

Article

# Distributed Information Compression for Target Tracking in Cluster-Based Wireless Sensor Networks

Shi-Kuan Liao, Kai-Jay Lai, Hsiao-Ping Tsai and Chih-Yu Wen \*

Department of Electrical Engineering, Graduate Institute of Communication Engineering,  
National Chung Hsing University, Taichung 402, Taiwan; cliffqoo@hotmail.com (S.-K.L.);  
justdoit0215@hotmail.com (K.-J.L.); hptsai@nchu.edu.tw (H.-P.T.)

\* Correspondence: cwen@dragon.nchu.edu.tw; Tel.: +886-4-2285-1549 (ext. 811); Fax: +886-4-2285-1410

Academic Editors: Lyudmila Mihaylova, Byung-Gyu Kim and Debi Prosad Dogra

Received: 19 April 2016; Accepted: 17 June 2016; Published: 22 June 2016

**Abstract:** Target tracking is a critical wireless sensor application, which involves signal and information processing technologies. In conventional target position estimation methods, an estimate is usually demonstrated by an average target position. In contrast, this work proposes a distributed information compression method to describe the measurement uncertainty of tracking problems in cluster-based wireless sensor networks. The leader-based information processing scheme is applied to perform target positioning and energy conservation. A two-level hierarchical network topology is adopted for energy-efficient target tracking with information compression. A Level 1 network architecture is a cluster-based network topology for managing network operations. A Level 2 network architecture is an event-based and leader-based topology, utilizing the concept of information compression to process the estimates of sensor nodes. The simulation results show that compared to conventional schemes, the proposed data processing scheme has a balanced system performance in terms of tracking accuracy, data size for transmission and energy consumption.

**Keywords:** information compression; target tracking; wireless sensor network

---

## 1. Introduction

With the limited capabilities in a sensor node, an important issue of target tracking is how to perform efficient information processing of the target [1]. Thus, in order to describe the target behaviors and incorporate statistical models for the task, the idea of hierarchically organizing the sensors may be used to provide network scalability and achieve energy conservation. To fulfill the task, one of the major concerns is that the original local sensed data may be pre-processed, since there may be a redundant part of it, and its data size is closely related to the energy consumption. Accordingly, the proposed scheme, the Cooperative Target tracking via Compressing Information (CTCI) algorithm, presents a new view of the result for the target tracking, which explores a new relationship between the processed data and estimation accuracy and further investigates the trade-off between the data distortion and energy consumption.

The goal of this work is to develop a distributed target tracking method by considering two perspectives: (1) load-balanced tracking and (2) improving estimation accuracy. The first perspective is to build up a load-balanced network architecture. The concept of leader-based network operation is used to automatically achieve sensor tasking, considering sensor residual energy level, target state and estimation accuracy. To avoid the ambiguity, a clusterhead and cluster members refer to the original network topology of the static cluster-based network. A leader and sub-cluster members refer to the sensor group for the tracking task. Local criteria may be used to select a cluster with the tracking responsibility. Afterwards, a sub-cluster of the corresponding cluster for the tracking task is formed by a leader, which can be a clusterhead or a cluster member in the original hierarchical

network topology. Therefore, the Level 1 clustering utilizes a static clustering scheme to build up a basic network architecture, which may eliminate the overheads of dynamic network topology control. In the presence of the target, the Level 2 clustering utilizes a dynamic clustering scheme to adaptively form a sub-cluster for the tracking task. Thus, a hybrid static/dynamic clustering scheme is applied to achieve distributed target detection and tracking.

Since the sensed data have some uncertainty and redundancy inherently, the second perspective is to investigate the target characteristics, such that the presentation of certain supplementary information can be applied to improve estimation accuracy, as well as reduce the amount of transmitted data. Within the sub-cluster, the sensing nodes provide their local estimates through Bayesian particle filtering [2] and forward the pre-processed results to the leader, considering the Bayesian estimation to extract parameters from the Gaussian noise structure.

The major contributions and key features of this work are: (1) proposing a novel cooperative positioning approach to adaptively maintain the information of measurement uncertainties; (2) developing an information compression scheme for target tracking in a two-level hierarchical wireless sensor network, which allows the selected active sensors to utilize information compression for energy conservation (e.g., using a small number of samples to roughly reconstruct the estimation distribution); (3) one of the main advantages of the Bayesian framework is that the tasking sensor carries along a complete distribution of the estimates of the target position. Thus, the processed estimation distribution is inherently a measure of the accuracy of the positioning system. With proper settings to reduce the computational burden, a particle filter can be an accurate, practical and flexible location estimation technique. Our previous work [3] proposes a low-complexity indoor tracking system using Bayesian filtering with a few samples (20 samples) in a wireless sensor network. Moreover, [4] also presents a case study of applying particle filters to location estimation for ubiquitous computing and shows that it is practical to run particle filters on devices with limited capabilities.

The rest of the paper is organized as follows: Section 2 reviews tracking systems and compares and contrasts the related works with the proposed CTCI scheme. Section 3 depicts three data processing schemes. Section 4 estimates the data size of CTCI and presents an analysis to provide a lower bound of tracking accuracy. In Section 5, the performance evaluation is depicted with varying the system parameters. Finally, we summarize the conclusions in Section 6.

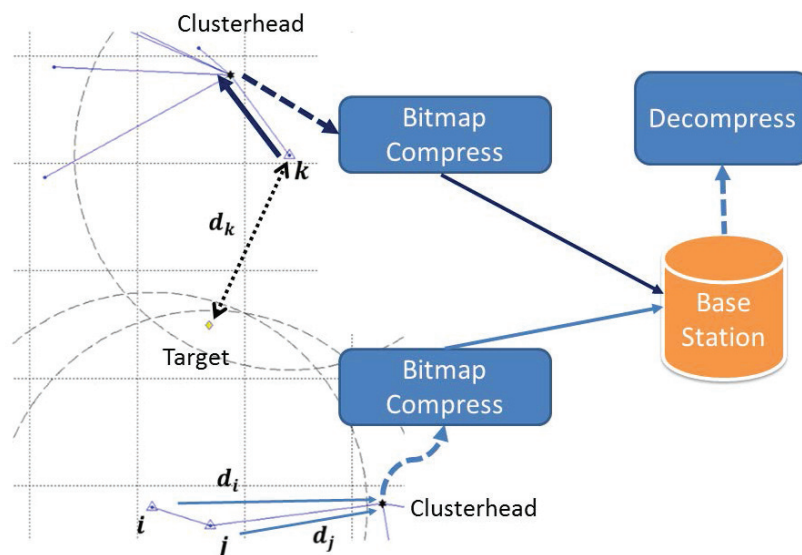
## 2. Literature Review

The authors in [5] propose a clustering-based routing protocol, Low-Energy Adaptive Clustering Hierarchy (LEACH), which distributes the energy consumption among the sensors in the network to enhance the lifetime of the network. The authors in [6] argue that their organization of clusters is triggered by the events of the monitor area, which means that the network operation is event-centric, and the clusters should be organized again according to the features of the event. The authors in [7] propose an energy-efficient distributed network scheme for phenomena detection, distinguishing between static sensors that belong to the original clusterhead and a dynamic set of sensors that belong to the phenomenon according to their readings, where the readings of the sensors are processed before transmission by using DFT for reducing the dimensionality of the data. The authors in [8–11] use the simple binary sensor for target tracking. The output of each sensor is only one bit, which means that the energy consumption of the binary sensor model is very low, but also this means there is limited information for the base station to localize the target.

