Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2009, Article ID 524145, 14 pages
doi:10.1155/2009/524145

*Research Article*

# An Energy-Efficient Target Tracking Framework in Wireless Sensor Networks

**Zhijun Yu, Jianming Wei, and Haitao Liu**

*Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences,*
*No. 865, Changning Road, Shanghai 200050, China*

Correspondence should be addressed to Zhijun Yu, seawave.yu@yahoo.com

Received 4 September 2008; Revised 9 February 2009; Accepted 27 May 2009

Recommended by Sudharman Jayaweera

This study devises and evaluates an energy-efficient distributed collaborative signal and information processing framework for acoustic target tracking in wireless sensor networks. The distributed processing algorithm is based on mobile agent computing paradigm and sequential Bayesian estimation. At each time step, the short detection reports of cluster members will be collected by cluster head, and a sensor node with the highest signal-to-noise ratio (SNR) is chosen there as reference node for time difference of arrive (TDOA) calculation. During the mobile agent migration, the target state belief is transmitted among nodes and updated using the TDOA measurement of these fusion nodes one by one. The computing and processing burden is evenly distributed in the sensor network. To decrease the wireless communications, we propose to represent the belief by parameterized methods such as Gaussian approximation or Gaussian mixture model approximation. Furthermore, we present an attraction force function to handle the mobile agent migration planning problem, which is a combination of the node residual energy, useful information, and communication cost. Simulation examples demonstrate the estimation effectiveness and energy efficiency of the proposed distributed collaborative target tracking framework.

## 1. Introduction

Recent developments in sensor, wireless communication, and embedded computing areas now make it possible to deploy a wireless sensor network composed of a large number of inexpensive microsensor nodes to “achieve quality through quantity” in complex applications [1–3]. The nodes are typically with limited processing ability, battery power, and sensing range. In order to ensure their sustained operations, the power consumption must be kept to a minimum. Most of the signal and information processing tasks must be accomplished in network, where some nodes close to the events locally share information and resource. Only the processed data or results will be sent to the sink. This is the so-called collaborative signal and information processing (CSIP) in wireless sensor networks.

Target tracking is one of the key motivating applications of wireless sensor networks [4–8]. Passive acoustic sensor is often used in wireless sensor networks because of its universality and low cost. In this study, we address the issue of designing high energy-effective CSIP framework for acoustic target tracking applications in sensor networks, that is, to estimate the position and velocity of a moving target by collaboration. The time difference of arrival- (TDOA-) based-method is particularly attractive in this context [6, 7] since it offers higher precision than acoustic energy-based method [8] and does not require the prior knowledge of the signal generated by the potential target. One TDOA value can be calculated according to time series data from a pair of nodes by certain time delay estimation techniques such as generalized cross-correlation (GCC) methods [9, 10]. While the basic concept of the TDOA-based method can be adopted to the sensor networks problem, the energy-efficient data aggregation procedure needs to be developed and characterized. But few contributions are dedicated to this issue for TDOA-based tracking in sensor networks. A conventional data aggregation procedure is that the central processing unit (e.g., the cluster head) aggregates all the data from nodes to make a final decision [11]. It is expectable that the energy expenditure for time series data exchange will be very high.

We will call this the first CSIP (CSIP-I) scheme hereafter. In [12], an energy-aware moving target localization strategy based on a two-step communication protocol between the cluster head (CH) and cluster members was presented. The nodes that detect a target only give a binary report to the CH. Then the CH will choose only a subset of sensor nodes that must be queried for detailed target information. The querying manner is that all chosen nodes send their local data to the CH. We will call this the second CSIP (CSIP-II) scheme hereafter. This scheme can save a large amount of energy and reduce communication bandwidth, but most signal and information processing tasks are performed at the CH, which will shorten the life-span of the CH and lead to poor scalability. In [13], an information-driven approach to sensor collaboration for tracking applications in ad hoc sensor networks is overviewed, which determines participants in a "sensor collaboration" by dynamically optimizing the information utility of data for a given cost of communication and computation. In this study, the essential point is that the algorithm must be distributed and energy efficient. We propose a distributed estimation method based on generic sequential Bayesian filtering and apply it to the target state estimation at each time step. The distributed algorithm is carried out by mobile agent (MA) computing paradigm. Mobile agent methods have been widely researched for data fusion and aggregation in sensor networks' applications such as target classification or tracking [14, 15]. In this computing model, mobile agents carrying data and executable code will migrate from node to node orderly to provide progressive accuracy. The advantages such as energy efficiency and scalability make it more attractive than traditional client/server computing mode for wireless sensor networks [16].

In our framework, sensor nodes that detect a target will send short *TargetInfo* messages to the CH at each time step. Then, a reference node will be chosen for broadcasting its own time series data for TDOA calculation on other nodes. We then use the developed distributed sequential Bayesian estimation approach to achieve progressive tracking accuracy during the MA migration. The main idea is that the state posterior density, also known as the belief, is updated incrementally by integrating the measurements one by one, until a desired accuracy is satisfied or all valid nodes are queried or the maximum MA migration period expires. Note that the belief is transmitted among nodes and updated incrementally in the space domain at each time step, but it is also updated sequentially in the time domain like ordinary sequential Bayesian methods when a new time step comes. Furthermore, we use an attraction force metric to handle the MA migration planning problem, which is a combination of the node residual battery power, useful information, and communication cost. Hence, we can decrease the total energy consumption while maintaining the processing performance above a desired threshold. The processing burden is also evenly assigned among all participating nodes in our method. For the sake of convenience in simulation comparison, we will call our proposed method the third CSIP (CSIP-III) scheme hereafter. The above three CSIP schemes abstract the representative computing and processing methods for target tracking in wireless sensor networks.

The rest of this paper is organized as follows. First, we briefly describe the acoustic target tracking problem in wireless sensor networks and make some assumptions in Section 2. Section 3 will give an overview of the distributed collaborative tracking framework. In Section 4, we detail the distributed sequential Bayesian estimation algorithm, including the distributed estimation and the belief approximation methods. In Section 5, the mobile agent migration planning problem is discussed. In Section 6, numerical simulations are given to demonstrate the performance of proposed algorithm. The last section is the conclusions of this paper.

## 2. Problem Statements

In this section, we first give some assumptions of our work; then the calculating methods of TDOA measurements used for target tracking are described. Finally, the target tracking system state space models are also given. The following distributed collaborative tracking algorithm is developed based on these assumptions and models.

*2.1. Assumptions.* Following assumptions are made about the sensors and sensor networks in the development of the energy-efficient distributed collaborative target tracking framework.

(i) All sensor nodes are homogeneous. The nodes are organized as clusters which are formed after initial deployment and are maintained by certain clustering protocol such as LEACH [17]. The cluster heads are responsible of task decision and routing tracking results to the base station.

(ii) All sensor nodes are synchronized with error not more than 50 microseconds. Several well-known Reference Broadcast Synchronization (RBS) [18] and Delay measurement time synchronization (DMTS) [19] can meet this requirement.

(iii) The maximum communication range of each sensor node is greater than twice the maximum sensing range. This can guarantee all activated nodes receive the reference signal successfully during the reference signal broadcasting phase (described in Section 3.2).

(iv) At any time, there is only one target in the sensor field at most. For multiple target situations, blind source separation technologies and data association algorithms are needed to preprocess the measurements of sensors, which will be lucubrated in our future work.

(v) All nodes start with the same fixed amount of battery energy.

(vi) To compare the energy consumption during target tracking in wireless sensor networks, the energy consumed by sensor nodes when there is no target is not considered.

*2.2. TDOA Measurement Calculation.* The acoustic time series data received by a generic pair of acoustic sensors can be modeled by the following conventional equations in the discrete-time domain as

$$x_1[n] = s[n] * h_1[n] + \omega_1[n], \tag{1a}$$

$$x_2[n] = s[n] * h_2[n] + \omega_2[n], \tag{1b}$$

where $s[n]$ is the source signal, $h_i[n]$ is the impulse response between the source and the $i$th sensor. $\omega_i[n]$ is uncorrelated white Gaussian noise. Then, the TDOA value $\Delta$ between the direct paths from the source to the acoustic sensors of the generic pair can be estimated as

$$\Delta = \arg\max\left\{R_{x_1x_2}^{(g)}(d)\right\}, \tag{2}$$

where

$$R_{x_1x_2}^{(g)}(d) = \int_{-\infty}^{+\infty}\left[\Psi_g(f)G_{x_1x_2}(f)\exp(j2\pi fd)\right]df \tag{3}$$

is the GCC between $x_1$ and $x_2$. $\Psi_g(f)$ is an appropriate weighting function such as the well-known phase transform (PHAT) function, Eckart filter, and Hannan-Thomson (HT) processor [10]; $G_{x_1x_2}(f)$ is the signal cross-power spectrum. The PHAT-based GCC method is adopted in this study because of its ability to avoid causing spreading of the peak of the correlation function. Note that the proposed distributed collaborative tracking framework is applicable whatever TDOA estimation method is used.

