm Design Challenges

Given a large scale area, the tracking algorithm must be able to scale to a large number of entities to cover the entire area. Generally, the deployed WSN is decentralized. In other words, the fixed superior/subordinate relationship does not exist in such a network. Although a centralized architecture is theoretically optimal and also conceptually simple [7], it is not suitable to a large scale environment because of the limited communication bandwidth and power supplies of wireless sensors. On the other hand, the decentralized architecture poses opportunities for achieving high survivability since there is no centralized point of failure that will result in an entire network failure.

For certain applications (e.g., military applications), it is desired that the sensors are dropped from an aircraft or by other means into the hostile environment without any further adjustment. Nevertheless, such random deployment strategies may lead to severe coverage problems subject to sensor communication and detection constraints [2]. Thus, the designed algorithm should be resilient to "blind" areas, i.e., the regions that are partially or fully uncovered by any sensor. In addition, while the position of the sensor is usually fixed after deployment, the network topology and membership may frequently change due to a functional failure, physical damage, lack of power or new sensors just joining the network. In other words, the WSN for tracking objects may be ad hoc.

Limited power supply is another challenge which is inherent from the nature of the sensors. In general, the sensors are equipped with limited and irreplaceable power sources. As a commonly used strategy to reduce the power usage of the sensing system, turning off unnecessary sensors can effectively enhance the lifetime of the entire system since the energy consumption increases significantly during the periods of activity [13]. However, for randomly deployed sensor networks there is a need for an activation mechanism to accurately decide which sensors are necessary to be activated for the quality of tracking. Moreover, due to unpredictable behaviors of objects, e.g., random moving speeds and directions, the tracking algorithm is needed to adapt to such changes.

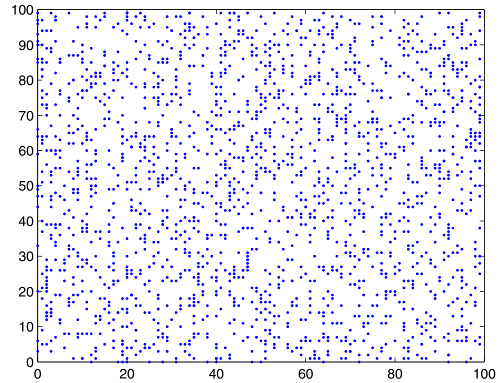

Figure 1. The Surveillance Zone Example

3 The PaM Algorithm

Assume that the wireless sensors are dropped randomly (e.g., with a uniform distribution) to cover a two dimensional, large scale region. One example of such a surveillance zone is depicted in Fig. 1, in which each dot represents a single sensor node. The sensors can collaborate to track objects. The key idea of this paper is that the lifetime of such a sensing system can be extended by using a set of prediction based activation mechanisms. In order to conserve the power usage, we have developed the PaM algorithm to control activities of the sensors (active/sleep) for the tracking purpose. The word "predict" denotes that the system is able to predict the next location of the object, by using the time series trajectory of its path. Given that predicted location, appropriate sensors in that region will be activated to sense the target. Making use of this mechanism, initially, most of the sensor nodes will be in the sleep status in order to conserve the powers until the activity is triggered by mobile targets. Once the target is detected, the active nodes, which are not only the sensing centers but also the coordinators (SCs), will activate another set of appropriate sensors as the next SCs. Subsequently, previous SCs will change to the sleep status (non-sensing and power down). It is clear that the fewer nodes that are active, the less power that is consumed. However, the prediction may fail due to the blind areas, the uncertainties of object behaviors and node failures. In this paper, we propose a mesh process in order to help the sensing system recover from potential prediction failures.

3.1 Preliminary Models

In this section, we describe underline assumptions and preliminary models.

3.1.1 The Power Usage Model

Given n sensors at time t, there are two possible states for each sensor, which are active and sleep, respectively [9]. Thus, a tracking system is modeled as a 4-tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{N}, \mathcal{F}, \mathcal{K}) \tag{1}$$

where $\mathcal{S}$ is a WSN, $\mathcal{N}$ represents active sensors, and $\mathcal{F}$ denotes sensors in the sleep mode. Let $\mathcal{K}$ be a set of events represented by a function as

$$\mathcal{K} : \mathcal{S} \rightarrow \mathcal{N} \times \mathcal{F}. \tag{2}$$

Note that $\mathcal{S}$ is a dynamic system since at any given time t there may be one or more sensors which change their states.

