WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2012; **12**:797–812
Published online 26 July 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.1014

RESEARCH ARTICLE

Prediction-based cluster management for target tracking in wireless sensor networks

Jing Teng[1]*, Hichem Snoussi[2] and Cédric Richard[3]

[1] School of Control and Computing Engineering, North China Electric Power University, 102206 Beijing, China
[2] ICD/LM2S, University of Technology of Troyes, 10000 Troyes, France
[3] Laboratory FIZEAU, UMR CNRS 6525, University of Nice Sophia, 06108 Nice cedex 02, France

ABSTRACT

The key impediments to a successful wireless sensor network (WSN) application are the energy and the longevity constraints of sensor nodes. Therefore, two signal processing oriented cluster management strategies, the proactive and the reactive cluster management, are proposed to efficiently deal with these constraints. The former strategy is designed for heterogeneous WSNs, where sensors are organized in a static clustering architecture. A non-myopic cluster activation rule is realized to reduce the number of hand-off operations between clusters, while maintaining desired estimation accuracy. The proactive strategy minimizes the hardware expenditure and the total energy consumption. On the other hand, the main concern of the reactive strategy is to maximize the network longevity of homogeneous WSNs. A Dijkstra-like algorithm is proposed to dynamically form active cluster based on the relation between the predictive target distribution and the candidate sensors, considering both the energy efficiency and the data relevance. By evenly distributing the energy expenditure over the whole network, the objective of maximizing the network longevity is achieved. The simulations evaluate and compare the two proposed strategies in terms of tracking accuracy, energy consumption and execution time. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS
wireless sensor networks; target tracking; decentralized signal processing

***Correspondence**
Jing Teng, School of Control and Computing Engineering, North China Electric Power University, 102206 Beijing, China.
E-mail: jing0404@gmail.com

1. INTRODUCTION

Owing to recent advances in wireless communications and electronic technologies, the manufacturing of low-cost, low-power, and multi-functional miniature sensors has become technically and economically feasible [1]. The reliability, flexibility, cost-effectiveness and ease of deployment make wireless sensor networks (WSNs) promise to revolutionize our life in a wide range of application domains, including military, civil, and ecological areas [2,3]. On the other hand, in spite of the diverse applications, WSNs face a number of unique technical challenges due to their inherent energy and bandwidth limitations, *ad hoc* deployment and unattended operation, etc. [4]. Therefore, WSNs have gained worldwide attention in recent years from both academia and industry [5]. In fact, the stringently constrained energy in sensor nodes is the primary limitation of WSNs. As battery-driven sensors are always left unattended, it is inconvenient or impossible to recharge or replace sensor batteries [6]. Considerable research has been focused on overcoming the deficiency of energy in the sensor nodes, increasing the sensor networks lifetime through power control schemes. Among them, clustering has proven to be an effective method to provide better quality of service (Qos) and scalability for WSNs, while conserving the limited energy and communication bandwidth. The principle of clustering is to subdivide the entire distributed WSN into small subsets of nodes depending on node density, location, or other criteria. Each of these individual subsets is called a cluster. Then, one sensor node is selected from a cluster to act as the cluster head (CH), monitoring the other sensor nodes (slaves) in the cluster. The benefit of using clusters lies in the fact that transmitting the data is reduced to small distances for most nodes, namely the slaves, while requiring only a few nodes (the CHs) to communicate in far distances.

The energy management strategy of WSN is application-oriented. Most applications of data-centric WSNs are based on the Low-Energy Adaptive Clustering Hierarchy (LEACH) scheme [6], which randomly rotates the role of CH to evenly distribute the energy load among sensors.

Misra *et al*. [7] extended LEACH by choosing the CH based on comparison of sensors' residual energy in order to achieve energy efficiency. Both the clustering schemes have been proposed for the data aggregation application, where all nodes need to participate in the cluster formation process, no matter if there is an event or not. However, as the amount of energy used in the network is proportional to the number of active sensor nodes, the use of WSNs tends to be event-driven for the sake of energy efficiency. Among the potential applications of WSNs, target tracking has attracted considerable attention in the literature, for instance, monitoring wildlife animals, vehicles on the freeway, and surveillance in the battle field etc. [8–10]. A source of difficulty in target tracking is the power consumption incurred by its inherent processes, such as the need to detect the target, to fuse the measurements and to collaborate with other sensors, which put significant requirements on the resource expenditure in WSNs. To balance the trade-off between the resource constraints and the tracking quality, energy management becomes a challenging issue that must be addressed in WSNs [11]. In fact, the relevance of the information about the target state can be achieved by a reasonable cluster activation scheduling. Generally speaking, there are two kinds of clustering strategies that are specially proposed for target tracking: The proactive and the reactive clusterings. In proactive clustering, sensor nodes are organized into clusters by statically dividing the network in advance, whereas the reactive clustering means that the clustering procedure is invoked only by events. Concerning the resource efficiency in WSNs, only one CH is responsible for tracking the target at each instant in both the clustering strategies. Examples of proactive clustering include Reference [12], which proposes a proactive cluster-based architecture for the real-world acoustic tracking applications. Their study shows that with more sensors in a cluster, the accuracy of triangulation gets higher. Because the method does not deal with the information between clusters, it cannot track a target whose motion lies in several clusters. Chen *et al*. [13] provides a hierarchical WSN composed of a static backbone of CHs and a dynamic slave sensor organization. Their design relies on dealing with collisions during the clustering phase. The main drawback of such approach is the performance degradation when the target speed increases. Tseng *et al*. [14] propose to use a mobile agent for efficient target tracking through collaborating sensors. Once a new object is detected, a mobile agent is initiated to track the roaming path of the object. The agent also invites some nearby slave sensors to cooperatively triangulate the object and inhibits other irrelevant (i.e., farther) sensors from tracking the object. However, the energy and the memory cost of this tracking strategy are not evaluated. Furthermore, in the case of irregular network topologies, it may be energy or time intensive to choose the master and the slave sensors. Yick *et al*. [15] proposes the idea of dynamically changing the size of the activated cluster and the sampling rate, based on the prediction of a Kalman Filter (KF). However, a greedy-like direct flooding approach is employed during the hand-off operation, which leads to unnecessary energy consumption. Reference [16] proposes a non-myopic clustering protocol, where a multiple detection action sequence is computed instead of simply looking at the next stage. The scheduling rule minimizes the energy usage while meeting the tracking accuracy constraint, but the computation is based on a non realistic constant-velocity target model.

Two strategies of cluster management are proposed in this contribution. Aiming at minimizing the resource overhead, the proactive cluster management (Pro-CM) statically forms clusters in advance. The WSN comprises two kinds of nodes, slave sensors and CHs, which are organized into clusters to provide an efficient cover of the whole network and to minimize the cluster overlaps. Each cluster consists of a single CH and a bunch of slave sensors. Each slave sensor belongs to exactly one cluster and is able to communicate with its CH directly via a single hop. Slave sensors simply detect the presence of the target in their sensing range and report their observations to the corresponding CHs. The CHs are in charge of coordination among the slave sensors within their clusters (intra-cluster coordination), data fusion, target tracking, and/or communication to the other CHs with external observations (inter-cluster communication). Therefore, only the CHs require additional memory, energy and much more computational resources. Compared to dynamic clustering strategies [10], the time and energy consumed in CH competition is no longer needed. Furthermore, owing to the comparatively small amount of CHs, the total hardware expenditure is reduced. In addition, when the target passes through the overlap field of several clusters, exception handling is invoked to coordinate these clusters to avoid loosing the target's track. For reactive cluster management (Re-CM), the major concern is the longevity of WSN. Since the lifetime of WSN critically depends not only on the overall energy consumption but also on the distribution of energy expenditure [17], it is rather important to evenly distribute energy consumption to avoid any drainage of the sensor batteries. In other words, all sensors in WSN should be of identical configuration in order to be capable of competing and performing the energy intensive CH role. At every sampling instant, a cluster is dynamically formed by sensors located around the phenomena of interest. To avoid biased triangulation of the target estimation, a Dijkstra-like algorithm is developed to choose the most informative sensors, ensuring optimal clustering and low computation complexity. Though the hardware expenditure is increased for the homogeneous highly configured sensors, the Re-CM strategy is much more robust to external attack than the Pro-CM, as each sensor is enabled to take over the CH task.

In the fully decentralized cluster scheme, only one cluster is triggered to perform target tracking at every sampling instant. Therefore, the belief estimate must be communicated between successive clusters when the activated cluster changes, namely the hand-off operation. This operation would defeat the purpose of decentralized signal processing, since it requires transmitting a huge amount of information. In particular, for the classic particle filtering (PF) algorithm, the raw estimates consist of large amount of particles and

their corresponding importance weights. Recently, a variational approach has been proposed in References [18,19], which approximates the *a posterior* distribution by one single Gaussian statistic to conform to the communication constraints of sensor networks. By adopting the variational filtering (VF), the inter-cluster information exchange is minimized, significantly reducing the communication bandwidth and the energy consumption. Based on the prediction of target tendency by VF, a non-myopic cluster activation rule is designed for the proactive cluster management strategy. The number of hand-off operations between successive CHs is thus reduced, minimizing corresponding energy consumption. In the reactive cluster management, clusters are also dynamically formed based on the predictive distribution. Therefore, the proposed cluster-based VF algorithm ensures tracking accuracy with minimum resource allocation.

The rest of the paper is organized as follows. The variational filtering method and its prediction phase are introduced in Section 2. Section 3 and Section 4 are devoted to the prediction-based strategy of Pro-CM and Re-CM protocols, respectively. The proposed schemes are evaluated by simulations in Section 5. Section 6 concludes the paper.

2. VARIATIONAL FILTERING ALGORITHM

The distribution of interest for target tracking takes the form of a marginal posterior distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, where $\boldsymbol{x}_t$ denotes the unobserved target position, $\boldsymbol{y}_{1:t}$ is the sequence of observed data (with the standard notation $\boldsymbol{y}_{1:t} \equiv \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_t\}$). In the Bayesian context, the task of target tracking can be formulated as recursively calculating the predictive distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ and the filtering distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ as follows:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1},$$

$$\text{and} \quad p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})}$$

One can notice from the filtering scheme that based on the state evolution model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, the estimation of the target state is updated by incorporating the observation model $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$. The observation model $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ depends on the sensing mode employed by the sensors, while the state evolution model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is always described by a parameter model for the simplicity of calculation.