The authors in [12] propose an approximation scheme, the Data Compressing-based Target Tracking Protocol (DCTTP), for target tracking with low overheads that may be practical in sensor networks. DCTTP [12] uses the distance measurements of sensor nodes to estimate the target location. The active sensors transmit their readings, including the estimated distances between them and the target and sensor ID to their clusterheads. The clusterheads gather the received data and then process the data in a centralized manner. The clusterhead processes the data by bitmap, which is a record indicating if the sensed data from a sensor can be approximated to the average value  $\mu$  of the data

gathered by the clusterhead within a given compressing bound (e.g., using the standard deviation  $\sigma$  to be the compressing bound). Therefore, if the value of a sensor reading is larger than the lower bound ( $\mu - \sigma$ ) and smaller than the upper bound ( $\mu + \sigma$ ), the data will be replaced by  $\mu$ , and the bit set to “0”, which means the data are compressed to  $\mu$ . If they are not, set the bit to “1”, and store the difference between the data and the average value for decompressing to the original data in the base station.

After scanning all of the data, there are a binary sequence (*i.e.*, data compression) and a decimal sequence (*i.e.*, the difference between the data and  $\mu$ ) generated by the bitmap in the clusterhead. Then, the clusterhead transmits the sensor ID, average value and sequences of bitmaps to the base station. The base station decompresses the received data and uses them with the position of each active sensor to estimate the location of the target. The positions of active sensors and available information (e.g., distance measurements) form a possible area for the target (shown in Figure 1), which means the target positioning will be influenced by the position of active sensors. The rationale is that if the active sensor is closer to the target, its weighting for estimation will be larger. Then, the result of position estimation will move toward the active sensor. Moreover, the number and the relative positions of the active sensors with respect to the target also influence the result of the position estimation. That is, if the active sensors are dense to a side, the position estimate will be much closer to the active sensors, and it may make the estimate far away from the target.

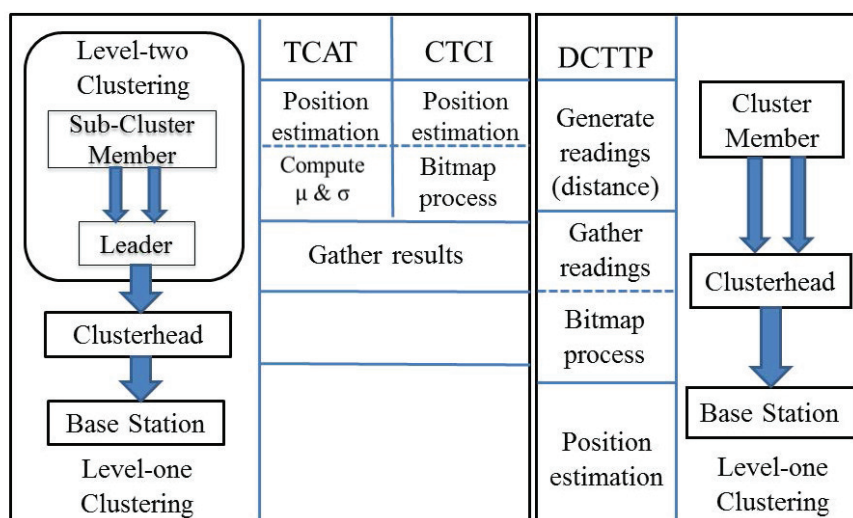


**Figure 1.** The result area of the Data Compressing-based Target Tracking Protocol (DCTTP).

In this work, for the information processing approach, we apply the concept of bitmap in [12] to compress the measurement information. The perspective of the binary sensor network inspires us to fully utilize the useful known information, which is not directly used by [13]. Accordingly, the CTCI method aims to balance the tracking performance in terms of estimation accuracy and energy consumption. In this paper, our previous work, the Two-level Clustering Approach via Timer (TCAT) [13] scheme, is used to build two-level clustering for the network architecture. Consequently, combining the Bayesian particle filter [14] and local neighboring information, the proposed CTCI scheme is able to efficiently complete the tracking task.

Here, we compare the information processing model of TCAT, CTCI and DCTTP for target tracking (shown in Figure 2). Observe that there is only a single hierarchy in DCTTP, but two hierarchies in TCAT. Thus, TCAT focuses on organizing the active sensors triggered by the target. Consequently, the system of TCAT is more stable and efficient than that of DCTTP, and TCAT can instantly acquire the result of target tracking at Level 2 clustering. Although there are many protocols proposed for

target tracking, the network architecture and information processing problems are not considered simultaneously [7–11,15–17]. Hence, we tackle the tracking task and the data processing problem based on the network architecture in TCAT. Even though the energy consumption and the data size of CTCI are slightly higher than that of TCAT and DCTTP in some cases, the system of CTCI is able to adaptively provide the measurement distribution via information compression, not just describing the target position with a single estimate. Notice that the TCAT scheme can be considered as a special case of CTCI with a large compression bound, which compresses the samples to the mean. Moreover, the estimation accuracy and the integrality of the tracking result of CTCI are better than that of TCAT (improved by 20%) and are much better than that of DCTTP (improved by 100%). We investigate the trade-off between the tracking quality and the energy consumption in Section 5.



**Figure 2.** The information processing model of Two-level Clustering Approach via Timer (TCAT) (left), Cooperative Tracking via Compressing Information (CTCI) (middle) and DCTTP (right).

### 3. Data Processing Schemes

This section discusses the proposed data processing scheme, Cooperative Tracking via Compressing Information (CTCI), including the tracking method, information processing model, information flow and the meaning of the location estimation. First, we introduce the network architecture and TCAT, which is the basic model of the proposed CTCI data processing scheme. Secondly, we compare and contrast the DCTTP and TCAT schemes. Finally, a new data processing scheme, CTCI, is proposed to balance the distortion of the processed data and the energy consumption.

#### 3.1. Problem Statement

Figure 3 shows an example of target movement with 25 time steps. In the presence of the target, to achieve load-balancing tracking, an important issue of target tracking is how to collect the desired data and to perform efficient information processing. Denote a sensor with tracking responsibility as an active sensor. Otherwise, a sensor is marked as an inactive sensor. Thus, referring to the Level 1 clustering in Figure 3, the Level 2 clustering utilizes a dynamic clustering scheme to adaptively form a local sub-cluster with active sensors for the tracking task in a distributed manner (Figure 4). Since there is a trade-off between estimation accuracy and energy consumption, this work aims to investigate the network performance on data size and energy conservation for target tracking from an information processing perspective. The main assumptions are: (1) all sensor nodes are assumed to be homogeneous with their fixed position information in a strongly-connected network; (2) the noise model is the zero-mean Gaussian noise measurement model; (3) Angle-of-Arrival (AOA) information or hybrid Time-of-Arrival (TOA)/AOA information is applied to perform target tracking; and (4) the

target broadcasts a message for measurement purposes periodically. Note that these assumptions may be applied to locate objects (e.g., patients or animals) in healthcare or habitat monitoring scenarios.