Typically, for each sensor, the power consumption will vary under various states. 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 which shuts everything but the necessary circuits for waking up. Thus, any event f in the sensing field may result in mode transition for k sensors as

$$f : \int \rightarrow \mathcal{I} \times \mathcal{E} \times \mathcal{R} \times \mathcal{T} \times \mathcal{D} \tag{3}$$

where $\int$ denotes any single sensor, $\mathcal{I}$ represents a sensor in the idle mode, $\mathcal{E}$ represents the sensor in the sensing mode, $\mathcal{D}$ indicates that the sensor is in the power down mode, $\mathcal{T}$ and $\mathcal{R}$ denote the sensor in the transmitting mode and the receiving mode, respectively. The sensor power consumption modes are summarized in Table.1.

If we denote $W_\mathcal{I}$, $W_\mathcal{E}$, $W_\mathcal{D}$, $W_\mathcal{R}$ and $W_\mathcal{T}$ as the power usage of a sensor under different working modes, the power consumption at any given time t for this sensor is given by

$$p = W_\mathcal{I} \cdot t_\mathcal{I} + W_\mathcal{E} \cdot t_\mathcal{E} + B_r \cdot W_\mathcal{R} \cdot t_\mathcal{R} + B_t \cdot W_\mathcal{T} \cdot t_\mathcal{T} + W_\mathcal{D} \cdot t_\mathcal{D} \tag{4}$$

where $t_\mathcal{I}$ is the idle period of the sensor, $t_\mathcal{E}$ denotes the time required for the sensor to optimally estimate the position of an object, $t_\mathcal{D}$ is the power down period of the sensor, $t_\mathcal{T}$ is the time required for the sensor to send out a unit packet, $t_\mathcal{R}$ 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, the sensor detection is imprecise, therefore a Kalman filter is often employed to estimate the position, velocity and acceleration of a target [1]. The dynamics model of a target under tracking can be defined as

$$x_k = Ax_{k-1} + w_k \tag{5}$$

where $x \in R^n$ is the state vector of the target and w_k is the process noise (e.g., unknown acceleration). A measurement model z_k is given by

Table 1. Power Usage Modes

	Idle	Sensing	Receiving	Transmitting	PowerDown
Active	Y	Y	Y	Y	N
Sleep	N	N	N	N	Y

$$z_k = Hx_k + v_k \tag{6}$$

where v_k is the measurement noise. In this work, we assume that w_k and v_k are white Gaussian noise. Thus, for a specific sensor, a time period $t_\mathcal{I}$ is required to get optimal estimation of the position of a target within its sensing range. Given a system $\mathcal{S}$ with n sensors, the power consumption matrix $\mathcal{P}$ can be represented by

$$\mathcal{P} = \begin{pmatrix} t_{1\mathcal{I}} & t_{1\mathcal{E}} & B_r \cdot t_{1\mathcal{R}} & B_t \cdot t_{1\mathcal{T}} & t_{1\mathcal{D}} \\ t_{2\mathcal{I}} & t_{2\mathcal{E}} & B_r \cdot t_{2\mathcal{R}} & B_t \cdot t_{2\mathcal{T}} & t_{2\mathcal{D}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n\mathcal{I}} & t_{n\mathcal{E}} & B_r \cdot t_{n\mathcal{R}} & B_t \cdot t_{n\mathcal{T}} & t_{n\mathcal{D}} \end{pmatrix} \times \begin{pmatrix} W_\mathcal{I} \\ W_\mathcal{R} \\ W_\mathcal{E} \\ W_\mathcal{T} \\ W_\mathcal{D} \end{pmatrix} \tag{7}$$

where $\mathcal{P}$ is a vector of which each entry represents the power usage of a single sensor.