*2.3. Target Tracking System Models.* The ultimate aim of target tracking is the online estimation of target position and velocity information from available multiple sensor observations, namely, the TDOAs. Generally, target tracking problem can be stated in terms of estimation of an unobserved discrete-time random signal in a dynamic system of the form

$$\mathbf{x}_t = f_x(\mathbf{x}_{t-1}, \mathbf{u}_t), \tag{4}$$

$$\mathbf{y}_t = f_y(\mathbf{x}_t, \mathbf{w}_t), \tag{5}$$

where $\mathbf{x}_t$ is the unknown system state vector of interest at time $t. f_x(\cdot)$ is the state transition function, and $\mathbf{u}_t$ is the process noise. $\mathbf{y}_t$ is the sensor measurement at time $t$. $f_y(\cdot)$ is the observation function, and $\mathbf{w}_t$ is the observation noise. $\mathbf{u}_t$ and $\mathbf{w}_t$ are assumed statistically independent of each other.

The unknown target state is composed of the position and velocity elements in $x$ and $y$ axes, respectively,

$$\mathbf{x}_t = \begin{pmatrix}\xi_t & \eta_t & \dot{\xi}_t & \dot{\eta}_t\end{pmatrix}^T, \tag{6}$$

where $\xi_t, \eta_t$ denote the target positions in $x$-axis and $y$-axis at time $t$, and $\dot{\xi}_t, \dot{\eta}_t$ denote the velocities in $x$-axis and $y$-axis at time $t$.

For nearly constant velocity model [20], (4) can be rewritten by

$$\mathbf{x}_t = \mathbf{F}_x\mathbf{x}_{t-1} + \mathbf{G}_x\mathbf{u}_t, \tag{7}$$

where

$$\mathbf{F}_x = \begin{bmatrix}1 & 0 & T & 0\\ 0 & 1 & 0 & T\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}, \qquad \mathbf{G}_x = \begin{bmatrix}\dfrac{T^2}{2} & 0\\ T & 0\\ 0 & \dfrac{T^2}{2}\\ 0 & T\end{bmatrix}. \tag{8}$$

Where $T$ is the sampling period of $\mathbf{y}_t$.

If the reference node for TDOA estimation is indexed by 0, the TDOA calculated at $k$th node can be modeled with respect to the target state as follows:

$$y_t^k = \frac{D_k - D_0}{v} + w_t^k = \frac{\|\mathbf{r}_s - \mathbf{r}_k\| - \|\mathbf{r}_s - \mathbf{r}_0\|}{v} + w_t^k, \tag{9}$$

where $v$ is the traveling speed of the acoustic signal. $D_k = \|\mathbf{r}_s - \mathbf{r}_k\|$ is the distance between the current target position $\mathbf{r}_s$ and the sensor node position $\mathbf{r}_k$. $w_t^k$ is the zero-mean measurement noise used to model the TDOA estimation error.

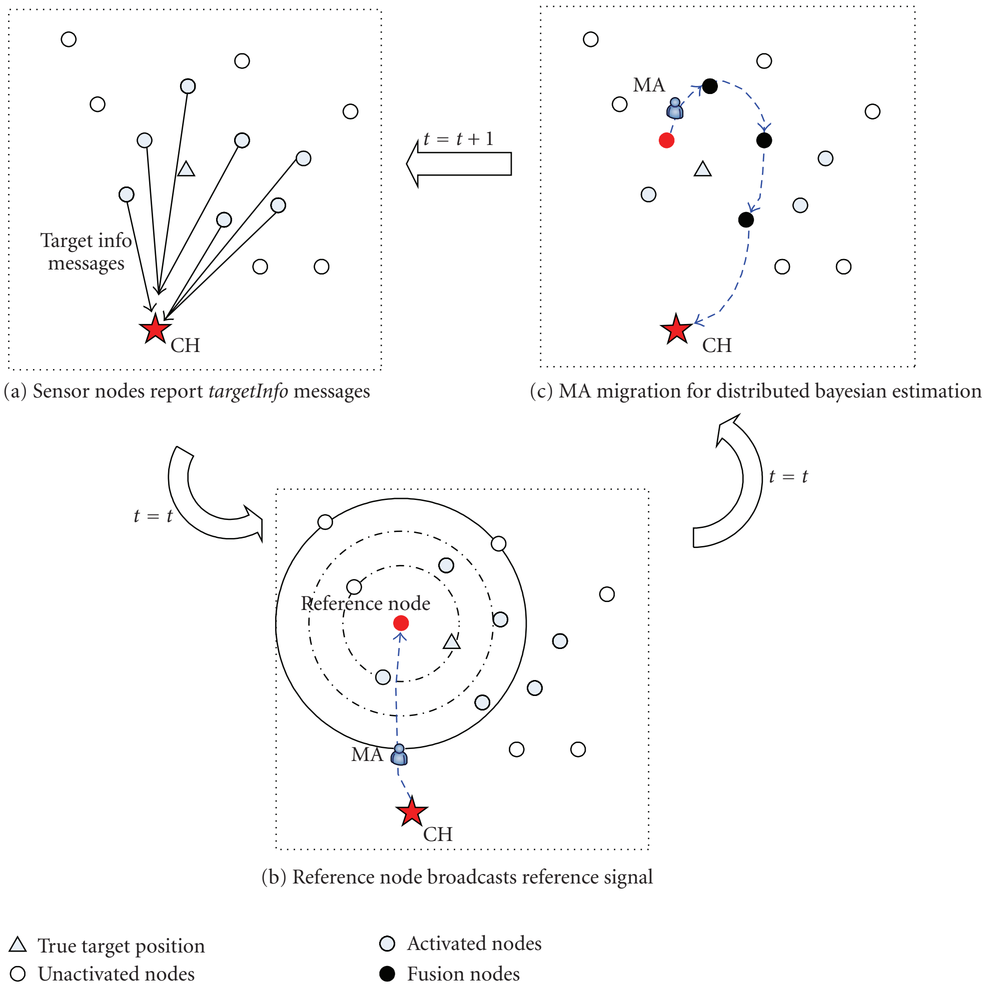
## 3. Distributed Collaborative Target Tracking Framework

In this study, we develop an energy-efficient distributed collaborative target tracking framework based on mobile agent computing paradigm. The target tracking task initialization, intracluster collaboration, intercluster collaboration, and task termination are four main aspects when implementing tracking function, which are detailed in this section.

*3.1. Tracking Task Initialization.* If a sensor node detects a target, we call it an activated node at current time step. These activated nodes will report the event to their CH. First, the CH needs to distinguish whether the tracking task has been established corresponding to this target. Because tracking results at each time step are forwarded to base station among CHs, a CH is easy to know whether the target is tracked by certain adjacent cluster. If no, the tracking task initialization will be triggered. The CH will send a *Registration* message to base station, which contains the IDs of all activated nodes. After receiving the *Registration* message, the base station will register an MA for this target. This time step is referred to as $t = 0$. Assume there are $N_0$ nodes that first detect the presence of the position of $j$th node is $(x_j, y_j)$, for $j = 1, \ldots, N_0$. The initial target state $\mathbf{x}_0$ can be estimated as

$$\hat{\mathbf{x}}_0 = \left(\frac{1}{N_0}\sum_{j=1}^{N_0}x_j \quad \frac{1}{N_0}\sum_{j=1}^{N_0}y_j \quad 0 \quad 0\right)^T. \tag{10}$$

The *registration acknowledgment* message together with initial target state $\mathbf{x}_0$ will be sent back to the CH thus the tracking task is initialized successfully. It is possible that the activated nodes may belong to several clusters, namely, there may be several CHs that send *Registration* messages to the base station. In this case, the base station will only send *registration acknowledgment* message to the cluster that has most activated nodes.



(a) Sensor nodes report *targetInfo* messages

(c) MA migration for distributed bayesian estimation

(b) Reference node broadcasts reference signal

FIGURE 1: The illustration of proposed distributed processing framework for acoustic target tracking.

*3.2. Intracluster Collaboration.* The process of intra-cluster collaboration is shown in Figure 1. There are mainly three phases.

(i) *Reporting phase*: at each time step, each activated node sends a *TargetInfo* message to the cluster head to report detected event, which contains the node ID, estimated signal-to-noise ratio (SNR), and the residual battery energy $E_i$, as listed in Table 1. To avoid collision, each activated node starts a random backoff timer before sending its *TargetInfo* message. The collection of *TargetInfo* messages is fulfilled in a time window in each cycling time step. Any *TargetInfo* message arriving after this time window will be discarded. If an activated node overhears any *TargetInfo* message from other activated nodes, it will receive and keep a copy of this message, which will be used for MA migration planning. Note that the *TargetInfo* message is very small compared with raw time series data.

(ii) *Reference signal broadcasting phase*: the CH will choose one node as the reference node according to the collected *TargetInfo* messages. The time series data of the reference node is used by other activated nodes to calculate TDOAs. First, the CH dispatches a mobile agent to the chosen reference node, which indicates the tasks of the destination and the transmission power when broadcasting the reference signal. The transmission power is large enough to guarantee that all activated nodes can receive the reference signal. Other unactivated nodes will ignore it.

(iii) *Distributed sequential Bayesian estimation phase*: in this phase, a series of sensor nodes will be queried by the MA. These nodes are called fusion nodes. They are chosen dynamically according to the *TargetInfo* messages as well as current belief estimation, which will be expatiated in Section 5. The fusion nodes will execute a distributed sequential Bayesian estimation algorithm (expatiated in Section 4) to obtain progressive tracking result by integrating the current TDOA into a Bayesian inference framework. If it is the last node needing to be queried or the new progressive result is satisfying, the MA will return to the CH. Then, the CH will pick up the final estimate and use it as a prior for the next time iteration.