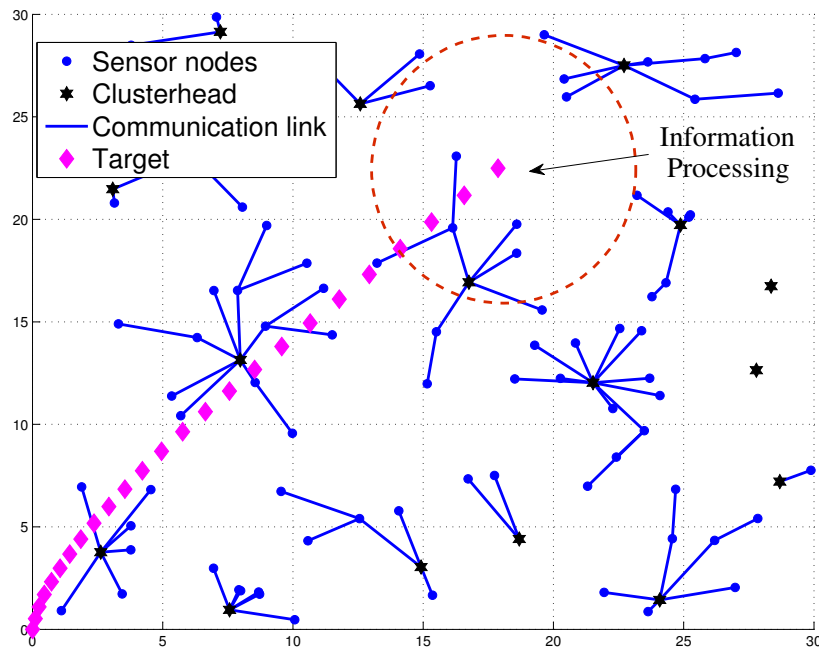


Figure 3. The cluster-based network topology and target movement with 25 time steps.

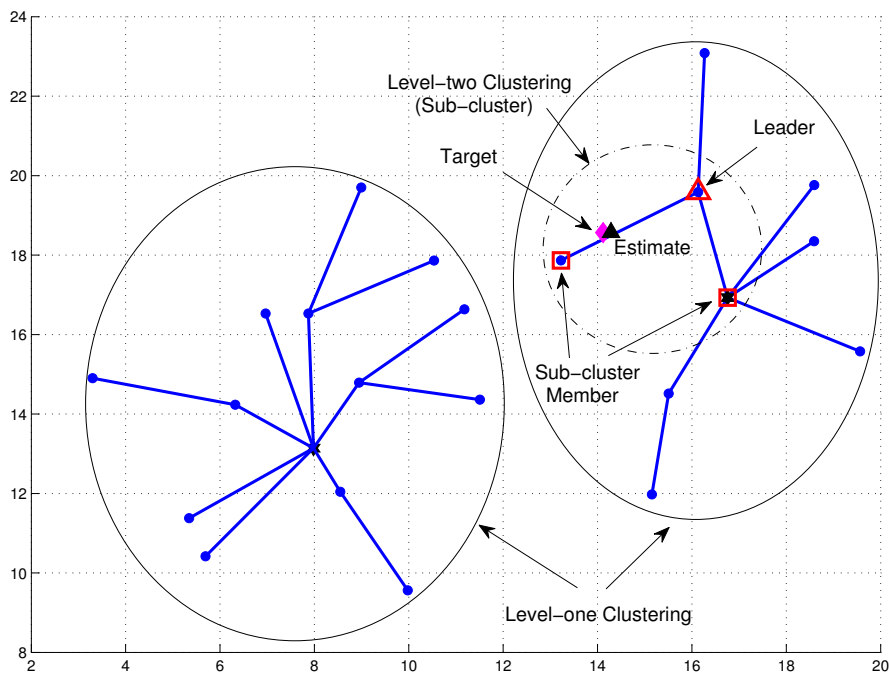
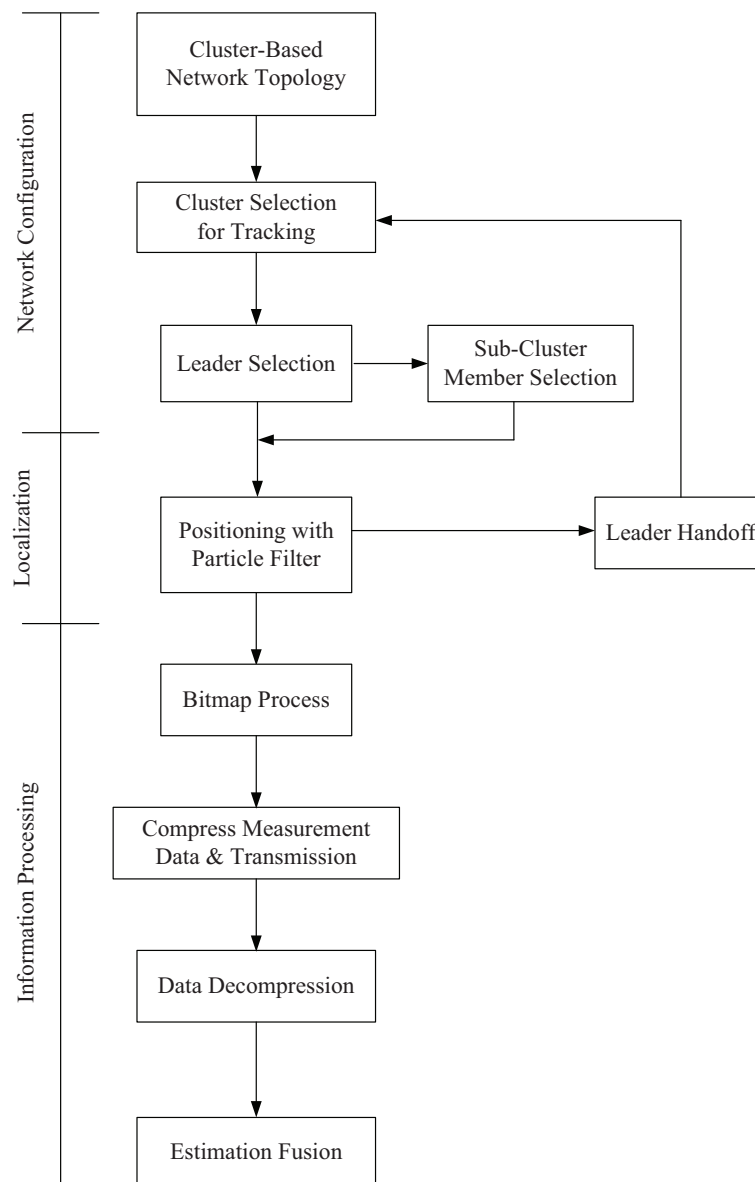


Figure 4. An example of Level 2 clustering for the tracking task.

Accordingly, as depicted in Figure 5, the CTCI scheme performs target tracking in three major phases: (1) network configuration; (2) target positioning; and (3) information processing. The following subsections describe each phase, respectively.



**Figure 5.** Illustration of the block diagram for the CTCI method.

### 3.2. Network Configuration and TCAT

This section describes the fundamental system configuration of CTCI. To achieve distributed scheduling target detection and tracking, a two-level clustering (*i.e.*, a hybrid static/dynamic clustering scheme) is applied to deal with the control of network operations and to handle the tracking task, which is based on our previous works, the Clustering Algorithm via Waiting Timer (CAWT) scheme [18] and the TCAT scheme [13]. During the network initialization phase, sensors may apply the CAWT to establish a Level 1 network architecture (Figure 3). In order to handle the impact of the target movement, the TCAT scheme may be applied to form a sub-cluster with tracking responsibility and allow each sub-cluster member to locally locate the target in a distributed manner. Figure 4 shows an example of leader and sub-cluster member selection. Therefore, the TCAT scheme may be applied to select the sensors with tracking responsibility and build a Level 2 network architecture. Notice that the information flow is bi-directional (*i.e.*, it goes through the sub-cluster members to the leader and then to the clusterhead, and vice versa).