Thus, assuming an object under tracking has been in the sensing field for the time interval $t_e - t_s$, the power consumed by $\mathcal{S}$ for tracking this target can be formulated as

$$\mathcal{P}_\mathcal{S} = \sum_{t_s}^{t_e} \sum_{i=1}^{n} p_i \tag{8}$$

where $p_i, i = 1, 2, ..., n$, denotes entries in a vector $\mathcal{P}$, t_s represents the time when the object enters the sensing field and t_e represents the end time when the object leaves the sensing field. Notice that $W_\mathcal{I}$, $W_\mathcal{E}$, $W_\mathcal{D}$, $W_\mathcal{R}$ and $W_\mathcal{T}$ highly depend on the hardware platform of the sensor node. Given a specific hardware platform, e.g., Berkeley MICA motes [5], one second of the sleep mode can save enough power for sending more than 70 packets, or performing 70K operations [8]. Therefore, the power consumption can be conserved by applying an efficient tracking algorithm through which most of sensors will be in the sleep status.

3.1.2 The Sensing and Communication Model

We assume that the sensors used to detect objects have 360^0 degree sensing scope r. Examples of this kind of sensors are acoustic, seismic, and electromagnetic sensors. In general, these sensors are physically small and equipped with limited power supplies, thus r is confined to be small. Within the detectable distance r, an active sensor node $\int_i$ is able to estimate its distance D and orientation θ to an object [8].

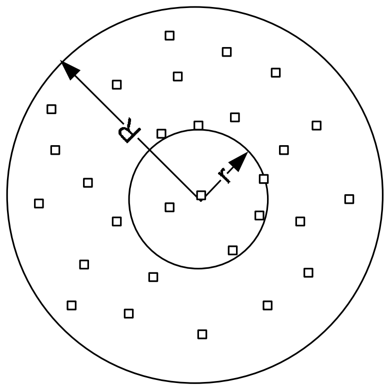

Figure 2. Sensing and Communication Model

In order to simplify the problem, we assume that the distance and the orientation information can be accurately estimated in $t_{\mathcal{E}}$ seconds. Thus, the relative position vector R of the target to sensor $\int_i$ 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 is out of the sensing scope of $\int_i$, another sensor node $\int_j$ needs to be activated to sense the target. In order to conserve the power usage, $\int_i$ will be switched to the sleep state after it gets an acknowledgement from sensor $\int_j$. Thus, the tracking task is carried out by sequentially activating and shutting the sensors located along the track of the moving object.

Notice that the sensing system $\mathcal{S}$ is distributed, thus there is information exchange among sensors. Typically, each sensor has a wireless communication scope R which is relatively larger than r (Fig. 2). Thus, the sensor node $\int_i$ can communicate with any other sensor node $\int_j$ within the circle area with radius R, centered by $\int_i$. In order to accurately communicate with $\int_j$ in a specific position, the sensing node $\int_i$ has to know its own position and the position of $\int_j$. Since the position of each sensor node $\int_i$ is not predetermined in a randomly deployed sensor network, there is a need for network localization method. We assume that the sensing system $\mathcal{S}$ consists of a subset $\mathcal{A}$ in which all of the sensor nodes are equipped with GPS receivers that have knowledge of their own locations [4][6].

3.2 Prediction Models

Assume that there are N wireless sensors (e.g., acoustic sensors), which are initially in the sleep mode, deployed in the sensing field. A mobile object traverses through this sensing field with speed $\mathcal{V}$. Given time t, the object is detected by an active sensor $\int_i$. Then, after Δt seconds, the object may be out of the detection range r of $\int_i$. Since the target may move randomly, it is necessary to estimate the possibility of this scenario to avoid losing the target. Due to the limited power of the sensor, the goal of the estimation is to activate as few sensors as possible to save power. Obviously, a good prediction mechanism should also maintain high quality and fidelity of tracking.