2.1. General state evolution model

Instead of the kinematic parameter model [15,16,20] which is usually used in tracking problems, we employ a General State Evolution Model (GSEM) [18,19,21]. This model is more appropriate to the practical nonlinear and non-Gaussian situations, where no *a priori* information on the target velocity or acceleration is available. Considering a planar geometry, the target position $\boldsymbol{x}_t$ at instant t is assumed to follow a Gaussian model, where the expectation $\boldsymbol{\mu}_t$ and the precision matrix $\boldsymbol{\lambda}_t$ are both random. The randomness of the expectation and the precision is used here to further capture the uncertainty of the target state distribution. A practical choice of these distributions is a Gaussian distribution for the expectation and a Wishart distribution for the precision matrix. In other words, the hidden state $\boldsymbol{x}_t$ is extended to an augmented state $\boldsymbol{\alpha}_t = \{\boldsymbol{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t\}$, yielding a hierarchical model as follows:

$$\begin{cases} \boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t) \\ \boldsymbol{\mu}_t \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\lambda}}), \\ \boldsymbol{\lambda}_t \sim \mathcal{W}_2(\bar{\boldsymbol{S}}, \bar{n}) \end{cases} \tag{1}$$

where the fixed hyper parameters $\bar{\boldsymbol{\lambda}}, \bar{n}$, and $\bar{\boldsymbol{S}}$ are respectively the random walk precision matrix, the degrees of freedom, and the precision of the Wishart distribution. Assuming random mean and precision for the state $\boldsymbol{x}_t$ leads to a probability distribution covering a wide range of tail behaviors, which allows discrete jumps in the target trajectory. In fact, the marginal state distribution is obtained by integrating over the mean and precision matrix:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \int \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t)p(\boldsymbol{\mu}_t, \boldsymbol{\lambda}_t|\boldsymbol{x}_{t-1})d\boldsymbol{\mu}_t d\boldsymbol{\lambda}_t \tag{2}$$

where the integration with respect to the precision matrix leads to the known class of scale mixture distributions introduced by Barndorff–Nielsen [22]. Low values of the degrees of freedom $\bar{n}$ reflects the heavy tails of the marginal distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. In fact, varying the hyper-parameters of model Equation (1) yields a wide range of tail behaviors, from Gaussian tails to the heavy tails of the Student t-distributions. In order to illustrate the properties of scale mixture distributions, also referred to as Generalized Hyperbolic distributions, we take some examples of one-dimensional distributions with different hyper-parameters. Figure 1 represents the logarithm of Generalized Hyperbolic distributions for different hyper-parameters values. One can note the ability to cover a wide range of tail behavior.

2.2. Variational inference

As have been described above, the hidden state $\boldsymbol{x}_t$ is extended to an augmented state $\boldsymbol{\alpha}_t = \{\boldsymbol{x}_t, \boldsymbol{\mu}_t, \boldsymbol{\lambda}_t\}$. The nonlinear and the non Gaussian aspect of the GSEM (1) leads to intractable integrals when evaluating the marginal distribution $p(\boldsymbol{\alpha}_t|\boldsymbol{y}_{1:t})$. The variational Bayesian method is proposed for approximating intractable integrals arising

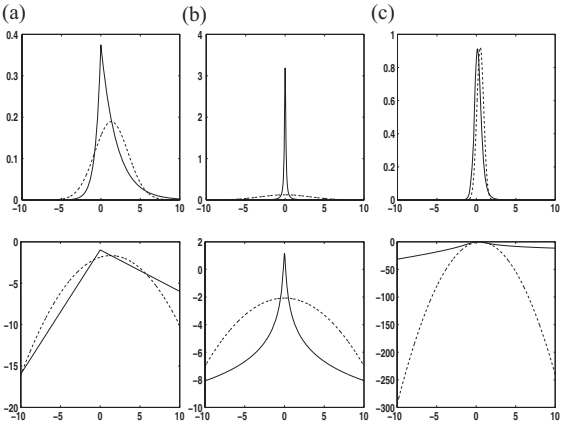

Figure 1. Examples of the generalized hyperbolic distributions: (a) Hyperbolic case; (b) Cauchy case; (c) Student case. Pdfs appear on the top row, log densities on the bottom row. The dashed lines of each column correspond to the Gaussian distributions with the same mean and variance.

in Bayesian statistics. By using a separable distribution $q(\boldsymbol{\alpha}_t)$ to lower bound the marginal likelihood, an analytical approximation to the parameter posterior probability is provided, which is useful for the prediction calculation.

$$D_{\mathrm{KL}}(q\|p) = \int q(\boldsymbol{\alpha}_t)\log\frac{q(\boldsymbol{\alpha}_t)}{p(\boldsymbol{\alpha}_t|\boldsymbol{y}_{1:t})}d\boldsymbol{\alpha}_t,$$

$$\text{where}\quad q(\boldsymbol{\alpha}_t) = \prod_i q(\boldsymbol{\alpha}_t^i) = q(\boldsymbol{x}_t)q(\boldsymbol{\mu}_t)q(\boldsymbol{\lambda}_t)$$

To minimize the KL divergence D_{KL} subject to the constraint $\int q(\boldsymbol{\alpha}_t)d\boldsymbol{\alpha}_t = \prod_i \int q(\boldsymbol{\alpha}_t^i)d\boldsymbol{\alpha}_t^i = 1$, the Lagrange multiplier method is used, yielding the following approximate distribution [18,19,21,23,24],

$$q(\boldsymbol{\alpha}_t^i) \propto \exp\langle \log p(\boldsymbol{y}_{1:t}, \boldsymbol{\alpha}_t)\rangle_{\prod_{j\neq i} q(\boldsymbol{\alpha}_t^j)}$$

where $\langle\cdot\rangle_{q(\boldsymbol{\alpha}_t^j)}$ denotes the expectation operator relative to the distribution $q(\boldsymbol{\alpha}_t^j)$. Therefore, these dependent parameters can be jointly updated. Taking into account the separable approximate distribution at time $t-1$, the filtering distribution at time t is written as

$$p(\boldsymbol{\alpha}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)\int p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})q(\boldsymbol{\alpha}_{t-1})d\boldsymbol{\alpha}_{t-1}}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})}$$
$$\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{\mu}_t,\boldsymbol{\lambda}_t)p(\boldsymbol{\lambda}_t)q_p(\boldsymbol{\mu}_t), \quad (3)$$

with $q_p(\boldsymbol{\mu}_t) = \int p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1})q(\boldsymbol{\mu}_{t-1})d\boldsymbol{\mu}_{t-1}$

The temporal dependence on the past is hence reduced to the incorporation of only one Gaussian component $q(\boldsymbol{\mu}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}^*, \boldsymbol{\lambda}_{t-1}^*)$, limiting the inter-cluster communication to simply sending the mean $\boldsymbol{\mu}_{t-1}^*$ and the precision $\boldsymbol{\lambda}_{t-1}^*$ of it, when a hand-off operation comes up. Considering the GSEM proposed in Equation (1), the evolution of $\boldsymbol{\mu}_{t-1}$ is Gaussian, namely $p(\boldsymbol{\mu}_t|\boldsymbol{\mu}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\lambda}})$. Therefore, $q_p(\boldsymbol{\mu}_t)$ is also Gaussian, $q_p(\boldsymbol{\mu}_t) \sim \mathcal{N}(\boldsymbol{\mu}_t^p, \boldsymbol{\lambda}_t^p)$, whose hyper parameters are calculated in the following subsection 2.3.The update of the filtering distribution $p(\boldsymbol{\alpha}_t|\boldsymbol{y}_{1:t})$ and its approximation are jointly performed according to the Equation (3), yielding thus a natural and adaptive compression of it, which is propagated in the sensor network without lossy compression. Detailed information on the VF algorithm is addressed in Reference [19].

2.3. Prediction calculation

The prediction calculation is of great importance for the two proposed cluster management strategies: (i) the proactive strategy activates the data processing cluster according to the predictive distribution and (ii) the reactive strategy also uses the predictive distribution to dynamically cluster candidate sensors. The predictive distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ can be efficiently updated by variational inference. In fact, taking into account the separable approximate distribution $q(\boldsymbol{\alpha}_{t-1})$, the predictive distribution is written as

$$p(\boldsymbol{\alpha}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})q(\boldsymbol{\alpha}_{t-1})d\boldsymbol{\alpha}_{t-1}$$
$$\propto p(\boldsymbol{x}_t, \boldsymbol{\lambda}_t|\boldsymbol{\mu}_t)q_p(\boldsymbol{\mu}_t) \quad (4)$$

The exponential form solution yields Gaussian distributions for the target state and its mean, while Wishart distribution for the precision matrix:

$$\begin{aligned} q_{t|t-1}(\boldsymbol{x}_t) &\propto \mathcal{N}(\langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}, \langle\boldsymbol{\lambda}_t\rangle_{q_{t|t-1}}) \\ q_{t|t-1}(\boldsymbol{\mu}_t) &\propto \mathcal{N}(\boldsymbol{\mu}_{t|t-1}^*, \boldsymbol{\lambda}_{t|t-1}^*) \\ q_{t|t-1}(\boldsymbol{\lambda}_t) &\propto \mathcal{W}_2(\boldsymbol{S}_{t|t-1}^*, n_{t|t-1}^*) \end{aligned} \quad (5)$$

where the hyper parameters are updated according to the following iterative scheme:

$$\begin{aligned} \boldsymbol{\mu}_t^p &= \boldsymbol{\mu}_{t-1}^* \\ \boldsymbol{\lambda}_t^p &= ({\boldsymbol{\lambda}_{t-1}^*}^{-1} + \bar{\boldsymbol{\lambda}}^{-1})^{-1} \\ \boldsymbol{\mu}_{t|t-1}^* &= {\boldsymbol{\lambda}_{t|t-1}^*}^{-1}(\langle\boldsymbol{\lambda}_t\rangle_{q_{t|t-1}}\langle\boldsymbol{x}_t\rangle_{q_{t|t-1}} + \boldsymbol{\lambda}_t^p\boldsymbol{\mu}_t^p) \\ \boldsymbol{\lambda}_{t|t-1}^* &= \langle\boldsymbol{\lambda}_t\rangle_{q_{t|t-1}} + \boldsymbol{\lambda}_t^p \\ n_{t|t-1}^* &= \bar{n} + 1 \\ \boldsymbol{S}_{t|t-1}^* &= \big[\langle\boldsymbol{x}_t\boldsymbol{x}_t^T\rangle_{q_{t|t-1}} - \langle\boldsymbol{x}_t\rangle_{q_{t|t-1}}\langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}^T + \bar{\boldsymbol{S}}^{-1} \\ &\quad -\langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}\langle\boldsymbol{x}_t\rangle_{q_{t|t-1}}^T + \langle\boldsymbol{\mu}_t\boldsymbol{\mu}_t^T\rangle_{q_{t|t-1}}\big]^{-1} \end{aligned} \quad (6)$$

and the expectations of the predictive target state are now evaluated by the following expressions:

$$\begin{aligned} \langle\boldsymbol{x}_t\rangle_{q_{t|t-1}} &= \langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}, \\ \langle\boldsymbol{x}_t\boldsymbol{x}_t^T\rangle_{q_{t|t-1}} &= \langle\boldsymbol{\lambda}_t\rangle_{q_{t|t-1}}^{-1} + \langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}\langle\boldsymbol{\mu}_t\rangle_{q_{t|t-1}}^T \end{aligned} \quad (7)$$

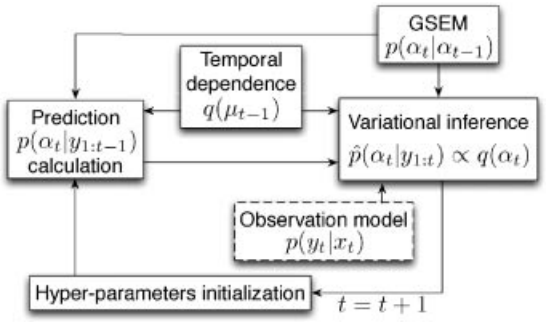

Figure 2. VF process during consecutive sampling instants.