### 3.3. Target Positioning

The major difference between TCAT and CTCI is the bounding area formed by the different numbers of active sensors. In TCAT, the bounding area is determined by the active sensors in the same cluster. In CTCI, the bounding area is determined by not only the active sensors in a single cluster, but also the active sensors in other clusters. It is worth mentioning that including the active sensors in different clusters can effectively decrease the scope of the bounding area for sampling. The bigger the distance between the active sensor and the others, the smaller the intersection area of the transmission range of the active sensor will be, which suggests that we can provide a better sample area for particle filtering and even use fewer sampling points for the tracking task as the sample area decreases. Observe that in Figure 6, we can see that an active sensor node is added to form the bounding area A in CTCI, which is smaller than the bounding area B in TCAT (as shown in Figure 7). Accordingly, we may use a smaller number of samples (Figure 8) to achieve a comparable estimation accuracy with particle filtering.

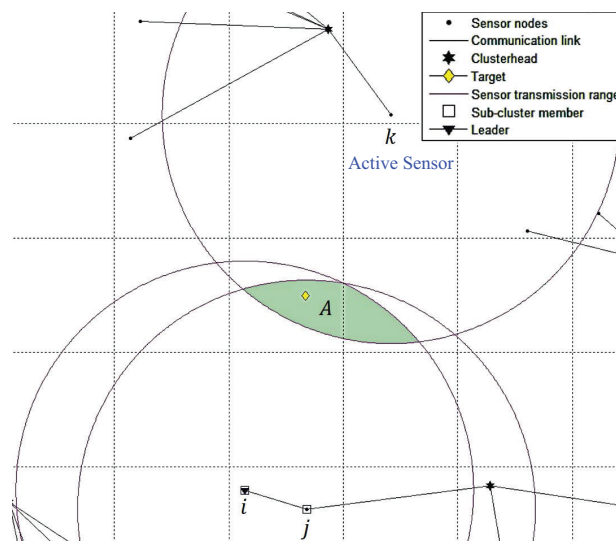


Figure 6. The bounding area A of CTCI.

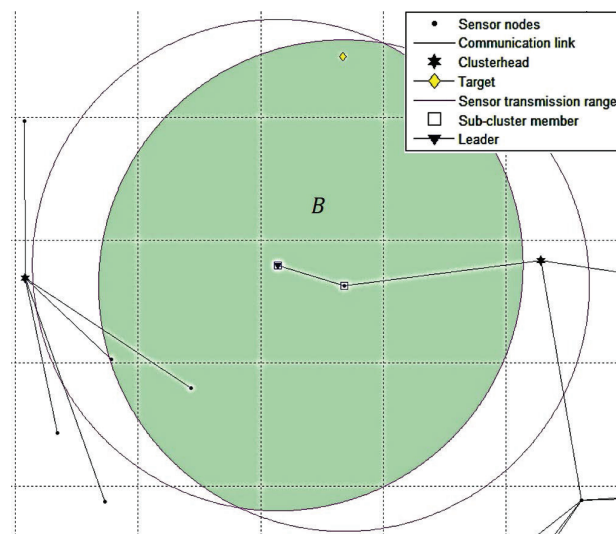
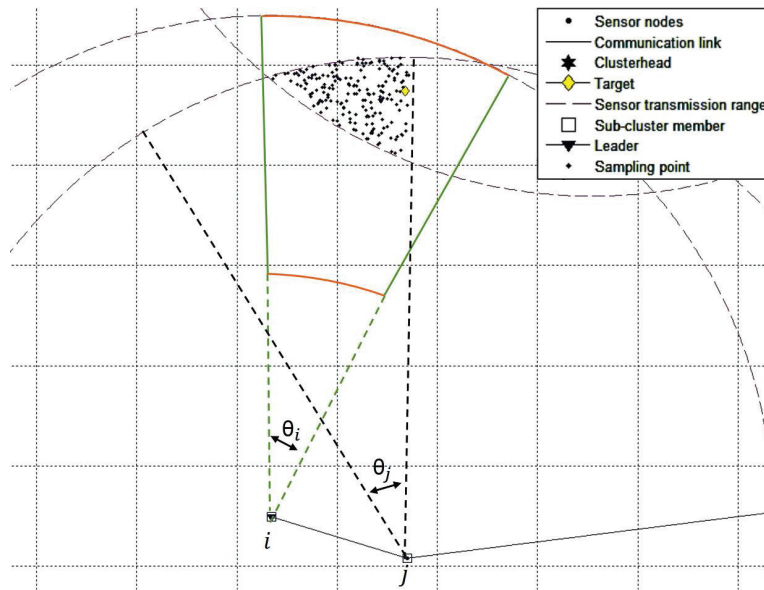


Figure 7. The bounding area B of TCAT.



**Figure 8.** The intersection of sensor measurements and the area A, which leads to the distribution of the sampling points of CTCL.

On the basis of the refined sample area, the position estimation method is as follows:

1. Create a random sample  $x_k(q)$ ,  $q = 1, 2, \dots, N$  from the refined sample area at time step  $k = 0$ .
2. Each random sample is passed through the state Equation (1) to obtain samples at time step  $k + 1$ .

$$\hat{x}_{k+1}(q) = \Phi x_k(q) + \Gamma \lambda_k(q) \quad (1)$$

where the system noise  $\lambda_k(q)$  is a sample drawn from the Gaussian noise,  $\Phi$  is characterized by the mobility model and  $\Gamma$  is assumed to be an identity matrix.

3. Upon receipt of the noisy measurement  $z_{k+1}$ , evaluate the likelihood of each prior sample and obtain the normalized weight of each sample. Therefore, after updating the weights of the likelihood function  $p(z_{k+1}|\hat{x}_{k+1}(q))$  for each sample, the normalized weights yield:

$$\gamma_{k+1}(q) = \frac{p(z_{k+1}|\hat{x}_{k+1}(q))}{\sum_{j=1}^N p(z_{k+1}|\hat{x}_{k+1}(j))} \quad (2)$$

4. Based on the normalized weights, it generates a new sample set  $x_{k+1}(q)$  by performing sampling with replacement from the set  $\hat{x}_{k+1}(q)$  ( $q = 1, 2, \dots, N$ ), which describes the distribution of the position estimate.

Note that the noise model is the zero-mean Gaussian noise measurement model. A typical setting of the prior parameter values is described in Section 5.1.1. For obtaining the distribution of position estimate, each tasking sensor sends the mean and variance of the distribution to the tasking leader.