However, the prediction scheme design is complex due to the unpredictable behaviors of the target. Based on the sensing and communication model, the prediction model may vary given a target with a different speed $\mathcal{V}$. For example, a target M_i travelling at speed $\mathcal{V}_i$ [1] is detected by a sensor $\int_i$ at time t at position (x_1, x_2) with the distance $D(M_i, \int_i)$ ($D(M_i, \int_i) \leq r$) and the orientation $\theta((M_i, \int_i))$ to $\int_i$. Intuitively, it will take at least t_o seconds for M_i to move out of the sensing range r of $\int_i$. We denote t_o as *escape period* that is given by

$$t_o = (r - D(M_i, \textstyle\int_i))/\mathcal{V}_i. \quad (10)$$

It is clear that the object can be detected again by sensor $\int_i$ within time t_o if $\mathcal{V}_i$ is not increased. Otherwise, $\mathcal{V}_i$ has to be updated as $\mathcal{V}_j$ as the target moves out of range of $\int_i$ and the sensor $\int_j$ has to be activated to detect this target. The $\mathcal{V}_j$ can be estimated by

$$\mathcal{V}_i = \mathcal{V}_o + \alpha_i \times t_o \quad (11)$$

where α_i denotes the possible acceleration of a specific object which uniformly falls in $[0, \alpha_{max}]$. Thus, $\mathcal{V}_j$ falls in a speed interval $[\mathcal{V}_i, \mathcal{V}_i + \alpha_i \times t_o]$. Notice that $(\mathcal{V}_i + \alpha_i \times t_o) \leq \mathcal{V}_{max}$, where $\mathcal{V}_{max}$ denotes the maximum speed of the target.

Assume that at time $t + t_o + t_{\mathcal{E}}$ the sensor $\int_i$ detects the target at position $(\hat{x_1}, \hat{y_1})$ so that there are two pieces of position information with respect to target M_i in the knowledge base of $\int_i$. Therefore, the speed $\mathcal{V}_i$ can be accurately calculated by

$$\mathcal{V}_i = \frac{\sqrt{(\hat{x_1} - x_1)^2 + (\hat{y_1} - y_1)^2}}{t_o + t_{\mathcal{E}}}. \quad (12)$$

Note that the sensor $\int_i$ will keep updating $\mathcal{V}_i$ until the target moves out of its sensing range. In light of this, we develop the following two prediction mechanisms for diverse tracking scenarios.

- *The collaborative prediction model.* Fig.3 illustrates the idea of the collaborative prediction model. Assume that the target is detected at position $R_1 = (x_1, y_1)$ at

[1] Notice that the speed $\mathcal{V}_i$ is recursively measured by active sensor nodes. The details of this measurement will be discussed later in this section.

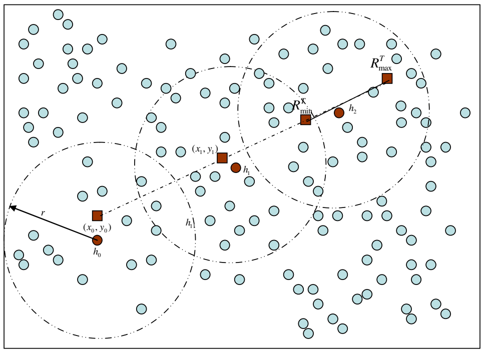

Figure 3. The Illustration of The Collaborative Prediction

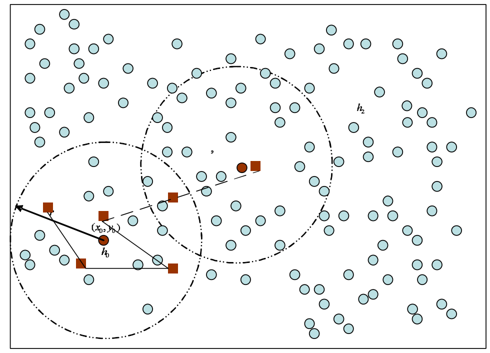

Figure 4. The Illustration of the N-step Prediction

time t and moves out of the sensing range of $\int_i$ after $t_o + t_{\mathcal{E}}$ seconds. Then the next possible position of the target can be estimated by

$$R_{max} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} \Im_{max} \times \cos\alpha \\ \Im_{max} \times \sin\alpha \end{pmatrix} \quad (13)$$

where $(\cos\alpha, \sin\alpha)$ can be calculated by

$$\begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} = \begin{pmatrix} \frac{x_1 - x_0}{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}} \\ \frac{y_1 - y_0}{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}} \end{pmatrix}, \quad (14)$$

and $\Im_{max}$ which denotes *maximum escaping distance*, given by

$$\Im_{max} = \mathcal{V}_i \times (t_o + t_{\mathcal{E}}) + \int_0^{t_o + t_{\mathcal{E}}} \alpha_{max} dt. \quad (15)$$

Note that the history position information (x_0, y_0) is provided by the previous active sensor node. Thus, this prediction is called collaborative prediction.