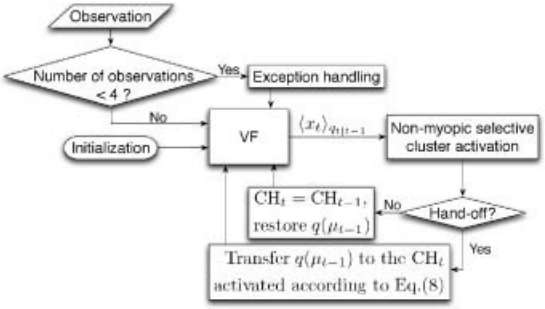

Figure 3. Flow chart of the Pro-CM strategy.

Compared to the particle filtering (PF) method, the computational cost and the memory requirement are dramatically reduced by the variational approximation in the prediction phase [18]. In fact, the expectations involved in the computation of the predictive distribution have closed forms. Thus the integral calculus in Equation (4) is reduced to several iterations to update the involved parameters till convergence (see Equation (6) for reference). On the contrary, the PF method uses a large amount of particles to bypass the integral calculus, which takes up much more storage space compared to the VF algorithm. In addition, the resampling procedure to avoid the degeneracy phenomenon of particles [25] further increases the computational cost of the PF.

An overview of the VF process during consecutive sampling instants is presented in Figure 2. The dashed rectangle denotes the procedure executed in the activated slave sensors, whereas the other sub-programs are executed in the activated CH. One can notice that the efficiency of the decentralized variational tracking algorithm depends on the relevance of the selected clusters in charge of processing the data and updating the filtering distribution. In the following, the cluster management strategies based on the predictive distribution, are described.

3. PRO-CM STRATEGY

For the single target tracking application, the region under surveillance is always quite a large area, but only a small portion of deployed sensors can detect the target at a specific time instance. Due to environmental perturbation, the observations obtained close to the phenomenon are more reliable than those obtained by distant sensors. Furthermore, local communication requires less network complexity for routing. Fewer resources, such as energy and bandwidth, are consumed by the local communication, in comparison to the multi-hop communication to the base station. Cluster-based network structure is more encouraged, since selecting the subset of informative sensors maintains desirable tracking accuracy while reducing resources consumption.

The Proactive Cluster Management (Pro-CM) strategy is proposed to minimize resource overhead in the network. It is composed of three pivotal component mechanisms (see Figure 3): (i) initialization, (ii) non-myopic selective cluster activation, and (iii) exception handling. The component mechanisms are described in the following subsections.

3.1. Initialization

The proactive aspect means that clusters are statically formed at the time of sensors deployment. Therefore, instead of identical hardware configuration for all sensors, a hierarchical WSN is formed. We assume the following properties about the network:

- all the sensors in the network are stationary and location-aware;
- slave sensors are randomly and densely deployed through the span of the network with a density ρ_s (sensors/m^2). Their sensing ranges are identically set to γ_s (m). Each slave sensor belongs to only one cluster and is able to communicate with its cluster head directly via a single hop. By employing specific observation model, they identify themselves and report their observations to corresponding cluster heads;
- cluster heads (CHs) are sparsely placed with a density ρ_{CH} (CHs/m^2) ($\rho_{\text{CH}} \ll \rho_s$) and the cluster radius are γ_{CH} (m) ($\gamma_{\text{CH}} \gg \gamma_s$). The configuration and the deployment of CHs guarantee the communication among neighboring CHs in one-hop. Furthermore, each CH is capable of rooting and forwarding the messages, therefore, all the CHs can communicate with each other in multi-hop. CHs also have sufficient memory and computational resources to update the target belief by the VF algorithm with observations collected by their slaves.

The coverage problem means that the deployment of sensors must ensure a high probability of detecting the

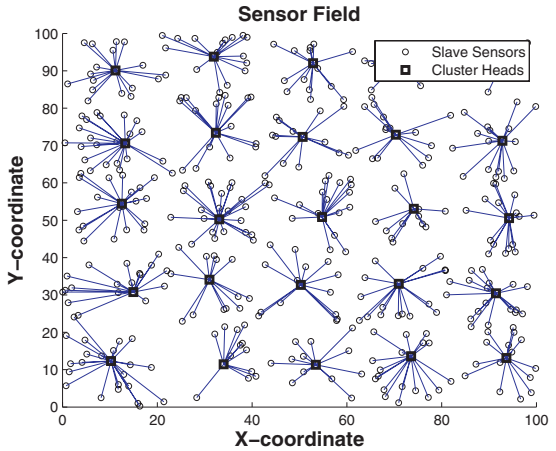

Figure 4. Initialization scenario of a sensor field, where the lines indicate the intra-cluster communication.

appearance of a target. In order to generate enough information for target tracking, at least four slave sensors are required to detect the target and to report their observations for further processing [13]. According to the network properties described above, the distribution of the number of slave sensors in any given area A is Poisson with rate $\rho_s A$. Therefore, the probability for any arbitrary point in the field to be sensed by at least four slave sensors is $p_s = \sum_{i=4}^{\infty} \frac{e^{-\rho_s \pi \gamma_s^2} (\rho_s \pi \gamma_s^2)^i}{i!}$ [8]. Similarly, the deployment of CHs should guarantee that at least one cluster is activated to track the target, the corresponding probability is $p_{\mathrm{CH}} = \sum_{i=1}^{\infty} \frac{e^{-\rho_{\mathrm{CH}} \pi \gamma_{\mathrm{CH}}^2} (\rho_{\mathrm{CH}} \pi \gamma_{\mathrm{CH}}^2)^i}{i!}$.

After the deployment of all sensors, each CH broadcasts its position within the transmission range. Based on the received signal power comparison, each slave sensor responds to the corresponding CH with its own location and ID to confirm the subordination. During the cluster formation process, CHs can register their neighboring CHs in the same way. After clustering, two kinds of lists are maintained in the CHs: the neighboring CHs list and the slave sensors list, both containing the corresponding node IDs and positions. Each CH is capable of scheduling its slave sensors and cooperating with specific neighboring CHs. Figure 4 describes such an initialization scenario of a sensor field. Initially, there is no target in the monitoring area. To minimize energy consumption, all slave sensors are set to 'Sleeping' mode. In contrast to the passive state of slave sensors, CHs are set to 'Sensing' mode to detect the appearances of unidentified objects. Such a configuration is set to ensure one-coverage throughout the whole WSN span. Once an unidentified object is detected, the corresponding CH activates its slave sensors immediately to track the target, while the remaining sensors are kept in the 'Sleeping' state. In particular, other non-invoked CHs are in the 'Sensing' mode to cope with exceptions and any possible intrusions.

3.2. Non-myopic selective cluster activation

The resource saving of cluster-based scheme lies in the fact that only one cluster is activated at each sampling instant to update the filtering distribution. As only the cluster sensors are activated to detect the moving target, the tracking accuracy thus essentially depends on their observations. Therefore, the cluster activation phase has a great importance not only in minimizing resource consumption but also in tracking accuracy. Various sensor activation strategies have been proposed in References [26,27]: (i) naive activation, where all the sensors are active; (ii) randomized activation, where a random subset of the sensors are active; (iii) selective activation, where a subset of the sensors are activated according to some performance metric; (iv) duty-cycled activation, where sensors are active for some duty cycle. By carefully choosing the activation parameters, selective activation strategy can provide great improvements in energy usage with quasi-optimal tracking quality [26]. Also, a commonly used solution [8] is to activate the CH that is nearest to the target. This strategy would incur unnecessary energy expenditure. Firstly, all the CHs need to measure the distances between the target and themselves at every sampling instant, then a comparison is required to choose the nearest one. The possibility of distributed signal processing is thus prevented and excessive communication is required. Secondly, when the target passes among the overlap of several clusters, frequent changes of activated clusters occur, resulting in frequent communication of the temporal dependence information and leading to unacceptable additional energy and bandwidth consumption. To terminate the unnecessary energy expenditure, cluster activation is based on the prediction $\boldsymbol{x}_{t|t-1} = \langle \boldsymbol{x}_t \rangle_{q_{t|t-1}}$ (subsection 2.3). If the predicted target position $\boldsymbol{x}_{t|t-1}$ remains in the vicinity of CH_{t-1}, then $\mathrm{CH}_t = \mathrm{CH}_{t-1}$. Otherwise, if $\boldsymbol{x}_{t|t-1}$ is going beyond range of the current cluster, a new CH_t is activated based on the target position prediction $\boldsymbol{x}_{t|t-1}$ and its future tendency:

$$\begin{aligned} \mathrm{CH}_t &= \mathrm{argmax}_{j=1,\ldots,k} \left(\frac{\cos \theta_t^j}{d_t^j} \right), \qquad (8) \\ \text{where} \quad d_t^j &= \| \boldsymbol{x}_{t|t-1} - \boldsymbol{p}_{\mathrm{CH}}^j \|, \\ \text{and} \quad \theta_t^j &\equiv \mathrm{angle}(\overrightarrow{\langle \boldsymbol{x}_{t-1} \rangle \boldsymbol{x}_{t|t-1}}, \overrightarrow{\langle \boldsymbol{x}_{t-1} \rangle \boldsymbol{p}_{\mathrm{CH}}^j}) \end{aligned}$$

where k is the number of CHs in the neighborhood of CH_{t-1} and $\boldsymbol{p}_{\mathrm{CH}}^j$ is the position of the j^{th} neighboring CH. Furthermore, a hand-off operation is triggered to transfer the temporal dependence information $q(\boldsymbol{\mu}_{t-1})$ from the CH_{t-1} to the CH_t. As illustrated in Figure 5, traditional cluster activation rule [8] activates CH^2 to update the filtering distribution at time t, since it is the closest CH to the target prediction $\boldsymbol{x}_{t|t-1}$. But according to the tendency, the target is very likely to go out of its vicinity in a short time, causing unnecessary hand-off operations. Based on the decision rule