#### 3.4. Information Processing and Data Management

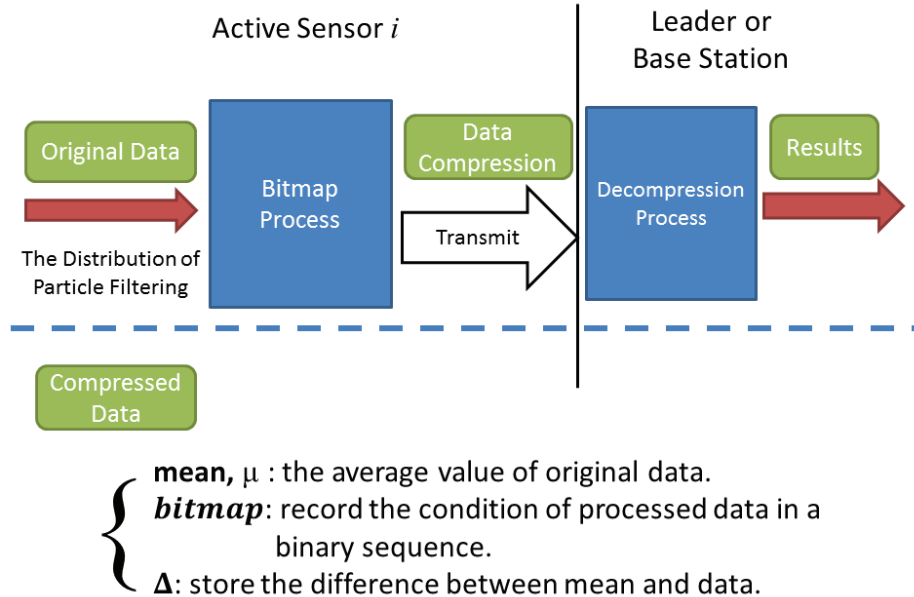
Figure 9 depicts the conceptual procedures of information processing. The concepts and techniques (in the upper row) represent the processing mechanisms of active sensors and the leader, respectively. Upon obtaining the distribution of the target position by using a particle filter in an active sensor, data analysis is performed to explore the distribution properties, execute information compression and then transmit the processed data to a leader (or a base station). Afterwards, a leader (or a base station) may apply the received processed data (in the lower row) to generate a distribution,



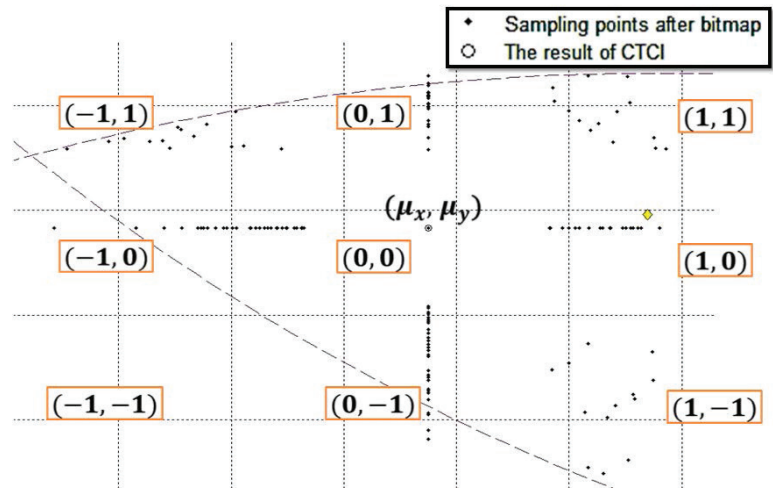
which roughly corresponds to the estimation distribution. The proposed algorithm uses the bitmap process and statistical properties to execute data processing, which consists of the following five steps:

1. *Preliminary settings*  $\sim$  Each tasking sensor performs target positioning with Bayesian filtering of  $N$  samples, which leads to a target position estimate at time step  $k$ ,  $P_k = (X_k, Y_k)$ , where  $X_k = (x_k(1), x_k(2), \dots, x_k(N))$ ,  $Y_k = (y_k(1), y_k(2), \dots, y_k(N))$ .
2. *Configuring the bitmap of  $P_k$*   $\sim$  Each tasking sensor computes the average  $\bar{\mu}_k = (\bar{X}_k, \bar{Y}_k)$ ,  $\sigma_{X_k}$ ,  $\sigma_{Y_k}$ , and generates the value of compressing bound,  $\varepsilon_{X_k} = \sigma_{X_k} \cdot \eta$  and  $\varepsilon_{Y_k} = \sigma_{Y_k} \cdot \eta$ , where  $\eta$  is an adjustable factor to trade off the compression ratio and positioning accuracy.
3. *Compression of  $(X_k, Y_k)$*   $\sim$  As shown in Figure 10b,c, scan and compress  $X_k$  and  $Y_k$  (using  $x_k(j)$  and  $y_k(j)$  as examples). Hence, if  $(|x_k(j) - \bar{X}_k| > \varepsilon_{X_k})$ , then  $\text{bit}(x_k(j)) = 1$  and  $\Delta_{x_k(j)} = x_k(j) - \bar{X}_k$ . Similarly, if  $(|y_k(j) - \bar{Y}_k| > \varepsilon_{Y_k})$ , then  $\text{bit}(y_k(j)) = 1$  and  $\Delta_{y_k(j)} = y_k(j) - \bar{Y}_k$ . Afterwards, the tasking sensor (*i.e.*, a sub-cluster member) transmits the compressed data  $(\bar{\mu}_k, \Delta(X_k), \Delta(Y_k), \sigma_{X_k}, \sigma_{Y_k})$ .
4. *Data decompression*  $\sim$  As shown in Figure 10a, decompress the received data by scanning  $\bar{\mu}_k$ ,  $\Delta(X_k)$  and  $\Delta(Y_k)$ . If  $\text{bit}(x_k(j)) = 1$ , then  $x_k(j) = \bar{X}_k + \Delta_{x_k(j)}$ ; otherwise,  $x_k(j) = \bar{X}_k$ .
5. *Estimation fusion*  $\sim$  Figure 11 shows the external contour of the sample region and the distribution of the decompressed data. The leader node scans the information of each sample sub-region. Let  $p_m$  be the target appearance probability of sample region  $m$ , which yields  $p_m = N_m/N_L$  ( $m = 1, 2, \dots, 9$ ), where  $N_m$  is the number of sample points in region  $m$  and  $N_L$  is the total number of sample points. Let  $\bar{\mu}_k(m)$  be the average estimate of sample region  $m$ . Accordingly, we obtain the location estimate  $P_L = \sum_{m=1}^9 p_m \cdot \bar{\mu}_k(m)$ .

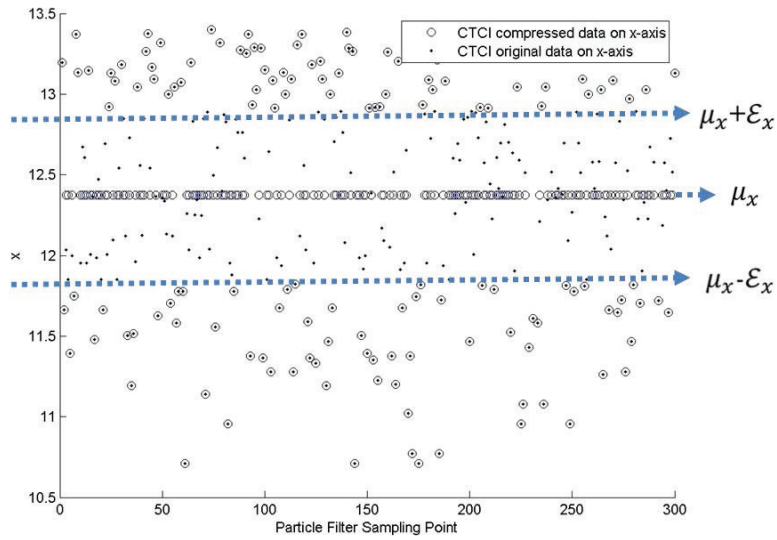
Therefore, the information processing of the CTCI scheme is completed.