Assuming that there is only one piece of position information of target M_i in the knowledge base of the previous active sensor $\hat{\int}_i$ and the sensor $\int_i$ is awakened to sense this target. Given the position (x_1, y_1) from $\int_i$, the speed $\mathcal{V}_i$ can be calculated by

$$\mathcal{V}_i = \frac{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}}{\hat{t_o} + t_{\mathcal{E}}} \quad (16)$$

where $\hat{t_o}$ is the escape time for sensor $\hat{\int}_i$.

- *The n-step prediction model.* As shown in Fig.4, assume that the target is still moving within the sensing range of $\int_i$ after $t_o + t_{\mathcal{E}}$ seconds and therefore the target can be detected at the position $(\hat{x_1}, \hat{y_1})$ at time $t + t_o + t_{\mathcal{E}}$. The moving speed $\mathcal{V}_i$ can be obtained from (12). By recursively updating $\mathcal{V}_i$ and t_o, the moving track of the target can be obtained. Assume that after n steps the target moves out of the sensing range of $\int_i$, then the next moving position can be predicted by

$$R_c = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} \Im_{max} \times \cos\beta \\ \Im_{max} \times \sin\beta \end{pmatrix} \quad (17)$$

where $\Im_{max}$ is given by

$$\Im_{max} = \mathcal{V}_i \times (t_{on} + t_{\mathcal{E}}) + \int_0^{t_{on} + t_{\mathcal{E}}} \alpha_{max} dt. \quad (18)$$

where t_{on} is the latest estimated escape period after n-step estimation, and

$$\begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix} = \begin{pmatrix} \frac{x_n - x_{n-1}}{\sqrt{(x_n-x_{n-1})^2+(y_n-y_{n-1})^2}} \\ \frac{y_n - y_{n-1}}{\sqrt{(x_n-x_{n-1})^2+(y_n-y_{n-1})^2}} \end{pmatrix} \quad (19)$$

where (x_{n-1}, y_{n-1}) and (x_n, y_n) are the last two positions in the sensing range of $\int_i$.

3.2.1 Heuristic Functions

Making use of equation (10) and (12), the sensing system $\mathcal{S}$ is capable of refreshing the knowledge of the object under tracking to preserve the quality of tracking and conserve the power usage as well. However, in reality, t_o in (10) may increase to infinity or decrease to zero. Thus, appropriate upper and lower bounds have to be placed on t_o. Intuitively, the upper bound can be given by

$$\bar{t} = \sqrt{\frac{2(R - \mathcal{D})}{\alpha_{max}}}. \quad (20)$$

The upper bound $\bar{t}$ 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 $\int_i$ within time $\bar{t}$. The lower bound $\underline{t}$ can be defined by t_c which is the time cycle for a sensor to finish the process of measuring. Note that t_c is usually hardware dependent. Thus, t_o can be expressed as

$$t_o = \begin{cases} \bar{t} & t_o \geq \bar{t} \\ t_c & t_o \leq t_c \\ (r - D(M_i, \int_i))/\mathcal{V}_i & others \end{cases} . \quad (21)$$

In reality, the $R - \mathcal{D}$ might be quite small and t_c is relative large. Thus, the possibility of losing the target will be high under such a scenario. In order to enhance the quality of tracking, $\int_i$ should be able to activate another sensor node that has a smaller distance to the target. This evaluation function can be expressed by

$$\int_i = \begin{cases} \int_i & if \;\; D(M_i, \int_i) \leq D(M_i, \int_j) \\ \int_j & if \;\; D(M_i, \int_i) > D(M_i, \int_j) \end{cases} . \quad (22)$$

Based on the sensor hardware platform and the need of the application, the heuristic function represented by (21) and (22) can be employed, alternatively.