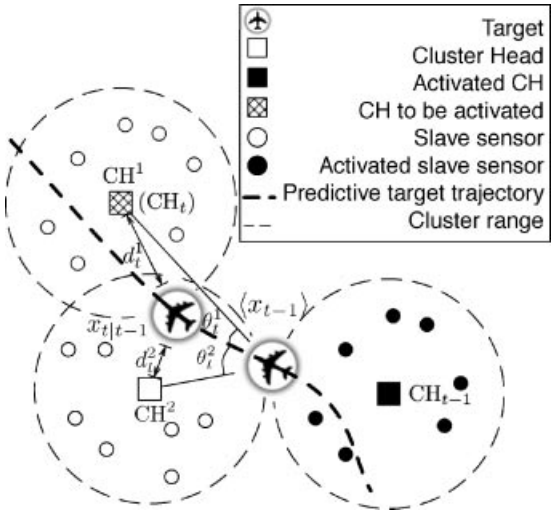

Figure 5. Prediction-based non-myopic CH_t activation.

described by Equation (8), it is CH^1 that is activated, as the target is most likely detected by it and would stay in its vicinity for a longer period. Taking into consideration the future tendency of the target, the non-myopic cluster activation rule avoids unnecessary hand-off operations compared to the traditional one. Therefore, the target tracking accuracy is ensured and the energy consumption is minimized as well. Thanks to the variational calculus, communication between CH_{t-1} and CH_t is limited to simply sending the mean and the precision of $q(\boldsymbol{\mu}_{t-1})$. Therefore, the cluster-based variational filtering (VF) algorithm outperforms the classical distributed particle filtering (PF) algorithm in resource (energy, bandwidth and memory) saving, as a large number of particles and corresponding weights are maintained and propagated in the latter case. With respect to the tracking accuracy, the VF and the PF algorithms approximate the true state distribution in different ways. When calculating the integral involved in the Bayesian filtering, the PF uses a large amount of particles whereas the VF introduces hidden variables to bypass the difficulty. These random variables introduced by the VF act as links that connect the observations to the unknown parameters via Bayes law. The error propagation problem is dramatically reduced as approximation of the filtering distribution is performed during observation incorporation.

3.3. Exception handling

Due to the restriction in the observation scope of a single cluster, the activated cluster may not provide reasonable tracking quality, when the target changes its velocity and direction of movement in an abrupt manner. Another kind of exception exists when the target moves among the overlap of several clusters. Exception handling scheme is introduced here to cope with these situations:

- For the case of an abrupt change in the target trajectory, the VF algorithm cannot export any accurate estimation, due to the irrelevant observation data of the activated slave sensors, namely the number of observations received at the activated CH is zero ($N_t^{ob} = 0$). The other CHs in the network are thus informed by a 'failure' signal, which means that an abrupt change has taken place during the target movement. As all CHs work on 'sensing' mode since the initialization phase (see the subsection 3.1), the CHs detecting the target respond to the current activated CH with their observations. The activated CH then hands-off to the best candidate by performing the cluster activation rule mentioned above in the subsection 3.2, where the predictive distribution $\boldsymbol{x}_{t|t-1}$ is replaced by the target location $\boldsymbol{x}_t$.
- For the case of lack of sufficient information, the amount of observations N_t^{ob} gathered at the activated CH is not enough for locate the target ($1 < N_t^{ob} < 4$), which suggests that the target is passing among different clusters. As the coverage problem is guaranteed in the initialization phase, at least one neighboring cluster detects the appearance of the target within its range. Charging only the activated cluster with data fusion task results in poor tracking quality. As the activated CH can only get a partial observation, biased target estimation is unavoidable. To address this issue, the observations of the neighboring CHs is transferred to the current activated CH as complementary information.

As the clusters are formed statically in advance, the number of exceptions is mainly affected by the target's moving manner (predictability of the trajectory) and the coverage of each cluster. Among these factors, only the coverage of a cluster is controllable under our assumptions. If the coverage of a cluster is too small, the second kind of exceptions will increase, which leads to frequent exception handling and energy inefficiency. On the other hand, if the coverage of a cluster is too big, too many sensors would be activated at one instant, which makes no sense of using the clustering protocol. In our strategy, the CHs are uniformly and sparsely deployed with one-coverage of the surveillance area, whereas the four-coverage is guaranteed by the randomly distributed slave sensors. This deployment strategy reduces the occurrence of the second kind of exceptions. On the other hand, the first kind of exceptions is uncontrollable as the moving manner of the target is assumed to be unpredictable in our study.

To sum up, the Pro-CM strategy yields several advantages according to its design. Firstly, the cost of hardware configuration drops sharply owing to the low-cost slave sensors. Secondly, the needed bandwidth and the consumed energy in communication are dramatically reduced. Since the signal processing task is assigned to only one activated CH, just the slave sensors belonging to the active CH are required to transmit their observations over small distances. In addition, avoiding CH competition puts an end to unnecessary resource consumption. Only when the hand-off operation occurs does the active CH need to communicate the tem-

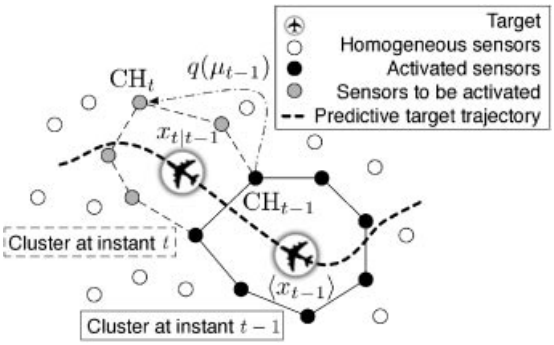

Figure 6. Reactive cluster management during successive sampling instants.

poral dependence information for a considerable distance to the subsequent CH. However, the occurrence of hand-off operations is reduced by the non-myopic selective CH activation rule and the temporal dependence information is minimized by the VF algorithm. The limitation of Pro-CM lies in its vulnerability to external attack. In fact, the activated CH is the only sensor performing tracking algorithm and none of its slave sensors would be able to take over the task.

4. RE-CM STRATEGY

Unlike the Pro-CM strategy, the reactive cluster management (Re-CM) strategy forms a cluster dynamically at every sampling instant. To ensure that each sensor in the WSN is capable of performing the task of an active CH, all of them are assumed to have an identical configuration with sufficient battery and computational power. By densely deploying the homogeneous sensors, the four-coverage problem is ensured, where $p_{hs} = \sum_{i=4}^{\infty} \frac{e^{-\rho_{hs}\pi\gamma_{hs}^2}(\rho_{hs}\pi\gamma_{hs}^2)^i}{i!} \simeq 1$. Each of the sensors keeps an address list of its neighboring sensor IDs and locations by exchanging information between each other. At the initialization step, all of them are set to the 'Sensing' mode to monitor the whole region. As soon as an intrusion is detected, sensors within the phenomenon of interest exchange information to form a cluster dynamically. CH_0 is randomly chosen among these sensors since their residual energy are identical initially. By performing the VF algorithm, the target position $\boldsymbol{x}_t$ is estimated, and a prediction of it is generated for reactive clustering. Specifically, at instant $t-1$, the target prediction $\boldsymbol{x}_{t|t-1}$ is broadcasted by the CH_{t-1}. Sensors that within the vicinity of the target prediction then respond to CH_{t-1} with their own positions and residual energy. By executing the Dijkstra-like algorithm, the CH_{t-1} activates the most potential candidate nodes among them. To balance energy consumption, the energy-intensive task of CH_t is assigned to the sensor with the most residual energy in the activated cluster. If $\mathrm{CH}_t \neq \mathrm{CH}_{t-1}$, a hand-off operation is trigged to transfer the temporal dependence $q(\boldsymbol{\mu}_{t-1})$ for further work. The Re-CM strategy during successive sampling instants is demonstrated in Figure 6.

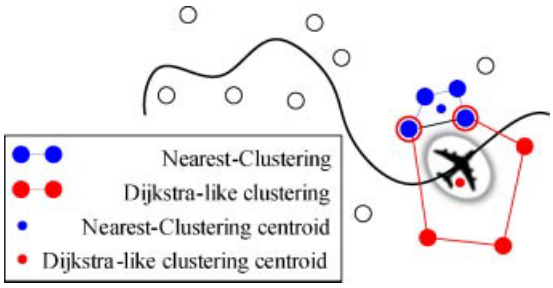

Figure 7. Clustering effectiveness comparison.

4.1. Dijkstra-like algorithm

Reactive clustering algorithm activates sensors based on two major criteria: (i) Sensors that are more informative [10,27]; (ii) Sensors that have more available residual energy [7,28]. Since target tracking triangulates the timely location of the target based on the observations of cluster sensors, the size of the cluster and the cluster sensor locations have great influence on tracking quality. The most frequently used way of forming a dynamic cluster is to choose sensors that are nearest to target. This solution raises however some questions: (i) How to decide the optimal scale of a dynamic cluster (NP-Complete problem [29])? (ii) What if the activated sensors are all gathered at the same side of the target (see Figure 7)?

We formulate the clustering problem by considering sensors as vertice of a polygon, where two sensors are connected by an edge if and only if they can communicate with each other in one-hop. The clustering strategy is thus reduced to find a polygon, whose centroid c[vertices] is nearest to the predicted target position $\boldsymbol{x}_{t|t-1}$:

$$\boldsymbol{S}_{op} = \{\text{vertices}| \arg\min[\text{distance}(c[\text{vertices}], \boldsymbol{x}_{t|t-1})]\} \quad (9)$$

where $\boldsymbol{S}_{op}$ is the optimal cluster. Note that the polygon vertices number is not fixed, which suggests no bound on the number of reactive cluster sensors. Our algorithm is inspired by the Dijkstra algorithm, which finds the path with the lowest cost (i.e., the shortest path) from a given *source* vertex to every other vertex in a graph. Instead of a single *source*, we extend the Dijkstra algorithm to calculate the path with the lowest cost between two arbitrary vertices in the graph, namely the global minimal cost, by iterating each vertex in the graph to be the *source*. Furthermore, in our algorithm, the cost to be minimized is defined as the distance between the predicted target position $\boldsymbol{x}_{t|t-1}$ and a centroid c[v] of a polygon $\boldsymbol{S}[v]$ containing the vertex v. The optimal cluster is obtained by calculating the global minimal cost as follows (see Algorithm 1).