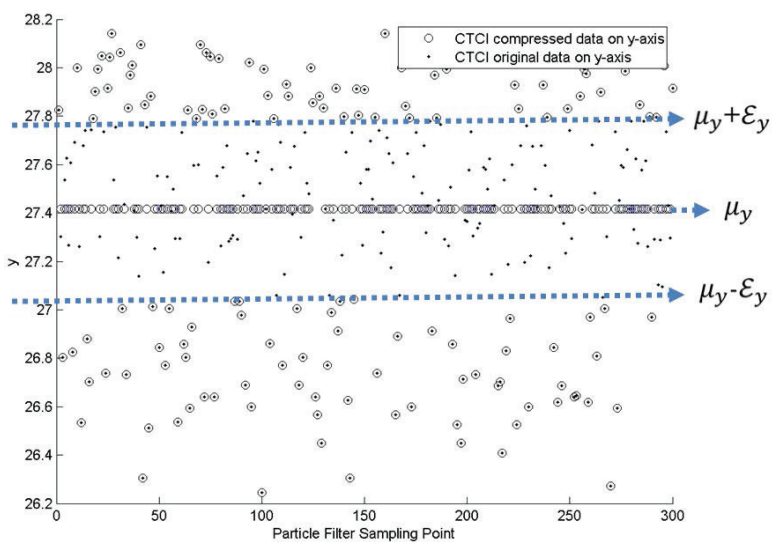
**Figure 9.** The conceptual processes of the CTCI scheme.



(a)



(b)



(c)

**Figure 10.** The sampling points after the bitmap process (a) and the impact of the compressing bound on distortion:  $x$ -axis (b) and  $y$ -axis (c).

Compared to DCTTP (a one-dimensional data processing method), the bitmap process in CTCI is a two-dimensional data processing method. Thus, the sampling points after the bitmap process will have several combinations for the  $x$  and  $y$  coordinates. From Figure 10a, we observe three basic combinations of the sampling points: (1) (0,0) means both the values of the  $x$  and  $y$  coordinates are approximated to the average value at the same time; (2) (1,0) or (0,1) means a value of the  $x$  or  $y$  coordinate is approximated to the average value, but the other one is not; (3) (1,1) means both values are not represented by the average value. The minus “-” means that the value of the  $x$  or  $y$  coordinate is smaller than the average value. Thus, the sample region is divided into nine parts by the bitmap process. In Figure 10b,c, the distributions of the original and compressed sampling points show the distortion of the  $x$  and  $y$  coordinates, respectively. The data within the compressing bound are represented to the average value, which causes the distortion. Thus, there is a trade-off between the energy consumption and the extent of distortion.

Therefore, each sub-cluster member transmits the compressed data, including (1) the average value and the standard deviation of the original sample points and (2) the differences between the sample points and the average value, to the leader node. Afterwards, the leader may transmit the position estimate to the base station through clusterheads and gateways in a multi-hop manner or directly disseminate the data from the clusterhead to the base station. In the leader node, the processed data are decompressed, and the external contour of the sample region is depicted by recovering the original sample points associated with (1,1), (-1,1), (1,-1), (-1,-1) representations as the results for data processing. The meaning of the result is shown in Figure 11, which means that the result of the target positioning is not only just a position estimation, but also a similar area of the original region for sampling. After obtaining the distribution of the position estimate, each tasking sensor sends the mean and variance of the distribution to the tasking leader.

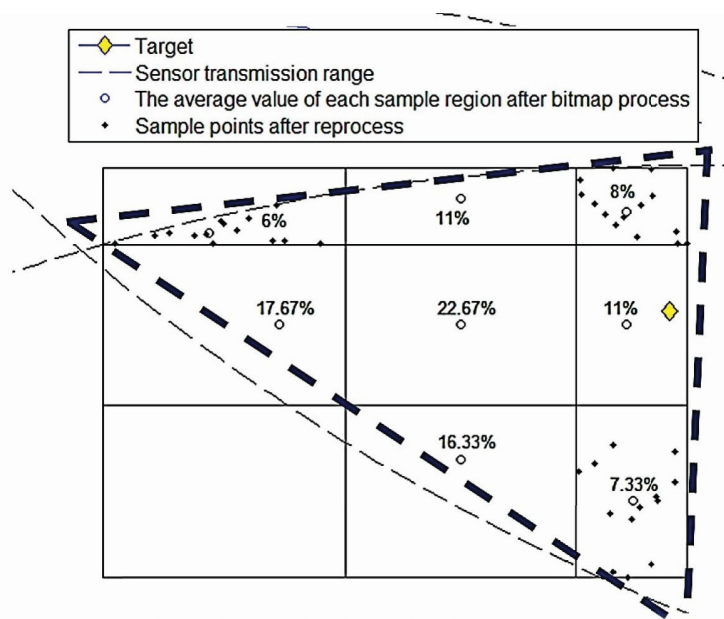


Figure 11. The external contour of the sample region and the distribution of the decompressed data.

#### 4. Performance Analysis

This section performs an analysis of the data size and describes a conceptual-stochastic distribution of the position estimation.

#### 4.1. Analysis of the Data Size

In order to assess the energy consumption during the data processing procedures, the data size is estimated, assuming that the sensor nodes transmit data in a binary system. Instead of using the floating point representation like IEEE standards, which reserve bits for exponents to represent a wider range of numbers, we use fixed point representation, which may cost less bits than floating point representation. This is because we only transmit the compressed data, *i.e.*,  $\Delta x_i$  and  $\Delta y_i$ , which are the differences of  $x_i$  and  $\bar{X}_i$  and  $y_i$  and  $\bar{Y}_i$ , respectively. Since the values of the differences are usually small, our algorithm can dynamically decide the number of bits for the integer part and reserve more bits for the fractional part, depending on the value and the required precision. Specifically, before transmitting the data stream, active sensor nodes scan the data stream first to find the maximum of it, which decides the largest bit number of the integer part and the position of the decimal point.

For instance, in Table 1, the first bit represents the sign of the value; the second bit to fourth bit specify the position of the decimal point; the fifth to seventh bit represent the integer part of the value; the eighth to 16th bit represent the fractional part of the value; the code line is the bit stream to represent  $-5.678$ . Thus, if the required precision is to  $2^{-5}$ , we only use 12 bits to represent a value in that case. Observe that, in the example of Table 1, using IEEE standard with nine-bit mantissa to represent the difference leads to an error between the original value and the value after processing lower than  $2^{-7}$ . In contrast, our approach can achieve better precision with an error lower than  $2^{-9}$ .

**Table 1.** An example for the fixed point representation.

Value	−5.678																
Code	1	1	0	0	1	0	1	1	0	1	0	1	1	0	1	1	
Meaning	$\pm$	$2^2$	$2^1$	$2^0$	$2^2$	$2^1$	$2^0$	.	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	$2^{-8}$	$2^{-9}$
Bit list	1	2	3	4	5	6	7		8	9	10	11	12	13	14	15	16

#### 4.2. Analysis of Positioning Accuracy

As the Cramer–Rao lower bound (CRLB) tests the performance of the proposed measurement mechanism, this subsection describes the achievable tracking accuracy of joint TOA/AOA positioning. The measurements at the reference sensor can be modeled as:

$$\hat{\tau} = \tau + \delta_{\tau} \quad (3)$$

$$\hat{\phi} = \phi + \delta_{\phi} \quad (4)$$

where  $\tau$  is the true propagation time and  $\phi$  is the true angle information. Note that  $\delta_{\tau}$  and  $\delta_{\phi}$  are uncorrelated Gaussian noises with the distributions  $\delta_{\tau} \sim \mathcal{N}(0, \sigma_{\tau}^2)$  and  $\delta_{\phi} \sim \mathcal{N}(0, \sigma_{\phi}^2)$ . Assuming that the direct path exists between the sensor and the target, the estimated target position is given by:

$$\hat{x} = x_s + v\hat{\tau} \cos(\hat{\phi}) = x_s + \hat{r} \cos(\hat{\phi}) \quad (5)$$

$$\hat{y} = y_s + v\hat{\tau} \sin(\hat{\phi}) = y_s + \hat{r} \sin(\hat{\phi}) \quad (6)$$

where  $\hat{r}$  is the distance measurement (*i.e.*,  $\hat{r} = v\hat{\tau} = r + v\delta_{\tau}$ ),  $(x_s, y_s)$  is the true position of the sensor and  $v$  is the speed of the signal. Assuming  $\delta_{\tau}$  and  $\delta_{\phi}$  are sufficiently small, the variance of the position estimation  $\hat{p}$  is approximated by:

$$\sigma_p^2 \approx v^2 \sigma_{\tau}^2 + d^2 \sigma_{\phi}^2 = \sigma_r^2 + d^2 \sigma_{\phi}^2 \quad (7)$$

Given the above assumptions [19], the CRLB with a single sensor is derived as follows.