Table 1: The fields contained in *TargetInfo* message.

| Field | Description |
|---|---|
| ID | The individual identification of the sensor node |
| SNR | The current estimated signal-to-noise ratio |
| $E_i$ | The current residual energy of the sensor node |

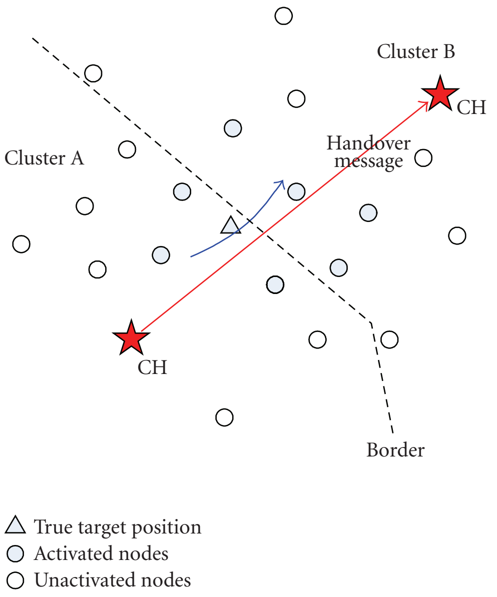


Figure 2: Illustration of target tracking task handover between clusters.

*3.3. Intercluster Collaboration.* At every time step, when a new tracking result is obtained, the CH will send out the result, which is forwarded among CHs until it arrives at the base station.

As shown in Figure 2, when the target is about to leave the current cluster (denoted by cluster A) and enter another cluster (denoted by cluster B) in the vicinity, it is intractable but important to hand over the target tracking task to cluster B at the right time. Although there is only one cluster that in charge of the target tracking task at each time step, other neighboring clusters also can give help to this cluster for better estimation. When the tracking results are forwarded to base station among CHs, each CH keeps a copy of the results. If the target is near the boundary of the active cluster, some members of neighboring clusters can also detect the presence of the target. These nodes will send the *TargetInfo* messages to their own CHs. Knowing the target tracking task is held by cluster A, the CHs will then forward the collected *TargetInfo* messages to the active CH. Upon doing so, it is expectable that better estimation will be obtained when the nodes around the current hot point are very sparse. The tracking task handover procedure will be triggered in case the number of activated nodes belonging to cluster A is less than the number of activated nodes belonging to cluster B and the estimated target motion direction is outward. The CH of cluster A will send a *Handover* message including the current estimated target state belief together with some necessary algorithm parameters to the CH of cluster B. Then cluster B will undertake the target tracking task.

*3.4. Tracking Task Termination.* When there is no sensor node that can detect the target, the current tracking task will terminate. At this time, the CH of the cluster in which the target last appears will send a short *Cancellation* message to the base station, which indicates that the previous registration of MA corresponding to the current tracking task will be cancellation. The registration-cancellation mechanism of mobile agent can guarantee that there is only one MA assigned to a target, which is very important for identification management in our future multiple target tracking study.

## 4. Distributed Sequential Bayesian Estimation

In this section, the distributed sequential Bayesian estimation algorithm is developed and applied to the tracking of a moving target using wireless sensor networks. Here, "*distributed*" means that the task of belief update for a certain time step is spatially distributed on a set of nodes; "*sequential*" means the belief is also updated in time domain when a new time step comes. In our algorithm, we need to update the state belief in time domain when a new time step comes, and transmit the belief in the network to update it in the space domain using the measurement from a new sensor node during the current time step. How to approximate the state belief properly is also critical for efficient state estimation and decreasing the communication burden.

*4.1. Algorithm Description.* To derive the sequential Bayesian estimation, we extend the basic Bayesian estimation such that it can incrementally combine measurements over space domain. Assume the local posterior estimate $p(\mathbf{x}_t \mid y_t^{1:k})$ is available after fusion node $k$ is queried. $y_t^{1:k}$ denotes the measurement sequence from fusion node 1 to fusion node $k$. At fusion node $k+1$, the posterior belief $p(\mathbf{x}_t \mid y_t^{1:k})$ carried by the MA can be used as prior information. New measurement $y_t^{1:k}$ can be used to update the prior by applying Bayes' rule, namely,

$$p\left(\mathbf{x}_t \mid y_t^{1:k+1}\right) = \frac{p\left(y_t^{k+1} \mid \mathbf{x}_t\right) p\left(x_t \mid y_t^{1:k}\right)}{p\left(y_t^{k+1} \mid y_t^{1:k}\right)}, \tag{11}$$

where the denominator is a normalizing constant which can be expressed as

$$p\left(y_t^{k+1} \mid y_t^{1:k}\right) = \int p\left(y_t^{k+1} \mid \mathbf{x}_t\right) p\left(\mathbf{x}_t \mid y_t^{1:k}\right) d\mathbf{x}_t, \tag{12}$$

so we can see that

$$p\left(\mathbf{x}_t \mid y_t^{1:k+1}\right) \propto p\left(y_t^{k+1} \mid \mathbf{x}_t\right) p\left(\mathbf{x}_t \mid y_t^{1:k}\right), \tag{13}$$

where $p(y_t^{k+1} \mid \mathbf{x}_t)$ is the likelihood function that can be achieved from the measurement model (9). Because the measurement model is nonlinear, we use Monte Carlo method to represent the required belief by a set of random samples with associated weights [21]. The details of how to obtain the belief by Monte Carlo method are given in the appendix.

In (13), the measurement $y_t^{k+1}$ is used to modify the prior density to obtain the required posterior filtering density of the current state. Then the current minimum-mean-square error (MMSE) state estimation can be calculated as

$$\begin{aligned}\hat{\mathbf{x}}_t &= \mathrm{E}\left[\mathbf{x}_t \mid y_t^{1:k+1}\right] \\ &= \int \mathbf{x}_t\, p\left(\mathbf{x}_t \mid y_t^{1:k+1}\right) d\mathbf{x}_t \\ &= \frac{\int \mathbf{x}_t\, p\left(y_t^{k+1} \mid \mathbf{x}_t\right)\, p\left(\mathbf{x}_t \mid y_t^{1:k}\right) d\mathbf{x}_t}{\int\, p\left(y_t^{k+1} \mid \mathbf{x}_t\right)\, p\left(\mathbf{x}_t \mid y_t^{1:k}\right) d\mathbf{x}_t},\end{aligned} \tag{14}$$

and the covariance matrix of the current state estimate is

$$\begin{aligned}\boldsymbol{\Sigma}_t^{k+1} &= E\left[(\mathbf{x}_t - \hat{\mathbf{x}}_t)(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \mid y_t^{1\,:k+1}\right] \\ &= \frac{\int (\mathbf{x}_t - \hat{\mathbf{x}}_t)(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T\, p\left(y_t^{k+1} \mid \mathbf{x}_t\right)\, p\left(\mathbf{x}_t \mid y_t^{1\,:k}\right) d\mathbf{x}_t}{\int\, p\left(y_t^{k+1} \mid \mathbf{x}_t\right) p\left(\mathbf{x}_t \mid y_t^{1:k}\right) d\mathbf{x}_t}.\end{aligned} \tag{15}$$

From (13) it also can be seen that the current belief is a product of the previous belief at last fusion node and the current likelihood function, which is very suitable for distributed implementation. But, there are still two aspects unsolved as follows.

(1) How to obtain the initial belief $p(\mathbf{x}_t \mid y_t^1)$ at the first fusion node from the final belief $p(\mathbf{x}_{t-1} \mid y_{t-1})$ of the last time step, where $\mathbf{y}_{t-1}$ is the vector of all TDOAs integrated at time $t-1$.

This is a belief update problem in time domain. From Bayes' rule, we also can get that

$$\begin{aligned}p(\mathbf{x}_t \mid y_t^1) &= \frac{p(y_t^1 \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{t-1})}{\int\, p(y_t^1 \mid \mathbf{x}_t)\, p(\mathbf{x}_t \mid \mathbf{y}_{t-1}) d\mathbf{x}_t} \\ &\propto p(y_t^1 \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{t-1}),\end{aligned} \tag{16}$$

where $p(\mathbf{x}_t \mid \mathbf{y}_{t-1})$ is the predictive state distribution, which can be calculated as

$$p(\mathbf{x}_t \mid \mathbf{y}_{t-1}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{t-1}). \tag{17}$$

$p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ can be calculated there according to the state transition equation (7). Known $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{t-1})$, the predictive belief $p(\mathbf{x}_t \mid \mathbf{y}_{t-1})$ can be obtained. If we obtain $p(\mathbf{x}_t \mid \mathbf{y}_{t-1})$ at the reference node and carry it to the next fusion node, the distributed Bayesian estimation process will be able to execute iteratively according to (13) and (16).

(2) How to represent the belief $p(\mathbf{x}_t \mid y_t^{1:k})$ and transmit it to the new fusion node $k+1$ in an accurate and energy-efficient manner.

In our algorithm, we need to transmit the current belief to the next node. Because of the nonlinear or even non-Gaussian characteristic of the measurement model, we cannot obtain an analytical form of the belief density. Directly transmitting a large number of samples of the belief would require significant energy consumption. Therefore, we need to represent the belief in an appropriate way.