Let $\boldsymbol{V}$ denotes the original set of candidate sensor nodes, which are all the sensors within the disk of radius γ_{hs} around the target prediction $\boldsymbol{x}_{t|t-1}$. The vertex *source* denotes the cursor vertex of each step, whose initial value is the activated CH at instant $t-1$. For each vertex v in $\boldsymbol{V}$, the following data structures are maintained: (i) the current

Algorithm 1 Dijkstra-like clustering algorithm

function DIJKSTROID($\boldsymbol{V}, \boldsymbol{x}_{t|t-1}$)
  for $i = 1; i \leq n$ **do**
    float $d[i] = \text{distance}(i,\ \boldsymbol{x}_{t|t-1})$
    $p[i] = null$
    $c[i] = null$
  end for
  $\boldsymbol{S}_{op} = null$
  $source = \text{CH}_{t-1}$
  $c[source] = source$
  $queue = \{source\}$
  while not end of $queue$ **do**
    $cursor = queue(1)$
    for $v \in \{\boldsymbol{V} - queue\}$ **do**
      $\text{dist}_{\text{new}}(v) = \text{distance}(\text{avg}(\{\boldsymbol{S}[v],\ cursor\}),\ \boldsymbol{x}_{t|t-1})$
      if $\text{dist}_{\text{new}}(v) < d[v]$ **then**
        $d[v] = \text{dist}_{\text{new}}(v)$
        $p[v] = cursor$
        $c[v] = \text{avg}(\{\boldsymbol{S}[v],\ cursor\})$
        $queue = \{queue,\ v\}$
      end if
    end for
    $queue = queue - cursor$
  end while
  $Index = \min(d[:])$
  $s = \boldsymbol{V}(Index)$
  while $p[s] \neq null$ **do**
    $\boldsymbol{S}_{op} = \{s, \boldsymbol{S}_{op}\}$
    $s = p[s]$
  end while
  return $\boldsymbol{S}_{op}$
end function

optimal cluster $\boldsymbol{S}[v]$ containing the vertex v; (ii) the centroid $c[v]$ of $\boldsymbol{S}[v]$, where $c[v] = \text{avg}(\boldsymbol{S}[v])$; (iii) the cost $d[v]$, which equals distance($c[v]$, $\boldsymbol{x}_{t|t-1}$); and (vi) $p[v]$ denotes the predecessor of the vertex v in $\boldsymbol{S}[v]$. The algorithm works by keeping the cost $d[v]$ minimum between the target prediction $\boldsymbol{x}_{t|t-1}$ and the polygon centroid $c[v]$ so far. Initially, for the vertex *source*, namely the CH_{t-1}, $d[source]$ equals distance(CH_{t-1}, $\boldsymbol{x}_{t|t-1}$), and $c[source] = \text{CH}_{t-1}$, $p[source] = null$, accordingly. For all other vertices v, $d[v] = \text{distance}(v,\ \boldsymbol{x}_{t|t-1})$, $c[v] = null$ and $p[v] = null$, these definitions represent the fact that we do not know any cluster containing those vertices. The algorithm also maintains two sets of vertices *queue* and $\{\boldsymbol{V} - queue\}$. The set *queue* contains all the vertices for which the temporal optimal costs are already calculated, and the set $\{\boldsymbol{V} - queue\}$ contains all the other vertices. The set *queue* initially contains only the vertex *source*. For each sensor v in $\{\boldsymbol{V} - queue\}$, the algorithm checks if involving *source* in $\boldsymbol{S}[v]$ can improve its known cost $d[v]$. If so, the algorithm updates $d[v]$ with the new smaller value, $p[v] = source$ and $c[v] = \text{avg}(\{\boldsymbol{S}[v],\ source\})$ is the new polygon centroid accordingly. The temporal optimal cluster $\boldsymbol{S}[v]$ is thus refreshed. Meanwhile, the vertex v is moved from $\{\boldsymbol{V} - queue\}$ to *queue*. After the check of all the vertices in $\{\boldsymbol{V} - queue\}$, the *source* vertex is popped from the set *queue*, and the *cursor* points to the first vertex in *queue* to begin a new iteration. When the algorithm converges, $\boldsymbol{S}[v]$ is the cluster that involve sensor v and minimizes $d[v]$ among all the possible clusters. Accordingly, $d[v]$ is the shortest distance from the centroid $c[v]$ of $\boldsymbol{S}[v]$ to the target prediction $\boldsymbol{x}_{t|t-1}$. By choosing the minimal value in the array d and tracing back the corresponding cluster by the predecessor array p, the optimal cluster $\boldsymbol{S}_{op}$ is formed.

Similar to the Dijkstra algorithm, the time complexity of our algorithm is $\mathcal{O}(n^2)$, where n is the number of sensor nodes in $\boldsymbol{V}$. Deployment of sensors guarantees the four-coverage problem and suggests a low probability of too many sensors detecting the target at the same sampling instant. Therefore, not only the time complexity but also the space complexity of the algorithm have dropped dramatically, leading to a high resource efficiency in WSN. The validity of the algorithm is guaranteed by its similarity to Dijkstra algorithm. A simple demonstration in Figure 7 compares its clustering result with that of the traditional nearest-clustering rule. We can see that the Dijkstra-like clustering algorithm considers not only the sensor positions but also their relationship to the target prediction. Even if the cluster members are not as close to the target as the sensors following the nearest-clustering, a much more precise and unbiased estimation is maintained by the Re-CM strategy.

4.2. Overview of the Re-CM strategy

The flow chart of the Re-CM strategy is depicted in Figure 8. Based on the prediction $\boldsymbol{x}_{t|t-1}$, the activated CH_{t-1} calculates the successive optimal cluster by the Dijkstra-like clustering algorithm. The new CH_t is elected by comparing the residual energy among the cluster members. Advantages of the Re-CM strategy can be summarized as follows. Firstly, the tracking accuracy is guaranteed by choosing the most potential sensors to dynamically form a cluster. Secondly, as the lifetime of WSN is defined as the time elapsed until the first sensor depletes its energy [7,30], it is essential to evenly distribute the energy consumption over the whole WSN. By dynamically forming clusters, CHs performing a series of energy-intensive functions are changing frequently in order to balance the energy expenditure. Thirdly, the Re-CM strategy is much more robust to external attack. However, all these advantages are at the expense of homogeneous high hardware configuration.

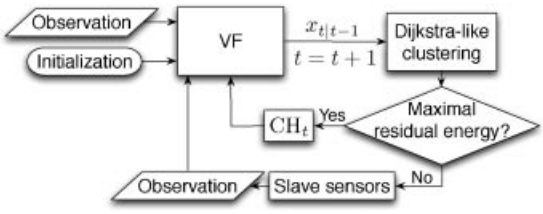

Figure 8. Flow chart of the Re-CM strategy.

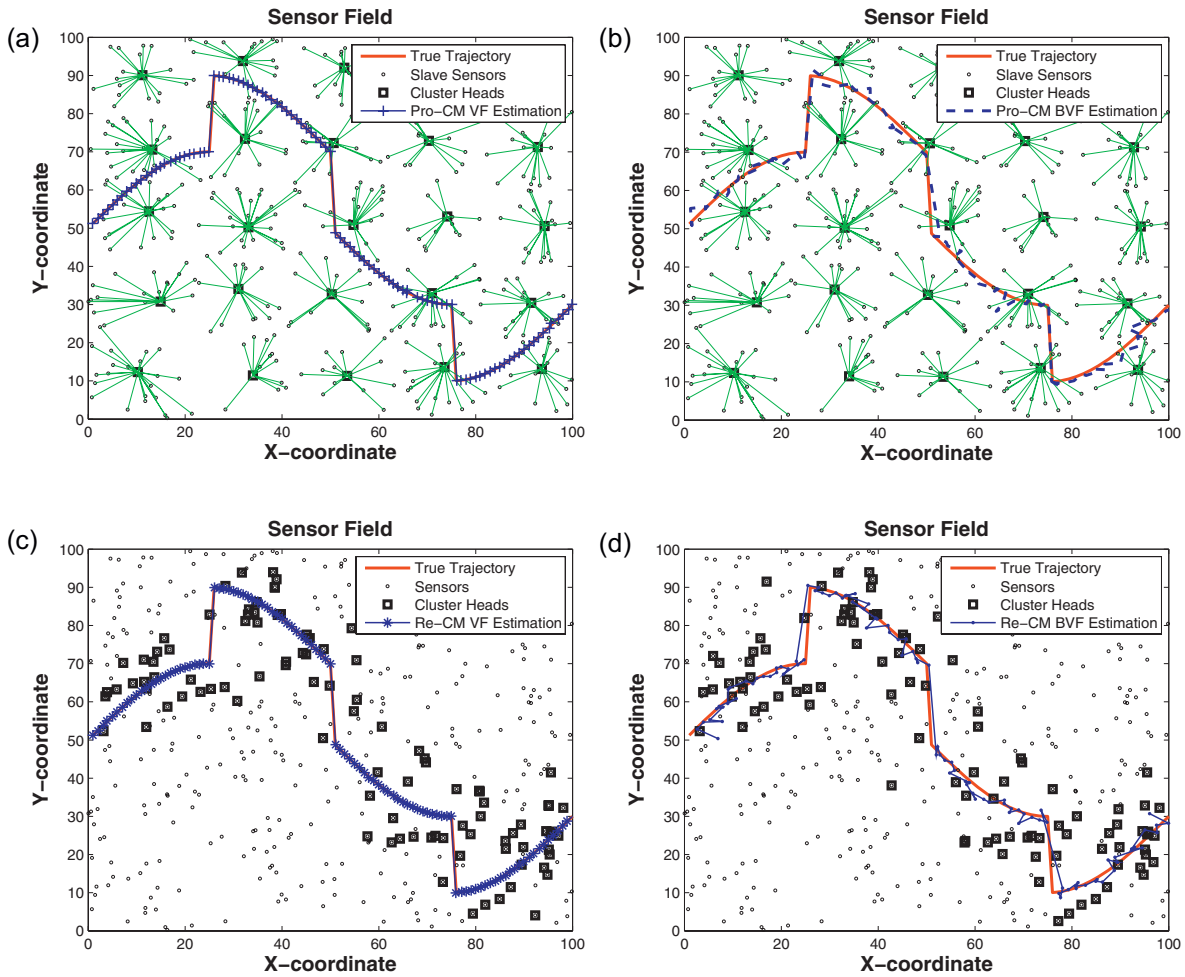

Figure 9. Target tracking performance: (a) Pro-CM with range-based measurements; (b) Pro-CM with binary proximity observation; (c) Re-CM with range-based measurements; and (d) Re-CM with binary proximity observation.