Assuming that a line-of-sight path exists between a sensor and the target, the joint density function of the distance measurement  $\hat{r}$  and the angle measurement  $\hat{\phi}$  yields:

$$f(\mathbf{g}; x, y) = \frac{1}{\sqrt{2\pi\sigma_r^2}} \cdot \exp\left[-\frac{1}{2\sigma_r^2} (\hat{r} - d)^2\right] \cdot \frac{1}{\sqrt{2\pi\sigma_\phi^2}} \cdot \exp\left[-\frac{1}{2\sigma_\phi^2} \left(\hat{\phi} - \arctan\left(\frac{y-y_s}{x-x_s}\right)\right)^2\right] \quad (8)$$

where  $\mathbf{g} = [\hat{r}, \hat{\phi}]$ ,  $(x_s, y_s)$  is the true sensor position and  $d$  is the true distance between pairwise sensors. Thus, the Fisher information matrix is given by:

$$I(\mathbf{x}^{(t)}) = \begin{bmatrix} \frac{\cos^2(\phi)}{\sigma_r^2} + \frac{\sin^2(\phi)}{d^2\sigma_\phi^2} & \frac{\sin(2\phi)}{2} \left[ \frac{1}{\sigma_r^2} - \frac{1}{d^2\sigma_\phi^2} \right] \\ \frac{\sin(2\phi)}{2} \left[ \frac{1}{\sigma_r^2} - \frac{1}{d^2\sigma_\phi^2} \right] & \frac{\sin^2(\phi)}{\sigma_r^2} + \frac{\cos^2(\phi)}{d^2\sigma_\phi^2} \end{bmatrix} \quad (9)$$

which leads to the CRLB for the best possible estimation for target tracking. Figure 12 shows a typical run of the conceptual-stochastic distribution of the position estimation. The readers may refer to [19] for details.

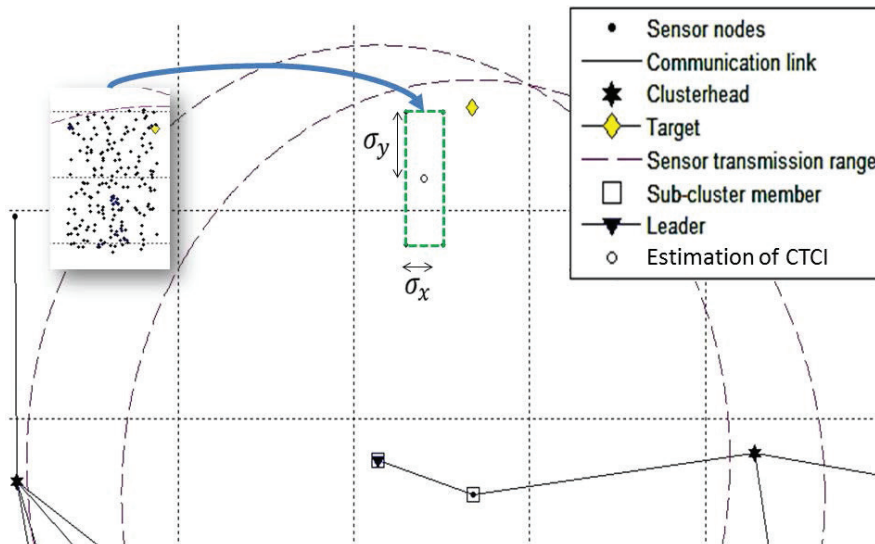


Figure 12. The stochastic distribution of position estimation.

## 5. Simulation

This section describes the simulation settings and discusses the influence of simulation parameters on the tracking task. The simulation is conducted with various experimental settings (e.g., different network density, the communication range and network topology). The MATLAB© (MathWorks, Natick, MA, USA) simulation tool is applied to perform the comparison among our proposed approach CTCI, DCTTP [12] and TCAT [13]. The following subsections respectively describe the parameters, mobility model and energy consumption model used.

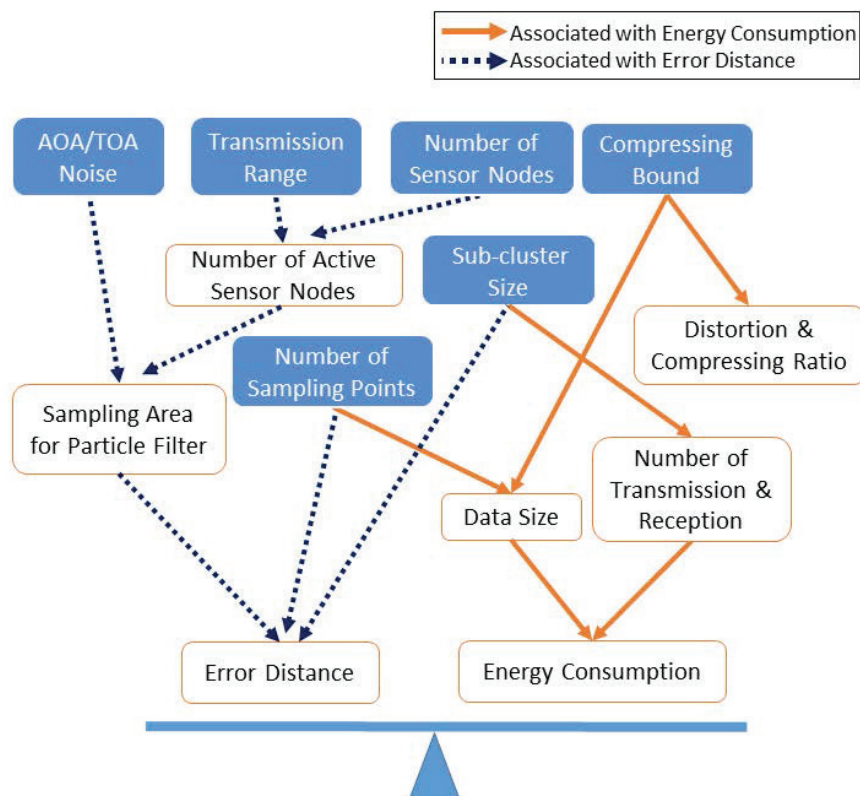
### 5.1. The Parameters

As illustrated in Figure 13, system parameters fall basically into two categories: Category I is associated with distance estimation and tracking: (a) AOA/TOA measurement noise; (b) transmission range; (c) number of sensor nodes; (d) compressing bound; (e) number of sub-cluster size; and (f) number of sampling points; Category II is associated with energy conservation: (1) the number

of active sensor nodes; (2) the sampling area for the particle filter; (3) the distance estimation error; (4) distortion and compressing ratio; (5) data size; (6) the number of transmissions and receptions; and (7) the energy consumption. Figure 13 shows the impacts of system parameters on the tracking performance, such as distance estimation and energy conservation. Table 2 depicts the values of the simulation parameters for the target position estimate.