To reduce communication burden, the posterior belief obtained at each node can be approximated by certain parameterized distribution such as Gaussian distribution, beta distribution, or Gaussian mixture model (GMM) [22]. Hence, only the distribution parameters which are much smaller than raw samples need to be transmitted among nodes. Assume that $\{\mathbf{x}_{t,k}^{(i)}\}_{i=1}^{N}$ is a set of support points to characterize the belief $p(\mathbf{x}_t \mid y_t^{1:k})$, where $N$ is the number of samples. For Gaussian approximation, the mean and covariance of the approximated posterior Gaussian can be calculated as

$$\hat{\boldsymbol{\mu}}_{t,k} = \sum_{i=1}^{N} p\left(\mathbf{x}_{t,k}^i \mid y_t^{1:k}\right) \mathbf{x}_{t,k}^i, \tag{18}$$

$$\hat{\mathbf{Q}}_{t,k} = \sum_{i=1}^{N} p\left(\mathbf{x}_{t,k}^i \mid y_t^{1:k}\right) \left(\mathbf{x}_{t,k}^i - \hat{\boldsymbol{\mu}}_{t,k}\right) \left(\mathbf{x}_{t,k}^i - \hat{\boldsymbol{\mu}}_{t,k}\right)^T. \tag{19}$$

At each hop of the MA, only the Gaussian mean $\hat{\mathbf{x}}_{t,k}$ and covariance $\hat{\mathbf{Q}}_{t,k}$ need to be transmitted. New samples can be retrieved from this distribution at the destination node.

For GMM approximation, the belief is approximated as a mixture of several Gaussian distribution

$$p\left(x_t \mid y_t^{1:k}\right) \approx \sum_{m=1}^{C} \lambda_{t,k}^m \mathcal{N}\left(\hat{\mu}_{t,k}^m, \hat{Q}_{t,k}^m\right), \tag{20}$$

where $C$ is the number of mixtures. Thus, the belief can be transmitted through the transmission of the GMM parameters $\lambda_{t,k}^m$, $\hat{\boldsymbol{\mu}}_{t,k}^m$, and $\hat{\mathbf{Q}}_{t,k}^m$, rather than the raw samples of the belief.

The number of mixtures in GMM, $C$, can be decided in advance [23] or adaptively adjusted [24]. If $C$ is fixed, the parameters of GMM are estimated using expectation-maximization method [25]. Using Lagrange multiplier, we have

$$\begin{aligned}\lambda_{t,k}^m &= \frac{1}{N}\sum_{i=1}^{N} \lambda_{t,k}\left(m \mid \mathbf{x}_{t,k}^i\right), \\ \lambda_{t,k}\left(m\mathbf{x}_{t,k}^i\right) &= \frac{\mathcal{N}\left(\mathbf{x}_{t,k}^i, \hat{\boldsymbol{\mu}}_{t,k}^m, \hat{\mathbf{Q}}_{t,k}^m\right)\lambda_{t,k}^m}{\sum_{l=1}^{C} \mathcal{N}\left(\mathbf{x}_{t,k}^i, \hat{\boldsymbol{\mu}}_{t,k}^l, \hat{\mathbf{Q}}_{t,k}^l\right)\lambda_{t,k}^l}, \\ \hat{\boldsymbol{\mu}}_{t,k}^m &= \frac{\sum_{i=1}^{N} \mathbf{x}_{t,k}^i \lambda_{t,k}\left(m \mid \mathbf{x}_{t,k}^i\right)}{\sum_{i=1}^{N} \lambda_{t,k}\left(m \mid \mathbf{x}_{t,k}^i\right)}, \\ \hat{\mathbf{Q}}_{t,k}^m &= \frac{\sum_{i=1}^{N} \lambda_{t,k}\left(m \mid \mathbf{x}_{t,k}^i\right)\left(\mathbf{x}_{t,k}^i - \hat{\boldsymbol{\mu}}_{t,k}^m\right)\left(\mathbf{x}_{t,k}^i - \hat{\boldsymbol{\mu}}_{t,k}^m\right)^T}{\sum_{i=1}^{N} \lambda_{t,k}\left(m \mid \mathbf{x}_{t,k}^i\right)}.\end{aligned} \tag{21}$$

The $C$ also can be adaptively estimated by using the modified form of the general EM algorithm in [24]. But the computation complexity may be a question. We suggest using a fixed $C$ according to the practical application requirements. Though the GMM approximation needs to transmit more parameters than Gaussian approximation, it can describe the real belief more exactly, which gives the chance of decreasing the data transmission hops to obtain satisfying precision.

*4.2. Working Scheduling.* Figure 3 shows the general work scheduling of reference node and fusion nodes. If the reference node is indexed by 0 and fusion nodes are indexed in order by $i = 1, 2, \ldots$, we can obtain the distributed sequential Bayesian estimation algorithm summarized as follows.

At time step $t$,

(i) the reference node: after receive MA from CH and broadcast its own data, it calculates predictive belief $p(\mathbf{x}_t \mid \mathbf{y}_{t-1})$ of current time step according to $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{t-1})$ and system transition model (7). Then, $p(\mathbf{x}_t \mid \mathbf{y}_{t-1})$ is approximated by Gaussian or GMM method and carried by mobile agent to transmit to the next node;

(ii) the $i$th fusion node: after receive MA, it calculates a new belief according to the received previous belief and its own TDOA measurement by (16) when $i = 1$, or, by (13) when $i > 1$. Then, it tests the quality of the current tracking result. If the result is satisfying, the MA will terminate the migration and go back to the CH; otherwise, the MA will migrate to the next node.

## 5. Mobile Agent Migration Path Planning

The above distributed sequential Bayesian estimation algorithm incrementally updates the belief of current time step by incorporating the TDOAs of a series of nodes. However, not all available activated nodes in the network provide information useful enough to improve the estimation; furthermore, some inferior measurements may corrupt the distributed inference. Therefore, we still need to plan the mobile agent migration path properly, which can provide a faster reduction in estimation uncertainty than blind or simply nearest-neighbor sensor selection, and incur a lower communication burden for meeting a given estimation performance requirement. From Sections 2 and 3 we can see that the MA migration path planning consists of two parts: the reference node selection when the MA dispatched by CH and, the next fusion node selection during the MA migration.

*5.1. SNR Estimation.* In our collaborative target tracking framework, the estimation of SNR is crucial for reference node selection and fusion node selection. The noise power spectral density (PSD) estimation has been intensively studied in speech enhancement applications [26–28]. In [26], the authors estimate the noise PSD during the speech pauses using a classic recursive relation. Martin proposed a noise estimation algorithm based on the minimum statistics [27]. In [28], the minima controlled recursive averaging (MCRA) approach is introduced for noise estimation. There are several similarities between speech signal and the acoustic signal created by ground moving target. For example, there are pauses between the target signals, and the target signal and the background noise are usually considered statistically independent. It is reasonable to apply these algorithms to acoustic target tracking applications. Here, we adopt a simple SNR calculation method which contains three steps: (1) the energy of noise is estimated as mean square of the sample points in each frame of acoustic signal and is updated sequentially, when no target in the presence. (2) The target signal energy is calculated as mean square of the sample points in each frame of acoustic signal, when a target is detected. (3) Then, the SNR is derived from the ratio between the target signal energy and the noise energy. By using this method, the background noise is tracked in succession.

*5.2. Reference Node Selection.* The reference node chosen by the CH is the destination of the first MA hop. Reference node selection is very important for TDOA calculation, which will directly influence the performance of subsequent distributed estimation. For time delay estimation, high SNR of the reference signal will improve the estimation accuracy. On the other hand, the broadcasting of time series data is very energy consuming. Therefore, the CH will choose the reference node according to the SNRs and residual energy values contained in *TargetInfo* messages

$$s_0 = \max_i \{\mathrm{SNR}_i \mid E_i > E_{\mathrm{th1}}\}, \tag{22}$$

where $E_{\mathrm{th1}}$ is an energy threshold measuring whether a sensor node is powerful enough to play the role of reference node.

*5.3. Fusion Node Selection.* The fusion node selection will determine the total of energy consumption, data fusion accuracy, agent migration time, and has a significant impact on the overall performance of the sensor network. It needs to take into consideration the tradeoffs between the migration cost and the information benefit from fusion, since although visiting more nodes improves the fusion accuracy, it also increases the communication and computation overheads. So, the objectives of our fusion node selection strategy will be reducing energy consumption and improving reliability of collaborative tracking in sensor networks.

Assume the current MA host is node $s_i$ and the set of sensor nodes whose *TargetInfo* messages are overheard by node $s_i$ is $\mathbf{S}_i$. We define an attraction force $F_{ij}$ of $s_j$ which exerts on the current MA host $s_i$ as follows:

$$F_{ij} = \alpha F_{\mathrm{power},j} + \beta F_{\mathrm{info},j} + \gamma F_{\mathrm{comm},j}, \quad \text{for } j \in \mathbf{S}_i, \tag{23}$$

where $F_{\mathrm{power},j}$, $F_{\mathrm{info},j}$, and $F_{\mathrm{comm},j}$ are the power attraction component force, information attraction component force and communication attraction component force exerting on $s_i$ by $s_j$, respectively. They have the same orientation that points to $s_j$ from $s_i$. $\alpha, \beta$, and $\gamma$ are three nonnegative constants which adjust the ratios of above three component forces, and $\alpha + \beta + \gamma = 1$.