5. EVALUATION AND SIMULATION

The performances of the Pro-CM and the Re-CM strategies are evaluated by the following three criteria:

- tracking accuracy: quantified by the root mean square error (RMSE) between the estimated and the true trajectories of the target;
- energy consumption: computed with reference to the models mentioned below;
- execution time: reflected by the time complexity of the algorithm.

To calculate the energy expenditure during the whole process, we adopt two hypotheses: (i) the intra-cluster communication and the inter-cluster communication between neighboring CHs are via single hops; (ii) the energy consumed in computation can be neglected relative to energy consumed in communication. According to the energy consumption model proposed in References [16,31], the energy consumed in transmission is $E_T = \epsilon_e + \epsilon_a d^3$, where ϵ_e is the energy consumed by the circuit per *bit*, ϵ_a is the energy dissipated in *Joules* per *bit* per m^3 and d is the transmission distance ($\epsilon_a = 3.5 \times 10^{-3}$ pJ/bit/m^3, $\epsilon_e = 45$ nJ/bit). The energy consumed when receiving data is given by $E_R = \epsilon_r N$, where ϵ_r denotes the energy expended on receiving one bit of data ($\epsilon_r = 135$ nJ/bit). Similarly, the energy consumed in detection is defined by $E_S = \epsilon_s N$, where ϵ_s is the energy expended on sensing one bit of data ($\epsilon_s = 50$ nJ/bit).

Considering a target tracking duration of 100 time slots, 400 sensors were randomly deployed in a two-dimensional field (100×100 m^2). The detection radius were identically set to 9 m to ensure the four-coverage condition. For the Pro-CM strategy, all these sensors belonged to 25 sparsely positioned CHs. The cluster radius was 15 m for the requirement of one-coverage. The communication noise was assumed to be white Gaussian, with covariance set to 0.01. All the simulations shown in this paper were implemented by Matlab version 7.1, using an Intel Pentium D CPU 3.4 G Hz, 1.0 G Hz RAM PC.

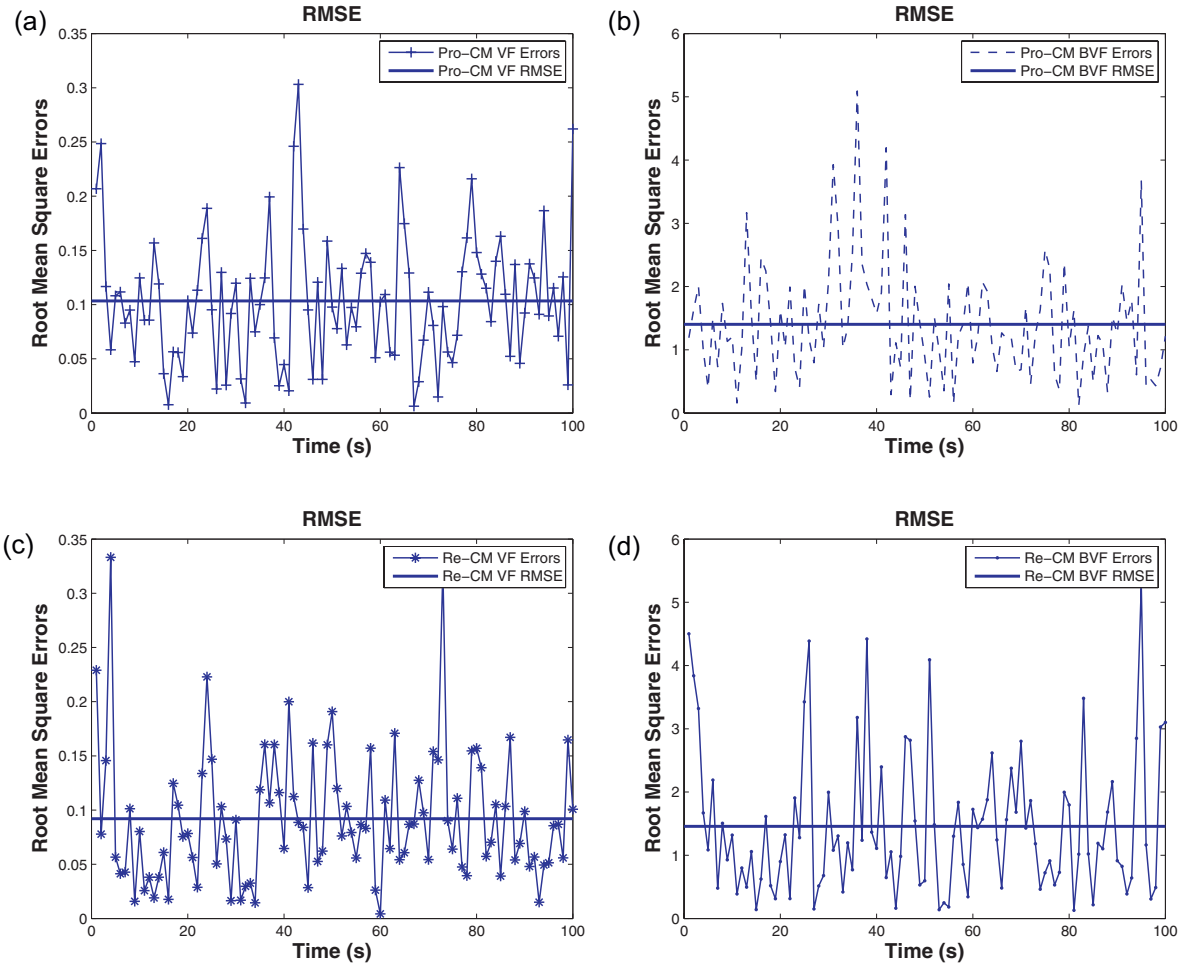

Figure 10. Root mean square errors (RMSE), for range-based measurements ((a) and (c)), and for binary proximity measurements ((b) and (d). (a) Pro-CM with range-based measurements; (b) Pro-CM with binary proximity observation; (c) Re-CM with range-based measurements; and (d) Re-CM with binary proximity observation.

Two kinds of observation models, the range-based measurement model and the binary proximity observation model, are considered to evaluate performances of the two cluster management strategies. For the range-based measurement model, slave sensors determine the distances between the target and themselves with reference to a path-loss model, then transmit their observations to the activated CH. When using the binary proximity observation model, a slave sensor simply makes a binary decision based on the strength of the perceived signal, and only one bit is transmitted to report the appearance of a target in the sensing range. A simplified hardware configuration and minimized energy consumption is thus achieved by the basic model. However, the binary proximity observation model suffers from problems of noisy links and poor estimation precision. Figure 9a and b demonstrate the tracking performances of the Pro-CM strategy by the two observation models. Accordingly, the tracking results of the Re-CM strategy with the two kinds of observation models are shown in Figure 9c and d. The corresponding root mean square errors (RMSE) are depicted in Figure 10. The simulation results confirm that, despite of the abrupt changes in the target trajectory, desired tracking quality is ensured by both the Pro-CM and the Re-CM strategies. The range-based measurement model outperforms the binary proximity observation model in tracking precision. However, taking into account the energy consumption, which is shown in Figures 11 and 12, we conclude that the binary proximity observation model is much more energy-efficient than the range observation model for both the Pro-CM and the Re-CM strategies. As shown in Figure 11, the energy consumption of the whole WSN employing the Pro-CM strategy is concentrated on the activated CHs, especially during the occurrence of hand-off operations. By assigning the energy-intensive task to the CHs of sufficient batteries, the energy consumption in the slave sensors is dramatically reduced. Based on the non-myopic selective CH activation rule of Equation (8), the number of hand-off operations, which is illustrated by

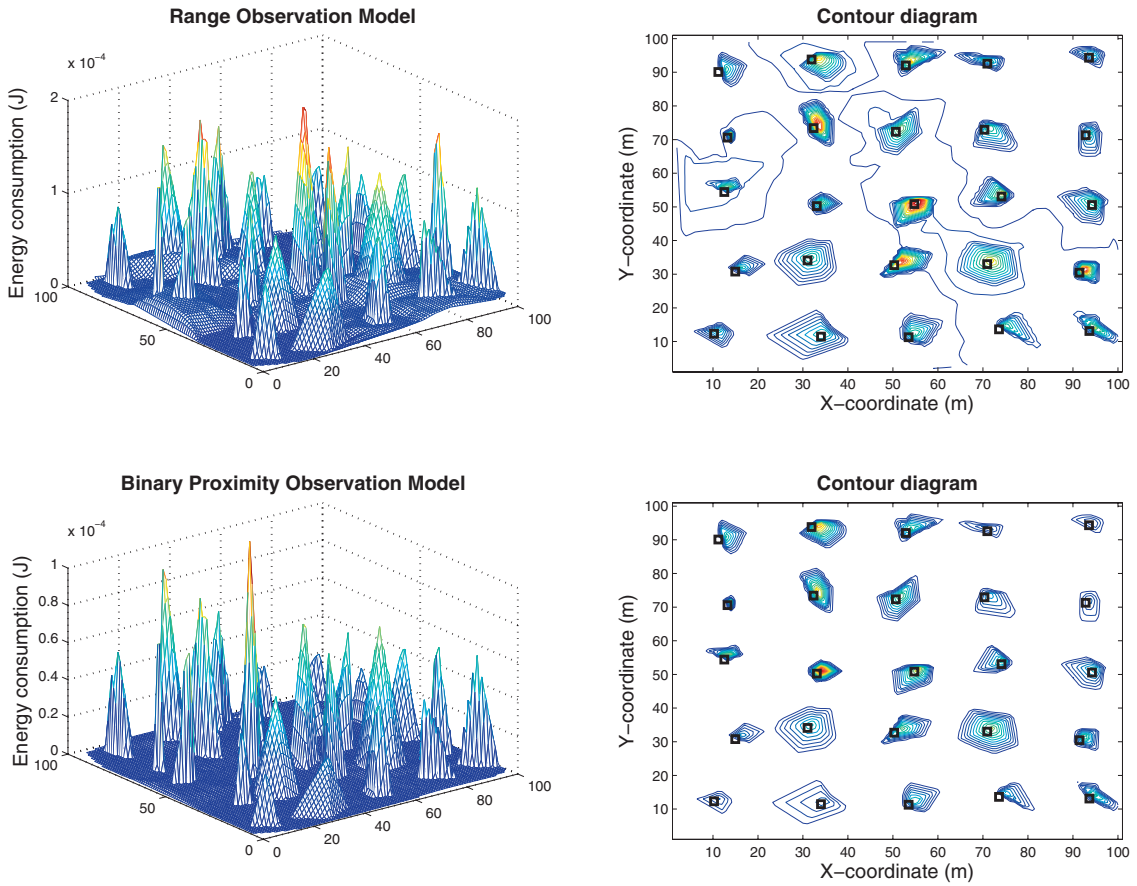

Figure 11. Energy consumption distribution of the WSN using the Pro-CM strategy.

the numbers of activated CHs in Figure 11, has dropped dramatically in comparison to that of the Re-CM strategy (see Figure 12). The overall energy consumption is thus minimized by the Pro-CM strategy. On the other hand, the Re-CM evenly distributes the energy expenditure over the whole network by selecting the candidate sensor with the most residual energy to be the activated CH. The energy consumption is thus evenly distributed along the target trajectory for both the information relevance and the energy efficiency. As illustrated in the Figures 11 and 12, the summit values of energy consumption in the Re-CM are lower than those of the Pro-CM. This observation reflects that the probability of battery drainage in the sensors employing Re-CM is comparatively reduced and prolongs thus the lifetime of the WSN.