3.3 The Mesh

As discussed in Section I, the prediction failure cannot be ignored due to blind areas in the sensing field and unpredictable behaviors of the object. We have developed a mesh approach to helping the system recover from the prediction failure efficiently. As shown in Fig. 5, assume that the target is not detected, i.e., the prediction fails. According to our best knowledge of the nature of the moving object, given the speed $\mathcal{V}_i$ at time t, it is not possible for the target to move out of the region $\mathcal{C}$ called the *mesh region* within time period $t_o + t_{\mathcal{E}}$. Notice that $\mathcal{C}$ is a closed area with radius μ which is given by

$$\mu = \mathcal{D} + \Im_{max} \quad (23)$$

where $\mathcal{D}$ is the spatial distance between sensor $\int_i$ and the target under tracking; and $\Im_{max}$ can be obtained from equation (18). Notice that $\mu \leq R$. Thus, if $\int_i$ activates several sensors by which the entire area of $\mathcal{C}$ is almost fully covered, the target will be detected. This is called the *mesh* process. In order to save the power usage of 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 $\int_i$. Thus, we can conclude that the mesh area $\mathcal{C}$ can be almost fully covered if there are real active nodes which

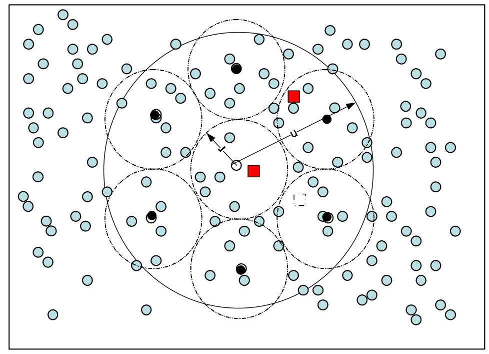

Figure 5. The Mesh Process

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 that can be found at the position of the virtual node. Therefore, it is intuitive to activate the closest sensors to the virtual nodes. While some areas may not be covered, the efficacy of the mesh process can be enhanced by rotating all the virtual nodes.

4 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. The specific emphasis is placed on the following aspects of the algorithm:

- *Power consumption per time unit*. We denote $E(\mathcal{P}_\mathcal{S})$ as the criteria to evaluate the power efficacy of the PaM algorithm. $E(\mathcal{P}_\mathcal{S})$ is given by

$$E(\mathcal{P}_\mathcal{S}) = \frac{\mathcal{P}_\mathcal{S}}{t_s - t_e} \quad (24)$$

  where $\mathcal{P}_\mathcal{S}$ can be obtained from Eq. (8).

- *The adaptability of the PaM algorithm*. We evaluate the adaptability of the PaM algorithm for diverse objects with various random motion patterns.

- *The quality of tracking*.

4.1 The Simulation Setup

In our simulations, the sensing field is 1000 meters by 1000 meters. For all simulation results presented in this section, distances are measured in units of meters. A total of 2500 sensors are uniformly and randomly placed in the sensing field. Each sensor has a detection radius of 30 meters and a communication radius of 90 meters. The power

Table 2. Values of the Parameters in Simulation

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

	Object 1: Human	Object 2: Vehicle
$\mathcal{V}_{max}$	2.0 m/s	40.0 m/s
α_{max}	6.0 m/s^2	10.0 m/s^2

consumption of each working mode of the sensors are based on Berkeley MICA motes operated at 3 volts. Human beings and vehicles are the two kinds of objects studied in our simulations. Assume that they move randomly in this two-dimensional sensing field. The moving speed $\mathcal{V}$ is varying between zero and $\mathcal{V}_{max}$ with an acceleration $\alpha \in [0, \alpha_{max}]$. The values of these paramters are shown in Table. 2.

4.2 Simulation Procedures

In our simulations, two objects (a human and a vehicle) sequentially enter the sensing field. When an object enters the sensing area, some sensors at the edge of the sensing field will 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 which are in the sleep mode use the PaM algorithm to monitor the movement of the target. Fig.6 and Fig.7 illustrate the snapshots of the simulations for tracking a human and a vehicle, 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 object. The motion dynamics can be found in Table.2.