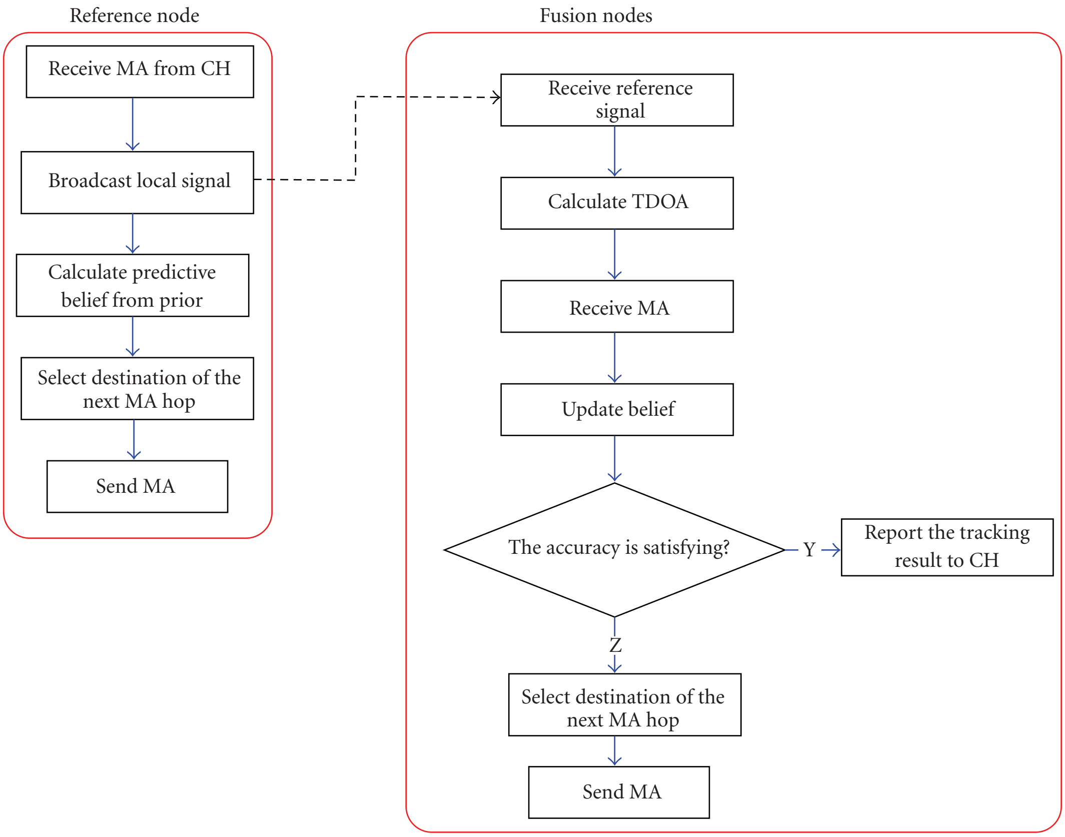

Reference node
Receive MA from CH
Broadcast local signal
Calculate predictive belief from prior
Select destination of the next MA hop
Send MA
Fusion nodes
Receive reference signal
Calculate TDOA
Receive MA
Update belief
The accuracy is satisfying?
Y
Report the tracking result to CH
Z
Select destination of the next MA hop
Send MA

Figure 3: The working flowchart of distributed sequential Bayesian estimation for target tracking.

(i) Power attraction component force $F_{\text{power},j}$. $F_{\text{power},j}$ is used to indicate the node battery energy level, which is defined as follows:

$$F_{\text{power},j} = \begin{cases} \dfrac{E_j}{E_{\max}}, & \text{if } E_j > E_{\text{th2}}, \\ -\infty, & \text{else}, \end{cases} \tag{24}$$

where $E_{\text{th2}}$ is an energy threshold measuring whether a sensor node is powerful enough to process the MA. $E_j$ is the residual energy of $s_j$. $E_{\max}$ is the maximum residual energy among allnodes in $\mathbf{S}_i$.

(ii) Information attraction component force $F_{\text{info},j}$. High SNR of signal can improve the accuracy of the TDOA calculation, so the SNR can be considered as an information measurement of a sensor node. $F_{\text{info},j}$ is defined as follows:

$$F_{\text{info},j} = \begin{cases} \dfrac{\text{SNR}_j}{\text{SNR}_{\max}}, & \text{if } \text{SNR}_j > \text{SNR}_{\text{th}}, \\ -\infty, & \text{else}, \end{cases} \tag{25}$$

where $\text{SNR}_{\text{th}}$ is the desired SNR threshold to guarantee correct TDOA estimation. If integrating incorrect TDOA into the distributed Bayesian estimation described in Section 3, the result will be corrupted. $\text{SNR}_j$ is the current SNR of $s_j$. $\text{SNR}_{\max}$ is the maximum SNR among all nodes in $\mathbf{S}_i$.

(iii) Communication attraction component force $F_{\text{comm},j}$. According to the wireless channel models, the single-hop communication energy consumption is nearly proportional to the square of distance between sender and receiver in free space field [29]. We define $F_{\text{comm},j}$ as follows:

$$F_{\text{comm},j} = -\frac{d_{ij}^2}{d_{\max}^2}, \tag{26}$$

where $d_{ij}$ is Euclidian distance between $s_i$ and $s_j \cdot d_{\max}$ is the maximum Euclidian distance among all nodes in $\mathbf{S}_i$ to node $s_i$.

Finally, the destination of the next MA hop will be chosen as

$$j^* = \max_{j\in\mathbf{S}_i}\left\{F_{ij} \mid F_{ij} \neq -\infty\right\}. \tag{27}$$

Note it is possible that there are multiple candidate nodes that have the same maximum attraction force. In this case, we will choose one node randomly among these nodes as the destination of the next MA hop.

*5.4. Return Conditions.* For our distributed collaborative tracking, the mobile agent can achieve progressive accuracy as it migrates. Once it accumulates enough information that the accuracy of the estimation meets the desire, the MA will terminate migration and return to the CH. The tracking accuracy can be measured by either the determinant of the estimation covariance $\mathbf{\Sigma}_t^{k+1}$ or the magnitude of the accuracy improvement between two successive hops. Namely, the MA can return to the CH when

$$\det\left(\mathbf{\Sigma}_t^{k+1}\right) \le \varepsilon_1, \tag{28a}$$

or

$$\left|\left|\hat{\mathbf{x}}_{t,k} - \hat{\mathbf{x}}_{t,k-1}\right|\right| \le \varepsilon_2, \tag{28b}$$

or there is no candidate nodes available, where $\varepsilon_1$, $\varepsilon_2$ are predefined performance thresholds. It is expectable that if appropriate fusion nodes are chosen, the MA will be able to have fewer hops to reach the desired tracking accuracy.

There may be some exceptions, for example, it is possible that the desired accuracy is not achieved even all activated nodes are queried. In this case, the final tracking result will be send to base station by the CH, and it can be refined by track smoothing methods later. Furthermore, there is a maximum MA migration period $T_{\text{migMax}}$ at each time step, which starts when the CH is ready to dispatch the MA and ends before the next time step is coming. Assume the time for a signal to propagate over the air to reach a receiver is negligible. If the total time for a node to receive, process, and transmit the MA is $\Delta T$, the maximum number of nodes that the MA can queried is $\lfloor T_{\text{migMax}}/\Delta T \rfloor$. The tracking accuracy may be dissatisfied when $T_{\text{migMax}}$ expires. If it happens, the MA will return to the CH immediately.

## 6. Simulations and Analyses

In this part, we set up a simulation platform to evaluate the performance of the proposed distributed collaborative target tracking framework. We will study the tracking performance of our distributed algorithm, compare the energy saving performance with CSIP-I and CSIP-II schemes, and consider the lifetime of the network which is defined as the life-span of the node whose energy is exhausted for the first.

In these simulations, $N = 64$ acoustic sensor nodes are deployed uniformly in a 35 m × 35 m square field, taking measurements corrupted by zero-mean i.i.d. Gaussian noise with variance $\sigma_w^2 = 1 \times 10^{-5}$. The data observation interval for time delay estimate is 1 second while the sampling rate is 2000 Hz. The algorithm parameters adopted in simulations are: weighting constants $\alpha = 0.2$, $\beta = 0.4$, $\gamma = 0.4$; energy thresholds $E_{\text{th1}}$ and $E_{\text{th2}}$ are set as 20% and 10% of the initial battery energy, respectively, SNR threshold $\text{SNR}_{\text{th}} = 1$ dB.

A typically tracking scenario is shown in Figure 4. The 64 nodes are managed by four clusters. Assume that a target enters the sensor field at time $t = 0$ with initial state vector $[0, 0, 0.6, 0.6]^T$and moves across the surveillance field in $T_{\text{sim}} = 30$. The target generates a 20–1000 Hz signal when moving. The process noise $\mathbf{u}_t$ is assumed Gaussian distribution with variance $\sigma_u^2 = \text{diag}([0.03, 0.03])$. The PSD of the acoustic signal is approximately even within the bandwidth. The acoustic signal is assumed propagating in isotropic air and the propagation velocity is 345 m/s. We implement the target tracking system using the CSIP-I scheme, CSIP-II scheme and the proposed CSIP-III scheme, respectively. In CSIP-I and CSIP–II, the TDOAs are calculated by CH and a generic centralized particle filter [30] is used for state estimation. The number of particles is 600 in our simulations. In CSIP-III, the Gaussian model is used to approximate the state belief. The determinant of state estimation covariance, $\det(\mathbf{\Sigma}_t^{k+1})$, is used to measure the tracking accuracy. The performance threshold $\varepsilon_1$ in (28a) is set as $2 \times 10^{-8}$.