Monte Carlo simulations were performed in order to evaluate the performances of the two strategies in terms of the three criteria mentioned above. The above expected qualitative performances of the two strategies have been confirmed as shown in Table I. With respect to the RMSE, both of them can ensure the desired tracking accuracy even with the coarse binary observation model. Typically, the RMSE obtained by the range observation model is much lower than that of the binary observation model. The reason why the performance of Pro-CM is slightly better than that of Re-CM lies in the fact that the former strategy has more redundant information. The hand-off occurrences are greatly reduced from over 97 times in the Re-CM strategy to about 16 times in the Pro-CM strategy, owing to the non-myopic selective cluster activation rule. Accordingly, the overall energy consumption of Pro-CM in the CHs is minimized compared to that of Re-CM for both the range and the binary observation models. The use of the binary proximity observation model dramatically reduces the intra-cluster and inter-cluster communication, which greatly cuts down the overall energy expenditure in both the CHs and the slave sensors. Concerning the average energy consumption, the Re-CM strategy changes the activated CHs much more frequently than the Pro-CM, in order to balance their energy consumption. The longevity of WSN is thus ensured by avoiding any battery exhaustion. In fact, the average energy consumption of the CHs in Re-CM is less than that of Pro-CM. With respect to the energy consumption in the slave sensors, as only the most informative sensors are activated to form the dynamic cluster in Re-CM, the activated cluster size is smaller in comparison to that of Pro-CM, reducing thus the overall energy consumption in slave sensors. However, those sensors employing the Re-CM strategy

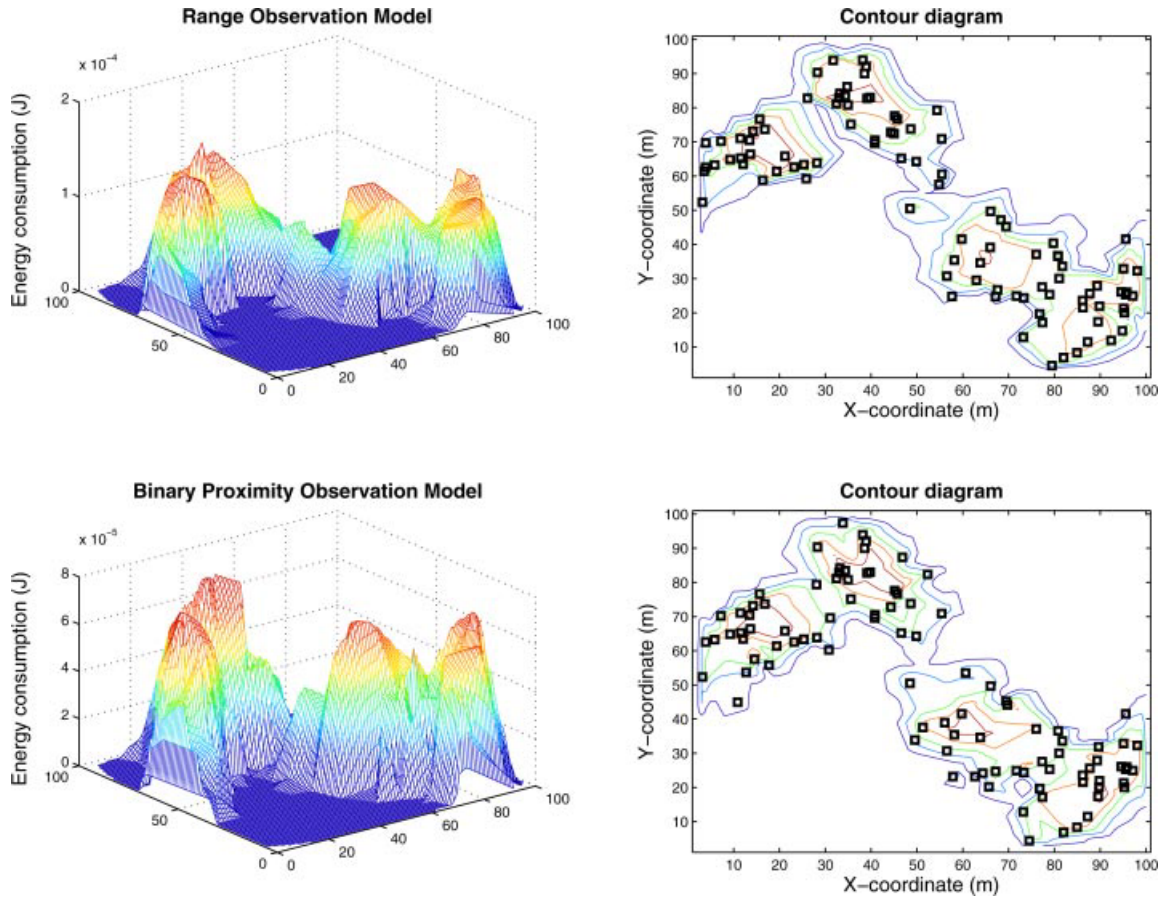

Figure 12. Energy consumption distribution of the WSN using the Re-CM strategy.

need to communicate their residual energy levels for CH competition. More energy is thus consumed by the Re-CM cluster slave sensors than the Pro-CM ones. That is why the average energy consumption by the slave sensors of Re-CM is higher than that of Pro-CM. Especially when the binary observation model is employed, the energy expenditure in observation communication is significantly reduced, which is much lower than the energy consumed in residual energy level communication. Therefore, the Pro-CM strategy demonstrates a grater improvement in energy efficiency than the Re-CM strategy, when using the binary observation model. For both strategies, accurate tracking by the range observation model comes at the expense of energy consumption and execution time, since transmitting the range

Table I. Monte Carlo evaluation of the Pro-CM and the Re-CM strategies.

	Pro-CM		Re-CM	
Evaluation	Range	Binary	Range	Binary
RMSE	0.1229	1.4389	0.2804	1.4769
Hand-off times	15.95	16.95	97.33	97.83
Overall energy consumption	CHs :3.21 mJ Slaves :3.30 mJ	CHs :1.01 mJ Slaves :0.21 mJ	CHs :9.71 mJ Slaves :1.52 mJ	CHs :5.02 mJ Slaves :1.20 mJ
Average energy consumption	129.01 μJ/CH 10.55 μJ/Slave	58.46 μJ/CH 0.67 μJ/Slave	100.67 μJ/CH 25.14 μJ/Slave	51.66 μJ/CH 18.21 μJ/Slave
Execution time	0.3872 s	0.0636 s	0.6099 s	0.0663 s

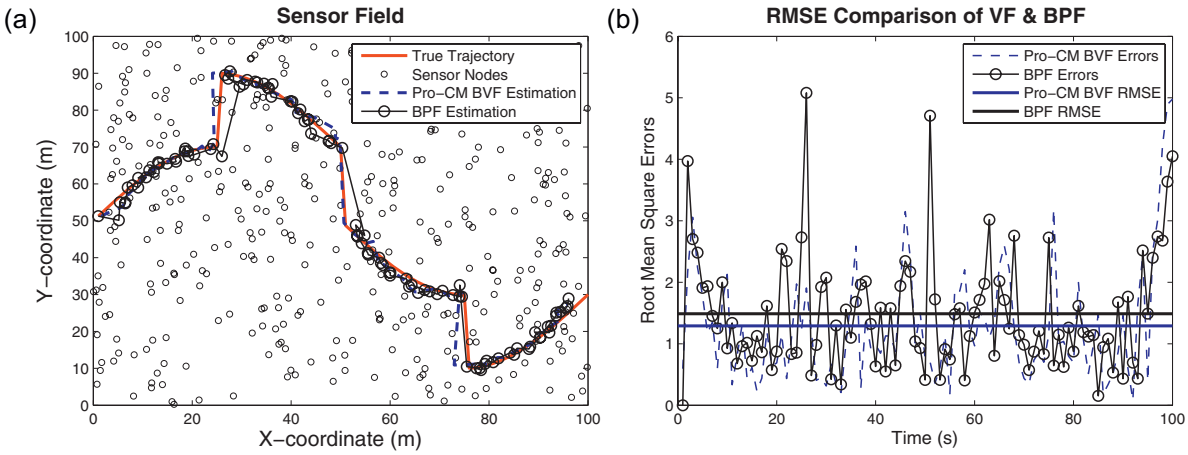

Figure 13. Tracking performance of the Pro-CM BVF and the BPF algorithms: (a) Tracking scenery and (b) RMSE.

measurements consumes much more energy, and it takes more time for the VF algorithm to converge. Nevertheless, compared to the sampling instant (1 s), both the strategies succeed in real-time target tracking even with the range observation model. In particular, for the Re-CM, more time is consumed during the dynamic clustering phase. However, based on the reliable prediction of the VF algorithm, the scale of the initial candidate set $\boldsymbol{V}$ has been reduced, leading to a significant reduction in the computation time of the Dijkstra-like clustering algorithm.