4.3 Simulation Results

4.3.1 Power Consumption and Tracking Quality

In order to evaluate the power consumption performance of the PaM algorithm, we run two simulations. Again, both are run for tracking two different objects, a human and a vehicle, respectively. In the first simulation, we assume that all sensors will be in the idle status until an event triggers them for sensing. Then, the total energy consumed per time unit is calculated based on Eq. (7). In the second simulation, all sensors are initially in the power down status till they receive the activation message. The comparison results are shown in Table.3.

Observe that while the PaM algorithm may result in a temporary loss of the target, the overall quality of tracking is still high enough. Moreover, making use of the PaM algorithm can dramatically reduce the power usage. Thus,

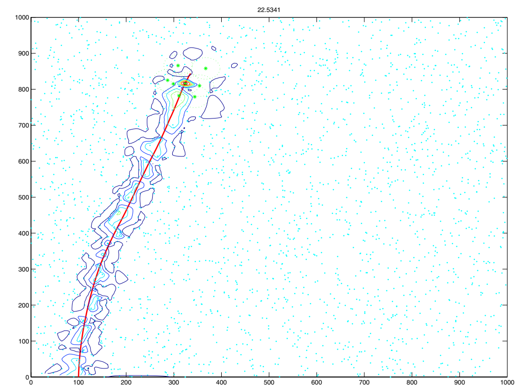

Figure 6. Snapshots of the Simulation for Tracking a Vehicle

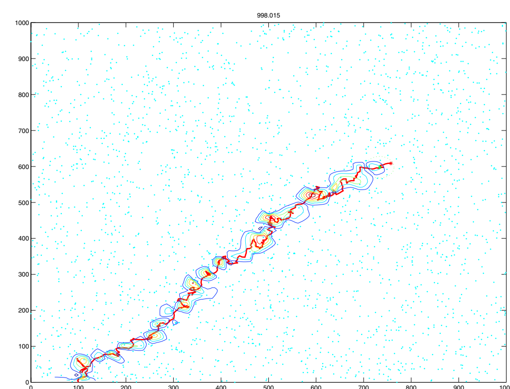

Figure 7. Snapshots of the Simulation for Tracking a Person

Table 3. Power Usage and QoS Evaluation

	Human	Vehicle
$E(\mathcal{P}_S)$ in Simulation 1	82.695 mW	207.25 mW
$E(\mathcal{P}_S)$ in Simulation 2	4661.46 mW	4660.24 mW
QoS in Simulation 1	100%	100%
QoS in Simulation 2	99%	95%

Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06)
0-7695-2546-6/06 $20.00 © 2006 IEEE

Table 4. The Adaptability Evaluation

$\triangle t = 0.2$	Human	Vehicle
$E(\mathcal{P}_S)$ in Simulation	387.67 mW	384.00 mW
Mesh Ratio	$< 1\%$	$< 1\%$
$\triangle t = 0.5$	Human	Vehicle
$E(\mathcal{P}_S)$ in Simulation	198.60 mW	225.99 mW
Mesh Ratio	$< 1\%$	$\approx 25\%$
$\triangle t = 1.0$	Human	Vehicle
$E(\mathcal{P}_S)$ in Simulation	126.80 mW	233.91 mW
Mesh Ratio	$< 1\%$	$\approx 75\%$
$Adaptive$	Human	Vehicle
$E(\mathcal{P}_S)$ in Simulation	82.695 mW	207.25 mW
Mesh Ratio	$< 1\%$	$\approx 5\%$

we can conclude that the PaM algorithm can preserve the quality of tracking and enhance the lifetime of the sensing system as well.