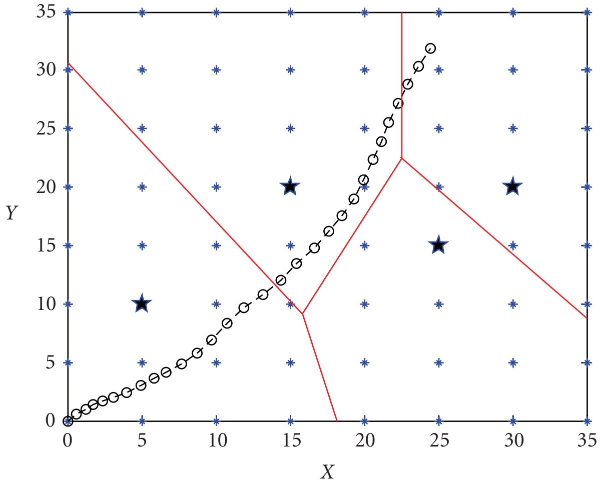


FIGURE 4: The typical tracking scenario under discussion, where the blue stars are the uniformly deployed nodes, the pentagrams are the cluster heads, and the dashed crossed black circles are the true target trace. $x$-axis unit: meter; $y$-axis unit: meter.

*6.1. Tracking Performance.* Figure 5 shows the root of mean square errors (RMSEs) of position and velocity estimations at each time step under $N_{\text{MC}} = 100$ Monte Carlo runs, according to the following equation:

$$\text{RMSE}(t) = \begin{cases} \sqrt{\dfrac{1}{N_{\text{MC}}} \displaystyle\sum_{j=1}^{N_{\text{MC}}} \left( \left(\hat{\xi}_t^j - \xi_t^{\text{true}}\right)^2 + \left(\hat{\eta}_t^j - \eta_t^{\text{true}}\right)^2 \right)}, \\ \qquad\qquad\qquad\qquad\qquad \text{for position}, \\ \sqrt{\dfrac{1}{N_{\text{MC}}} \displaystyle\sum_{j=1}^{N_{\text{MC}}} \left( \left(\hat{\dot{\xi}}_t^j - \dot{\xi}_t^{\text{true}}\right)^2 + \left(\hat{\dot{\eta}}_t^j - \dot{\eta}_t^{\text{true}}\right)^2 \right)}, \\ \qquad\qquad\qquad\qquad\qquad \text{for velocity}, \end{cases} \tag{29}$$

where $\hat{\xi}_t^j$, $\hat{\eta}_t^j$ are the estimated target positions at time step $t$ in $j$th Monte Carlo run, and $\xi_t^{\text{true}}$, $\eta_t^{\text{true}}$ are the true positions at time $t$. Similarly, $\hat{\dot{\xi}}_t^j$, $\hat{\dot{\eta}}_t^j$ are the estimated target positions at time $t$ in $j$th Monte Carlo run, and $\dot{\xi}_t^{\text{true}}$, $\dot{\eta}_t^{\text{true}}$ are the true positions at time $t$.

From Figure 5 we can see that all the three tracking information processing schemes can achieve good tracking accuracy. CSIP-I has the smallest estimation errors

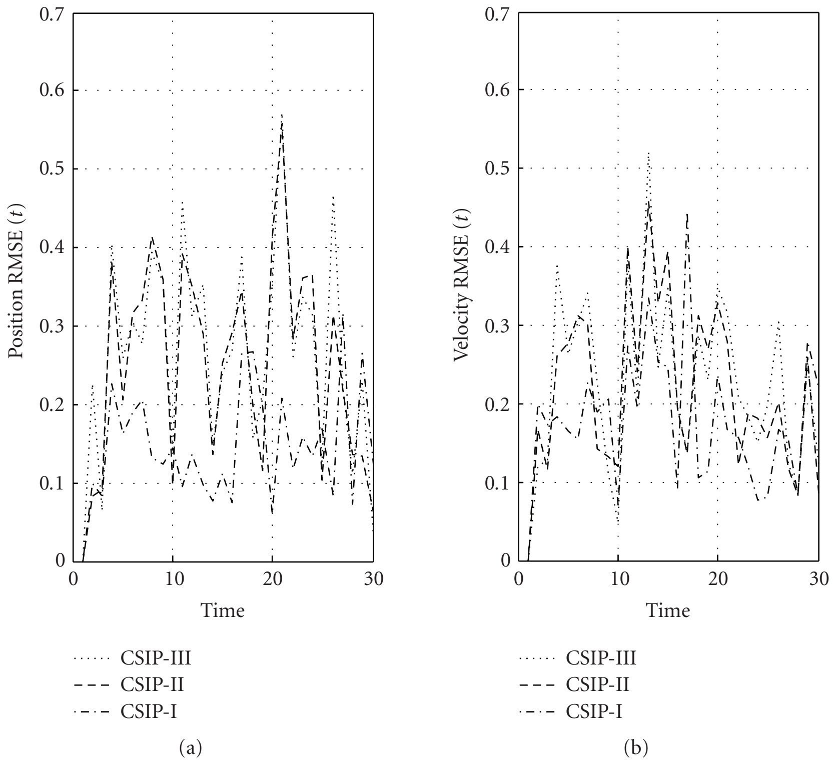


Figure 5: The position RMSE and velocity RMSE under 100 Monte Carlo simulations. $x$-axis unit: second; $y$-axis unit for left subplot: meter; $y$-axis unit for right subplot: meter per second.

in average, because data of all nodes that have detected the presence of the target are used. But, in Section 6.2, we will analyze that this high precision comes from the cost of vast energy consumption. On the other hand, the accuracy of CSIP-III is somewhat lower than CSIP-II. We think it arises from the state belief approximation during the MA migration that introduces information loss. Section 6.2 will show that the slight performance degradation is worthy in contrast to the significant energy saving benefit.

Figure 6 shows the approximated Gaussian belief of position estimation along the migrating of the MA at time snapshot $t = 24$ during one Monte Carlo run. The true target position locates at the centre of each subfigure. When the MA only visits one node, there is large estimating error and the variance of the Gaussian distribution is also very large, which means it is not a good estimate to the state. When more nodes are visited, the means of the Gaussians become very close to the true value, and the gradually constrictive colored girds indicate that the estimation uncertainty is also minished.

We also compare the performance of our method with the information-driven approach proposed in [13]. Figure 7 shows a plot of the number of fusion sensors incorporated versus the determinant of error covariance of the belief state at time step $t = 13$. In the information-driven approach, we use Mahalanobis distance as an information utility measure and Euclidean distance as an energy cost measure, thus the objective function for the optimization problem of node selection becomes

$$\begin{aligned} M(\mathbf{x}_j) = & -\alpha(\mathbf{x}_j - \hat{\mathbf{x}}_t)\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}_j - \hat{\mathbf{x}}_t) \\ & - (1-\alpha)(\mathbf{x}_j - \mathbf{x}_l)^T(\mathbf{x}_j - \mathbf{x}_l), \end{aligned} \tag{30}$$

where $\hat{\mathbf{x}}_t$, $\hat{\boldsymbol{\Sigma}}$, $\mathbf{x}_j$, $\mathbf{x}_l$ are the mean of the target position, its covariance, the position of queried sensor, and the position of querying sensor, respectively. In Figure 7, a nearest neighbor sensor selection method is also utilized as baseline for comparison.

We can see that the tracking performance is still unsatisfactory when 6 fusion nodes are queried under the nearest neighbor method. The volume of the error covariance under CSIP-III scheme is less than that under information-driven approach, except during the initial phase. To meet the predefined tracking accuracy, only 3 fusion nodes are needed to be queried under CSIP-III, while 5 fusion nodes are needed under information-driven approach. The reason that CSIP-III is superior to information-driven approach may be that CSIP-III utilizes explicit knowledge of candidate nodes, such as the SNR and residual energy. But in information-driven approach, the decision is made solely based upon the sensor characteristics such as the sensor position, and the predicted contribution of these sensors. Figure 8 is an example to indicate the difference between CSIP-III and the information-driven approach. Assume $s_2$ and $s_3$ have the

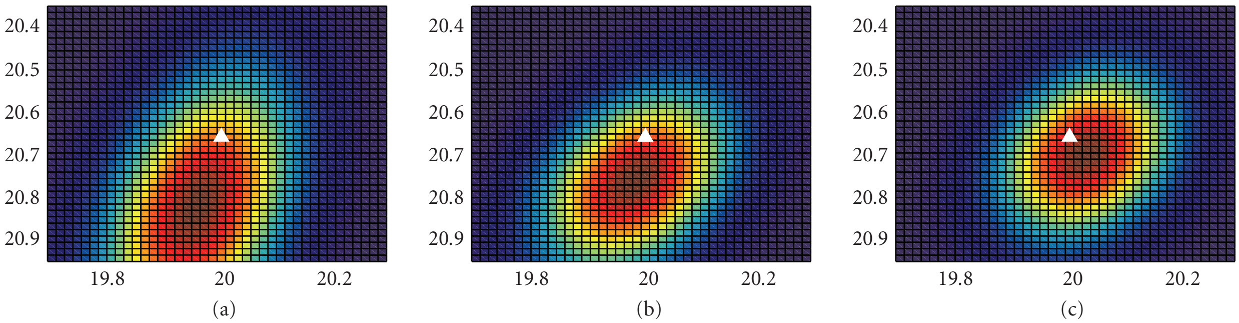


FIGURE 6: The approximated Gaussian belief after different number of TDOA values is integrated by the mobile agent. The white triangle at the centre of each subfigure is the true target position at time snapshot 15. (a) one TDOA is integrated; (b) two TDOAs are integrated; (c) three TDOAs are integrated. $x$-axis unit: meter; $y$-axis unit: meter.