To further demonstrate the efficiency of the proposed cluster-based variational filter (VF) algorithm, we compared it with the most concurrent technique—the particle filtering algorithm using binary proximity observation model (BPF) proposed by Djuric *et al*. [20]. The tracking performances of the Pro-CM based BVF algorithm are compared with those of the BPF algorithm in Figure 13a and b, respectively, demonstrating their effectiveness in non-Gaussian context. In fact, the approximation phase of the VF and BPF algorithms are essentially based on importance sampling. The precision of tracking thus depends on the choice of the importance sampling distribution. The VF algorithm yields an optimal choice of the sampling distribution over the target position $\boldsymbol{x}_t$ by minimizing the KL divergence D_{KL}. Variational calculus leads to a simple Gaussian sampling distribution whose parameters are iteratively optimized. Due to the importance sampling phase, the space complexities during a time slot of the two algorithm are comparable, which is proportional to the number of involved particles. Tracking performances of the VF and the BPF algorithm *versus* the number of particles are shown in Figure 14a. Computation times are reported in Figure 14b. As can be expected, with the amount of particles increasing, both the algorithms demonstrate much more accurate tracking at the cost of a higher computation complexity. In particular, the computation time of the BPF grows proportionally to the increment of the number of particles. Where as the time consumed for the Pro-CM based BVF algorithm in cluster activation, involved hyper parameters update etc. is independent of the number of particles. Therefore, the

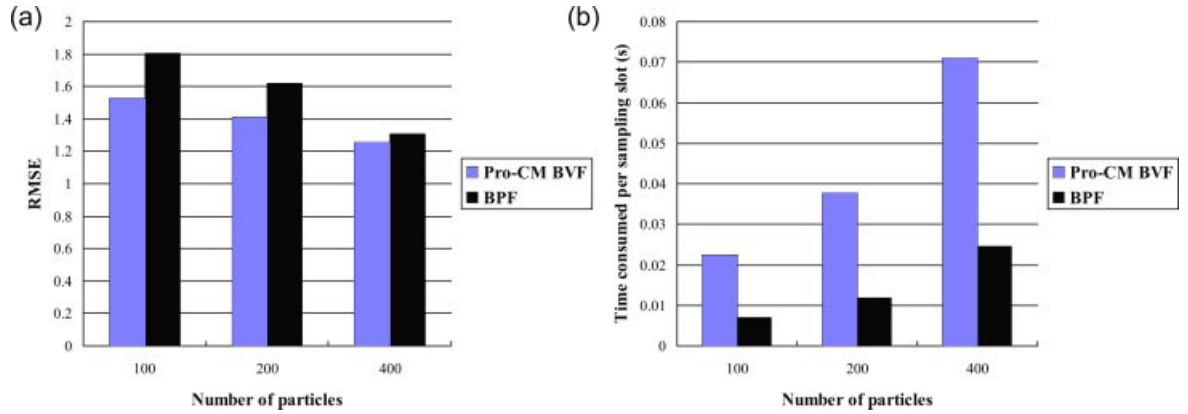

Figure 14. Comparisons of the Pro-CM BVF and the BPF algorithms versus the number of particles: (a) RMSE versus number of particles and (b) Computation time versus number of particles.

Pro-CM based BVF algorithm is more time-consuming compared to the BVF algorithm as shown in Figure 14b. Concerning the storage occupation rate during successive sampling instants, the VF algorithm maintains only one Gaussian statistic whereas the BPF algorithm stores a large number of particles.

6. CONCLUSION

Two customized target tracking schemes have been proposed in the context of wireless sensor network. Aiming at minimizing resource consumption, the Pro-CM strategy forms clusters in advance, leading to economical hardware configuration. The reaction time and the communication burden during the clustering phase are thus dispensed with. Owing to the non-myopic CH activation rule, hand-off operations dramatically decrease, further reducing energy consumption and inter-cluster communication. Taking into account the longevity and the robustness of WSN, the Re-CM strategy is introduced. It consists of forming clusters reactively to maximize observation data relevance and to evenly distribute energy consumption. Employing the Dijkstra-like clustering algorithm allows an unbiased estimation and an optimal solution to the clustering phase. The energy and communication intensive task of the CH is evenly distributed throughout the whole network by the residual energy level comparison among candidate sensors. Based on the reliable prediction of the VF algorithm, effective clustering is ensured not only in terms of accuracy but also of reduction in computation time. Furthermore, the VF algorithm guarantees the tracking quality even in nonlinear and non-Gaussian environments, and significantly reduces the inter-cluster communication to a single Gaussian distribution. The binary proximity and the range observation models have been adopted in the simulations for evaluation purpose.

REFERENCES

1. Al-Karaki JN, Kamal AE. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications* 2004; **11**: 6–28.
2. Tilak S, Abu-Ghazaleh NB, Heinzelman W. A taxonomy of wireless micro-sensor network models. *ACM Mobile Computing and Communications Review* 2002; **6**: 28–36.
3. Wang Y, Attebury G, Ramamurthy B. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys and Tutorials* 2006; **8**: 2–23.
4. Ihler AT. Inference in sensor networks: graphical models and particle methods. *Ph.D. Thesis*, Massachusetts Institute of Technology, 2005.
5. Zhou ZY, Fang Y, Zhang Y. Securingwireless sensor networks: a survey. *IEEE Communications Surveys & Tutorials* 2008; **10**: 6–28.
6. Heinzelman W, Chandrakasan A, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications* 2002; **1**(4): 660–670.
7. Misra IS, Dolui S, Das A. Enhanced energy-efficient adaptive clustering protocol for distributed sensor networks. In *The 13th IEEE International Conference on Networks*, 2005.
8. Yang H, Sikdar B. A protocol for tracking mobile targets using sensor networks. In *The 1st IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.
9. He G, Hou JC. Tracking targets with quality in wireless sensor networks. In *ICNP 2005*, 2005.
10. Zang C, Haibin Y, Wei L, Bai J, Li G. Target tracking based on the dynamic cluster method in the acoustic sensor network. In *Proceedings of the The 5th International Conference on Computer and Information Technology*, IEEE Computer Society, Washington, DC, 2005; 362–367.
11. Wang X, Ma J-J, Wang S, Bi D-W. Prediction-based dynamic energy management in wireless sensor networks. *Sensors* 2007; **7**: 251–266.
12. Wang Q, Chen W, Zheng R, Lee K, Sha L. Acoustic target tracking using tiny wireless sensor devices. In *Information Processing in Sensor Networks*, 2003.
13. Chen W, Hou JC, Sha L. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing* 2004; **3**(3): 258–271.
14. Tseng Y-C, Kuo S-P, Lee H-W, Huang C-F. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. *The Computer Journal* 2004; **47**: 448–460.
15. Yick J, Mukherjee B, Ghosal D. Analysis of a prediction-based adaptive mobility tracking algorithm. In *The 2nd International Conference on Broadband Networks*, 2005; 809–816.
16. Chhetri AS, Morrell D, Suppappola AP. Energy efficient target tracking in a sensor network using non-myopic sensor scheduling. In *The 7th International Conference on Information Fusion*, 2005.
17. Younis M, Youssef M, Arisha K. Energy-aware management for cluster-based sensor networks. *Computer Networks* 2003; **43**: 649–668.
18. Snoussi H, Richard C. Ensemble learning online filtering in wireless sensor networks. In *IEEE International Conference on Communications Systems*, 2006.
19. Teng J, Snoussi H, Richard C. Binary variational filtering for target tracking in sensor networks. In *IEEE Workshop on Statistical Signal Processing*, 2007.
20. Djuric PM, Vemula M, Bugallo MF. Target tracking by particle filtering in binary sensor networks. *IEEE*

Transactions on Signal Processing 2008; **56**(6): 2229–2238.
21. Vermaak J, Lawrence ND, Pérez P. Variational inference for visual tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2003.
22. Barndorff–Nielsen Ole E. Exponentially decreasing distributions for the logarithm of the particle size. In *Proceedings of the Royal Society, London, Series A: Mathematical and Physical Sciences*, 1977.
23. Beal MJ. Variational algorithms for approximate Bayesian inference. *Ph.D. Thesis*, University of Cambridge, 2003.
24. Tzikas DG, Likas AC, Galatsanos NP. The variational approximation for bayesian inference. *IEEE Signal Processing Magazine* 2008; **25**: 131–146.
25. Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* 2000; **10**(3): 197–208.
26. Pattem S, Poduri S, Krishnamachari B. Energy-quality tradeoffs for target tracking in wireless sensor networks. In *International Conference on Information Processing in Sensor Networks*, 2003; 32–46.
27. Onel T, Ersoy C, Delic H. Information content-based sensor selection for collaborative target tracking. In *The 14th European Singal Processing Conference*, 2006.
28. Xu Y, Qi H. Decentralized reactive clustering for collaborative processing in sensor networks. In *The 10th International Conference on Parallel and Distributed Systems*, 2004.
29. Cleju I, Fränti P, Wu X. Clustering based on principal curve. In *The 14th Scandinavian Conference on Image Analysis*, 2005; 872–881.
30. Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In *the 33rd Hawaii International Conference on System Sciences*,2000.
31. Wu H, Abouzeid AA. Error robust image transport in wireless sensor networks. In *The 5th Workshop on Applications and Services in Wireless Networks*, 2005.

AUTHORS' BIOGRAPHIES

Jing Teng was born on February 28, 1981 in Hunan, China. She received the B.Eng. degree in Electronic Information Engineering from the Central South University, China, in 2003. From 2003 to 2006, she was a graduate student in Computer Science at the Central South University. In November 2009, she received the Ph.D. degree of Systems Optimization and Security from the University of Technology of Troyes, France. Since 2010, she is a lecturer at the North China Electric Power University. Her research interests include statistical signal processing and its applications in wireless sensor networks.

Hichem Snoussi was born in Bizerta, Tunisia, in 1976. He received the diploma degree in Electrical Engineering from the Ecole Supérieure d'Electricité (Supélec), Gif-sur-Yvette, France, in 2000. He also received the DEA degree and the Ph.D. in signal processing from the University of Paris-Sud, Orsay, France, in 2000 and 2003, respectively. Between 2003 and 2004, he was postdoctoral researcher at IRCCyN, Institut de Recherches en Communications et Cybernétiques de Nantes. He has spent short periods as visiting scientist at the Brain Science Institute, RIKEN, Japan and Olin Neuropsychiatry Research Center at the Institute of Living in USA. Since 2005, he is associate professor at the University of Technology of Troyes, France. Since January 2008, he is leading a research group 'Surveillance' of LM2S laboratory. He is in charge of the regional research program S3 (System Security and Safety) of the CPER 2007–2013 and the CapSec plateform (wireless embedded sensors for security). He is the principal investigator of an ANR-Blanc project (mv-EMD), a CRCA project (new partnership and new technologies) and a GDR-ISIS young researcher project. He is partner of many ANR projects, GIS, strategic UTT programs. He obtained the national doctoral and research supervising award PEDR 2008–2012.

Cédric Richard (S'98–M'01–SM'07) was born on January 24, 1970 in Sarrebourg, France. He received the Dipl.-Ing. and the M.S. degrees in 1994 and the Ph.D. degree in 1998 from the University of Technology of Compiègne, France, all in Electrical and Computer Engineering. From 1999 to 2003, he was an Associate Professor at the University of Technology of Troyes, France. From 2003 to 2009, he was a Full Professor at the Institut Charles Delaunay (CNRS FRE 2848) at the UTT, and the supervisor of a group consisting of 60 researchers and Ph.D. In winter 2009, he was a Visiting Researcher with the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianòpolis, Brazil. Since September 2009, Cédric Richard is a Full Professor at Fizeau Laboratory (UNS, CNRS UMR 6525, Observatoire de la Côte d'Azur), University of Nice Sophia-Antipolis, France. His current research interests include statistical signal processing and machine learning.