4.3.2 The Adaptability

In [11], [3] and [12], a similar tracking using a linear prediction has been proposed. However, the prediction time interval has to be fixed subject to the moving speed of the object. Thus, the quality of tracking and the power consumption will vary in diverse scenarios of tracking different kinds of objects. In order to illustrate the excellent performance of the PaM algorithm, we run another two simulations to evaluate its adaptability. The simulations are run for tracking a human and a vehicle using the algorithm proposed in [11] with a parameter $\triangle t$ which denotes the prediction time interval. In both simulations, the power consumption and the mesh ratio are recorded. Here the mesh ratio is used to express the accuracy of the prediction. The larger mesh ratio indicates the poorer quality of tracking

The results are shown in Table.4 and Fig. 8 where we can observe that when tracking a human the overall performance of the PaM algorithm is much better than the non-adaptive prediction algorithm with respect to the power consumption and the quality of tracking. When tracking a vehicle, from the power efficacy perspective, the performance of these two algorithms are similar as the prediction interval in the non-adaptive approach is set to a value between 0.5 second and 1.0 second. However, the PaM algorithm can provide much better quality of tracking.

5. Conclusions

In this paper the PaM algorithm is proposed as a practical approach for the large scale surveillance applications. The PaM algorithm uses a set of two prediction models and a mesh process to monitor the movement of the objects. It

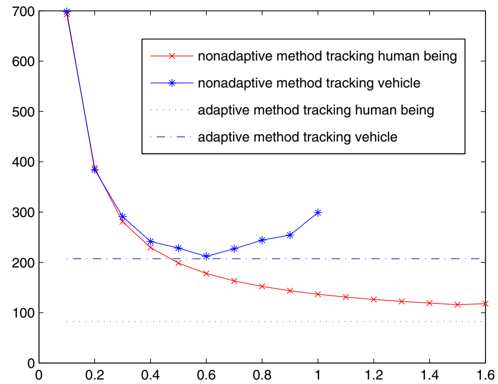

Figure 8. Energy Consumption Comparison

has proven successful in several ways. First, it provides more space for implementation since it is built based on the distributed network architecture that is randomly deployed, 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 is proved to be adaptive for tracking diverse kinds of objects with various random movement patterns.

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

References

[1] W. Blair and T. Bar-Shalom. Tracking maneuvering targets with multiple sensors: Does more data always mean better estimates? *IEEE Transactions on Aerospace and Electronic Systems*, 32(1):450–456, January 1996.

[2] T. Clouqueur, V. Phipatanasuphorn, and S. K. K. Sensor deployment strategy of target detection. *Proceedings of the AACM WSNA'02*, (1):42–48, 2002.

[3] R. Gupta and R. S. Das. Tracking moving targets in a smart sensor network. *Proceedings of IEEE VTC Fall 2003 Symposium*, pages 3035–3039, 2003.

[4] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization schemes in large scale sensor networks. *Proceedings of The Ninth International Conference on Mobile Computing and Networking (MobiCom) 2003*, pages 81–95, September 2003.

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, pages 93–104, November 2000.

[6] L. Hu and D. Evans. Localization for mobile sensor networks. *Proceedings of The Ninth International Conference on Mobile Computing and Networking (MobiCom) 2004*, pages 45–57, September 2004.

[7] M. E. Liggins, C. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopouls. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107, January 1997.

[8] J. Liu, P. Cheung, L. Guibas, and F. Zhao. A dual-space approach to tracking and sensor management in wireless sensor networks. *Proceedings of First ACM International Workshop on Wireless Sensor Networks*, 13(1):131–139, November 2002.

[9] S. Pattem, S. Poduri, and K. B. Energy-quality tradeoff for target tracking in wireless sensor networks. *ACM Trans. on Embedded Computing Systems*, pages 32–46, April 2003.

[10] J. Peng, L. Yang, F. Momen, and J. Rozenblit. Level-of-detail based modeling of object movement in tracking and surveillance system. *Proceedings of the 2005 Winter Simulation Conference*, December 2005.

[11] D. Vaidya, J. Peng, L. Yang, and J. Rozenblit. A framework for sensor management in wireless and heterogeneous sensor network. *Proceedings of 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, (1):155–162, April 2005.

[12] H. Yang and B. Sikdar. A protocol for tracking mobile targets using sensor networks. *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 71–81, May 2003.

[13] Y. Zou and K. Charkrabarty. Sensor deployment and target localization in distributed sensor networks. *ACM Trans. on Embedded Computing Systems*, 3(1):61–91, Febuary 2004.