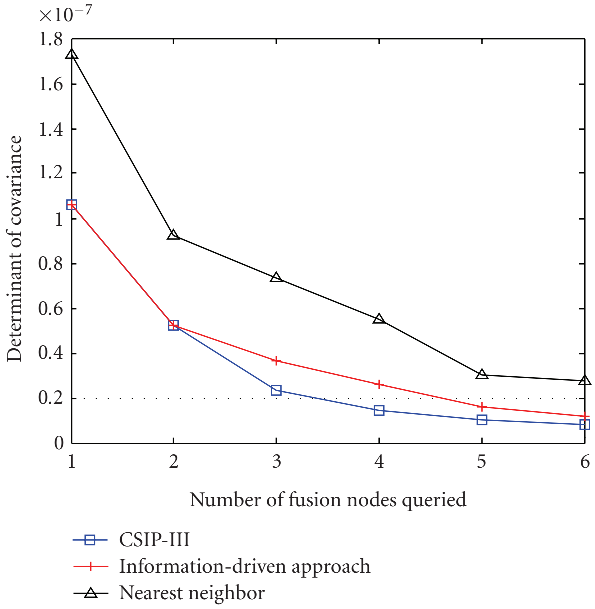


FIGURE 7: Determinant of the error covariance at time step $t = 13$ for different node selection method.

same residual energy. If the distance between $s_1$ and $s_2$ is equal to the distance between $s_1$ and $s_3$, and the current target position estimate is closer to $s_2$, information-driven approach will choose $s_2$ as the next fusion node. $s_3$ is closer to the true target position, which has higher SNR and will be chosen as the next fusion node in CSIP-III scheme.

*6.2. Energy Saving Performance.* In this section, we will evaluate the energy consumption of different collaborative processing schemes. The node energy consumption arises from sensing module, wireless communication module, and processing module. Then, the total energy depletion at time step $t$ can be expressed as

$$E(t) = E_s(t) + E_{\text{comm}}(t) + E_{\text{proc}}(t). \tag{31}$$

Among the above three parts, wireless communication module makes the main contributions. There has been a significant amount of research focusing on the low-energy radios. The first-order radio energy consumption model in [17] is adopted for rest simulations. The energy to transmit an $n$-bit message a distance $d$ is

TABLE 2: The message descriptions and their sizes used in simulation.

| Message | Sender | Receiver | Length (bit) |
|---|---|---|---|
| *TargetInfo* message for CSIP-III | nodes | CH | 80 |
| Report message for CSIP-II | nodes | CH | 32 |
| Request message for CSIP-II | CH | nodes | 96 |
| Mobile agent packet for CSIP-III | CH/nodes | nodes/CH | 312 |
| Time series data packet | nodes | CH/nodes | 32024 |

$$E_{\text{Tx}} = n * E_{\text{elec}} + n * e_{\text{free}} * d^2, \tag{32}$$

and the energy to receive an $n$-bit message is

$$E_{\text{Rx}} = n * E_{\text{elec}}, \tag{33}$$

where $E_{\text{elec}}$ is the energy spent to activate the baseband circuit to transmit or receive one bit. $e_{\text{free}}$ denotes the energy spent to run the radio frequency module to transmit one bit with acceptable bit-error rate in free space. In the simulation below, these energy parameters are set as $E_{\text{elec}} = 50\,\text{nJ/bit}$, $e_{\text{free}} = 10\,\text{pJ/bit/m}^2$. The size of each message used in the simulation is defined in Table 2.

Figure 9 shows the total communication energy consumption of the tracking process when the target moves along its trace as shown in Figure 4. Figure 10 shows the instantaneous communication energy saving of CSIP-III and CSIP-II at each time step in percentage, using CSIP-I scheme as baseline. The energy saving percentage is defined as

$$\eta = \frac{E_{\text{CSIP-I}}(t) - E(t)}{E_{\text{CSIP-I}}(t)} \times 100\%, \tag{34}$$

where $E(t) = E_{\text{CSIP-II}}(t)$or $E_{\text{CSIP-III}}(t)$. From Figures 9 and 10, we note that the CSIP-III scheme can achieve a large amount

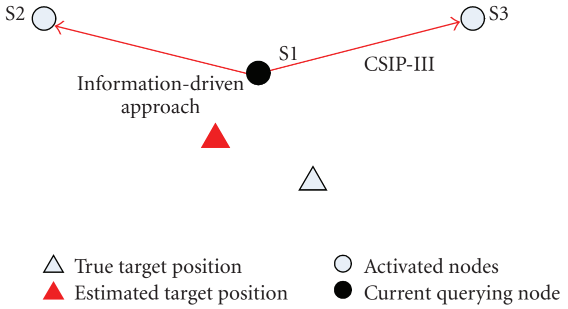


FIGURE 8: Comparison between information-driven approach and CSIP-III for node collaboration.

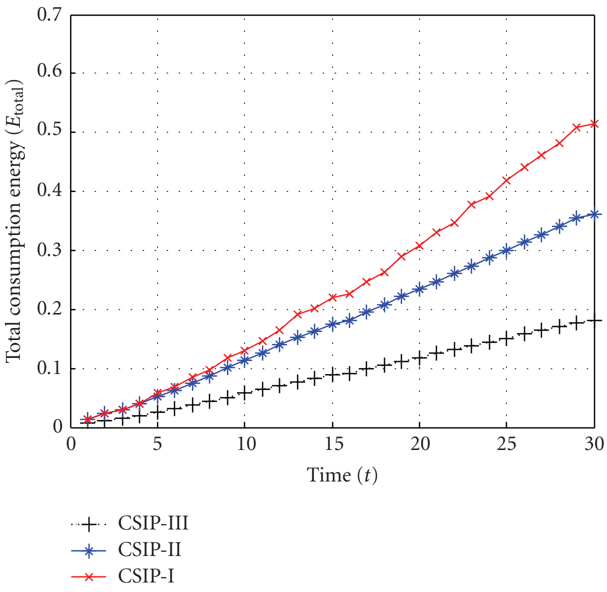


FIGURE 9: The cumulative communication energy consumptions of CSIP-1, CSIP-II, and CSIP-III after the entire trafcking process. $x$-axis unit: second; $y$-axis unit: Joule.

of energy saving comparing with CSIP-I and CSIP-II. The average instantaneous energy saving percentage of the CSIP-III scheme is above 60% relative to CSIP-I, while the CSIP-II can only obtain about 23% percentage energy saving relative to CSIP-I.

*6.3. Network Lifetime.* To prolong the sensor network lifetime, one needs to reduce total energy consumption as well as even the burden among all nodes. Unbalanced energy dissipation among nodes can lead to the situation that some nodes lose energy at a higher rate and die much faster than others, so each sensor node should have the nearly similar duration of life to prevent the blind area in coverage. The network lifetime can be measured by the time $T_{\text{dead}}$ when first node in the network is dead. A node is considered dead when its remainder energy is lower than a threshold that it can not send one data packet. In the simulation, a target moves continuously inside the sensor field in manner of the nearly constant velocity model. When arriving at any side of the field, the target will change its velocity to the mirror-reflection velocity immediately and move again, which guarantees that the target is always in the field. To avoid losing the target during dynamic clustering, the clustering architecture is kept steady during this simulation. Assume the initial energy of each sensor node is 2.5 J. Figure 11 presents the $T_{\text{dead}}$ values under different collaborative processing schemes. We can see that CSIP-I is nearly useless because the first node death occurs very early. The first dead nodes in CSIP-I and CSIP -II are expectable to be a CH because the CHs bear much more heavy tasks than their members.

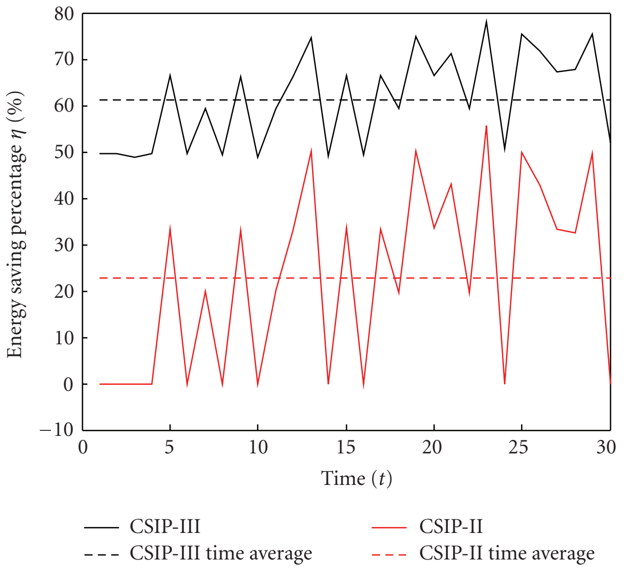


FIGURE 10: Instantaneous energy saving percentage of CSIP-III and CSIP-II relative to CSIP-I during the tracking process. $x$-axis unit: second; $y$-axis unit: percentage.

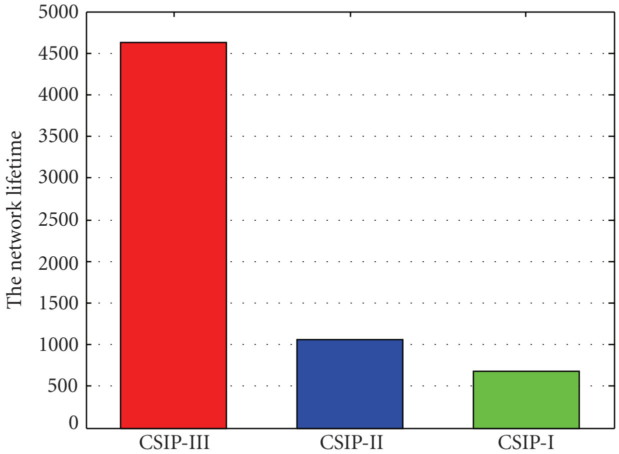


FIGURE 11: The network lifetime of different collaborative processing schemes.

## 7. Conclusions

The primary goal of this study is focusing on high energy-effective strategy of collaborative signal and information processing for acoustic target tracking applications in wireless sensor networks. Our approach is based on mobile agent computing paradigm. At each time step, the cluster head first chooses a reference node to broadcast its time series data used for TDOA calculation, and then we integrated the obtained TDOAs into the collaborative tracking framework by mobile agent migration and proposed a distributed sequential Bayesian estimation method. Actually, the proposed distributed estimation method provides a general manner for other signal and information processing applications besides target tracking. In our algorithm, we distributedly update the state belief in the space domain, which originates from the nonlinear recursive Bayesian filtering, that is, we transmit the state belief in the sensor networks by mobile agent and update the belief using the TDOA from the new fusion node. The representation of the belief is also very important because of the battery and computation capability limits for wireless sensor networks. We propose use Gaussian approximation or GMM approximation method to handle this issue. The mobile agent migration planning problem, containing the reference node selection and fusion node selection, is also considered in this study, which is implemented to maximize the available information and minimize the energy consumption cost during the mobile agent migration. Simulations show that this collaborative tracking framework can diminish the total energy consumption, prolong the network lifetime, and guarantee high tracking accuracy.

In future, we will extend our proposed methods to multiple target tracking situations, where the processing and scheduling are more complex. Associated blind source separation algorithms and light-weight data association algorithm will be investigated.

## Appendix

## A. Monte Carlo Method for Bayesian Estimation

Let $\{\mathbf{x}_{t,k+1}^{(i)}, w_{t,k+1}^{(i)}\}_{i=1}^{N}$ denote a random measure that characterizes the posterior pdf $p(\mathbf{x}_t \mid y_t^{1:k+1})$, where $\{\mathbf{x}_{t,k+1}^{(i)}\}_{i=1}^{N}$ is the set of support points with associated weights $\{w_{t,k+1}^{(i)}\}_{i=1}^{N}$. Then, $p(\mathbf{x}_t \mid y_t^{1:k+1})$ can be represented as

$$p\left(\mathbf{x}_t \mid y_t^{1:k+1}\right) \approx \sum_{i-1}^{N} w_{t,k+1}^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_{t,k+1}^{(i)}\right). \tag{A.1}$$

Therefore we have a discrete weighted approximation to the true $p(\mathbf{x}_t \mid y_t^{1:k+1})$. The weights are chosen using principle of importance sampling. Suppose $p(x) \propto \pi(x)$ is a probability density from which it is difficult to draw samples but for which $\pi(x)$ can be evaluated. In addition, let $x^{(i)} \sim q(x)$, $i = 1, 2, \ldots, N$ be samples that are easily generated from a proposal $q(\cdot)$ called an importance density. Then a weighted approximation to the density $p(x)$ is

$$p(x) \approx \sum_{i-1}^{N} w^{(i)} \delta\left(x - x^{(i)}\right), \tag{A.2}$$

where $w^{(i)}$ is the normalized weight of the $i$th particle, and

$$w^{(i)} \propto \frac{\pi\left(x^{(i)}\right)}{q(x^{(i)})}. \tag{A.3}$$

Therefore, if the importance density is $p(\mathbf{x}_t \mid y_t^{1:k})$, which is assumed known at fusion node $k+1$, then the weight $w_{t,k+1}^{(i)}$ in (A.1) can be derived to be

$$\begin{aligned} w_{t,k+1}^{(i)} &\propto \frac{p\left(\mathbf{x}_{t,k+1}^{(i)} \mid y_t^{1:k+1}\right)}{p\left(\mathbf{x}_{t,k+1}^{(i)} \mid y_t^{1;k}\right)} \\ &\propto \frac{p\left(y_t^{k+1} \mid \mathbf{x}_{t,k+1}^{(i)}\right) p\left(\mathbf{x}_{t,k+1}^{(i)} \mid y_t^{1:k}\right)}{p\left(\mathbf{x}_{t,k+1}^{(i)} \mid y_t^{1:k}\right)} \\ &= p\left(y_t^{k+1} \mid \mathbf{x}_{t,k+1}^{(i)}\right). \end{aligned} \tag{A.4}$$

## Acknowledgments

This work was supported by the Key Project of Shanghai Science and Technology Committee, China (no. 07dz15011).

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, 2002.

[3] M.-F. Wang, L.-L. Ci, P. Zhan, and Y.-J. Xu, “Design issues of wireless sensor networks in ubiquitous learning,” in *Proceedings of the 6th International Conference on Machine Learning and Cybernetics (ICMLC ’07)*, vol. 7, pp. 4149–4153, Hong Kong, August 2007.

[4] W.-P. Chen, J. C. Hou, and L. Sha, “Dynamic clustering for acoustic target tracking in wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, 2004.

[5] Y. You and H. Cha, “Scalable and low-cost acoustic source localization for wireless sensor networks,” in *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC ’06)*, pp. 517–526, Wuhan and Three Gorges, China, 2006.

[6] J. C. Chen, L. Yip, J. Elson, et al., “Coherent acoustic array processing and localization on wireless sensor networks,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1154–1162, 2003.

[7] Q. Wang, W.-P. Chen, R. Zheng, K. Lee, and L. Sha, “Acoustic target tracking using tiny wireless sensor devices,” in *Proceedings of the 2nd Workshop on Information Processing in Sensor Networks*, pp. 642–657, 2003.

[8] X. Sheng and Y.-H. Hu, “Energy based acoustic source localization,” in *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, pp. 285–300, 2003.

[9] G. C. Carter, “Time delay estimation for passive sonar signal processing,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 463–470, 1981.

[10] M. Omologo and P. Svaizer, “Use of the crosspower-spectrum phase in acoustic event location,” *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 288–292, 1997.

[11] H. Yang and B. Sikdar, “A protocol for tracking mobile targets using sensor networks,” in *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 71–81, 2003.

[12] Y. Zou and K. Chakrabarty, “Target localization based on energy considerations in distributed sensor networks,” *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 261–272, 2003.

[13] F. Zhao, J. Shin, and J. Reich, “Information-driven dynamic sensor collaboration,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[14] M. Ketel, N. S. Dogan, and A. Homaifar, “Distributed sensor networks based on mobile agents paradigm,” in *Proceedings of the 37th Annual Southeastern Symposium on System Theory*, vol. 37, pp. 411–414, 2005.

[15] Y. Xu and H. Qi, “Distributed computing paradigms for collaborative signal and information processing in sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 64, no. 8, pp. 945–959, 2004.

[16] Q. Wu, N. S. V. Rao, J. Barhen, et al., “On computing mobile agent routes for data fusion in distributed sensor networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 740–753, 2004.

[17] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[18] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” in *Proceedings of the 5th Symposium on Operating Systems Designs and Implementation*, Boston, Mass, USA, 2002.

[19] S. Ping, “Delay measurement time synchronization for wireless sensor networks,” Tech. Rep. IRB-TR-03-013, Intel Research Berkeley Lab, June 2003.

[20] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking—part I: dynamic models,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[21] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[22] J. A. O’sullivan, M. D. DeVore, V. Kedia, and M. I. Miller, “SAR ATR performance using a conditionally Gaussian model,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 1, pp. 91–108, 2001.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[24] R. Hassanpour, A. Shahbahrami, and S. Wong, “Adaptive Gaussian mixture model for skin color segmentation,” in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 31, pp. 1–6, Vienna, Austria, July 2008.

[25] T. K. Moon, “The expectation-maximization algorithm,” *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[26] P. Scalart and J. V. Filho, “Speech enhancement based on a priori signal to noise estimation,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’96)*, vol. 2, pp. 629–632, Atlanta, Ga, USA, May 1996.

[27] R. Martin, “Noise power spectral density estimation based on optimal smoothing and minimum statistics,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[28] I. Cohen and B. Berdugo, “Noise estimation by minima controlled recursive averaging for robust speech enhancement,” *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 12–15, 2002.

[29] T. S. Rappaport, *Wireless Communication: Principles and Practice*, Addison Wesley-Pearson, Reading, Mass, USA, 2nd edition, 2004.

[30] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.