**Table 2.** The values of the simulation parameters.

Parameter	Value
The sensing field	$100 \times 100 \text{ m}^2$
Base station location	(50,50)
Number of sensors	200
Transmission range	15 m
Number of samples	15
Sub-cluster size $n$	2
TOA measurement deviation $\sigma_w$	2
AOA measurement deviation $\sigma_\theta$	2
Compressing bound parameters	0.8



**Figure 13.** The impacts of system parameters on the tracking performance from two perspectives: (1) distance estimation and (2) energy conservation.

### 5.1.1. Mobility Model

Suppose that the target movement within the  $x - y$  sensing field is based on the standard second-order model [14]:

$$X_k = \Phi X_{k-1} + \Gamma w_k \quad (10)$$

over a four-dimensional state space, where  $X_k = (x, \dot{x}, y, \dot{y})_k^T$ ,  $\mathbf{w}_k = (w_x, w_y)_k^T$ . Note that  $x$  and  $y$  denote the Cartesian coordinate of the target, and the uncertainties are described by uncorrelated Gaussian diffusion terms,

$$\Phi = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and } \Gamma = \begin{pmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{pmatrix}$$

Assuming that the measurement is contaminated by noise, the noisy measurement is represented by:

$$z_k = \tan^{-1}(y_k/x_k) + v_k \quad (11)$$

where the measurement noise,  $v_k$ , is a zero mean Gaussian white noise process with a finite variance  $\sigma_v^2$ . Moreover, the system initial state is given by a Gaussian distribution with known mean  $\bar{x}_1$  and covariance:

$$M_1 = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix}$$

The target trajectory monitoring and measurements are generated based on Equations (10) and (11). The values of the parameters are the following: the covariance of the system noise,  $Q = qI_2$ , where  $I_2$  is the  $2 \times 2$  identity matrix,  $\sqrt{q} = 0.001$ . The target initial state is  $x_1 = (0.0, 0.6, 0.0, 2.5)^T$ . Referring to the settings in [14], the prior parameter values are set to  $\bar{x}_1 = (0.0, 0.0, 0.4, 0.05)^T$ ,  $\sigma_1 = 0.5$ ,  $\sigma_2 = 0.001$ ,  $\sigma_3 = 0.05$  and  $\sigma_4 = 0.01$ .

### 5.1.2. Energy Consumption Model

In order to further assess the network performance, we explore the communication cost for the tracking task. Since the computational cost in a wireless sensor network is usually neglected compared to the communication cost, we focus on the analysis of power consumption for communication. Currently, there have been a number of studies on measuring the energy dissipation per transmitted bit [5,20]. Here, we adopt the energy model presented in [5] to describe the energy dissipation. Thus, the energy dissipation of a transmitter  $E_{Tx}$  and a receiver  $E_{Rx}$  can be modeled as follows [5]:

$$E_{Tx} = \begin{cases} tE_{elec} + t\varepsilon_{fs}d^2, & d < d_o \\ tE_{elec} + t\varepsilon_{mp}d^4, & d \geq d_o \end{cases} \quad (12)$$

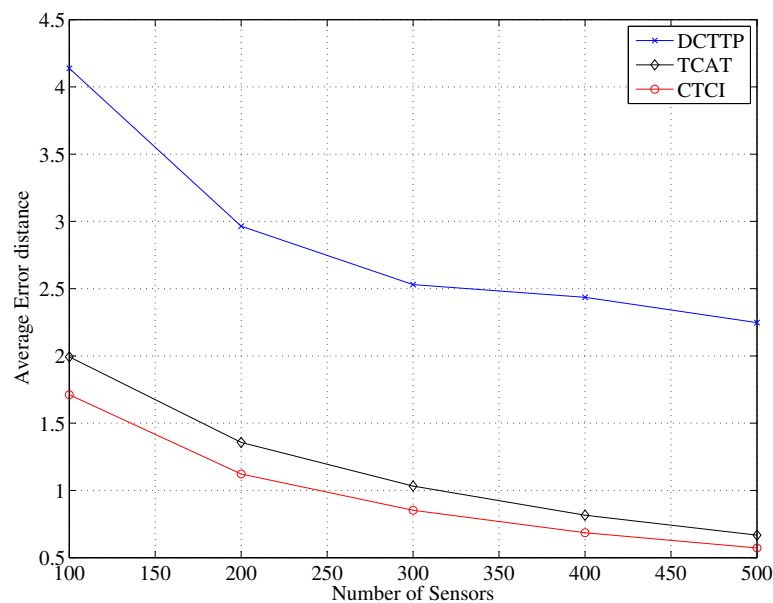
$$E_{Rx} = tE_{elec} \quad (13)$$

where  $t$  is the data packet size,  $E_{elec}$  denotes the energy consumption of the electronic circuitry,  $\varepsilon_{fs}$  and  $\varepsilon_{mp}$  depend on distance  $d$  between the transmitter and the receiver for maintaining an acceptable bit-error rate and  $d_o$  is a transmission threshold. The values of the simulation parameters are set as [5]:  $E_{elec} = 50$  nJ/bit,  $\varepsilon_{fs} = 10$  pJ/bit/m<sup>2</sup>,  $\varepsilon_{mp} = 0.0013$  pJ/bit/m<sup>4</sup>,  $d_o$  is the transmission range  $R$ . Observe that the key to the energy consumption is the transmission distance and the data size. In our case, given a fixed transmission distance, how to reduce the data size for transmission becomes an important issue.

### 5.2. Number of Sensor Nodes

Figure 14 depicts the error distance of each scheme, which decreases as the number of the sensor nodes increases, which implies that if there are more active sensor nodes participating in target tracking, a larger amount of sensed data will be generated. For DCTTP with a larger number of active sensor nodes surrounding the target, the result of the target tracking will be improved. Because DCTTP

uses centroid tracking, the base station receives the distance estimation of each active sensor node, (*i.e.*, the distance between each active sensor node and the target) and then uses the position of the active sensor nodes for target tracking. If the active sensor nodes are closer to the target and uniformly distributed, the performance of target tracking can be improved. On the other hand, if active sensor nodes are distributed in a certain direction, the result of the target tracking will move toward where the distributed active sensor nodes are, not to the target position. Therefore, the error distance of DCTTP may decrease with an increasing sensor density in the network.



**Figure 14.** Impact of the number of sensor nodes on average error distance.

In TCAT, with a higher sensor density (*i.e.*, a larger number of active sensor nodes in the cluster), the bounding area for sampling will be smaller, which leads to a lower error distance of TCAT. Comparing the method of target tracking of TCAT to that of DCTTP, TCAT uses multiple sensors with TOA/AOA information and a particle filter, and DCTTP just uses the distance estimation and the position of active sensor nodes. As shown in Figure 14, the error distance of TCAT is much lower than the error distance of DCTTP. In the proposed CTCI, with a larger sensor density, the bounding area for sampling points will be refined to a smaller region compared to TCAT. This is because the location information of active sensors can be used to further refine the sampling area. Therefore, the error distance of CTCI is lower than the error distance of TCAT.

Figure 15 depicts that the data size of DCTTP rapidly rises as the number of sensor nodes increases. Observe that when the number of sensor nodes is about 230, the data size of DCTTP is larger than that of TCAT. This is because the DCTTP does not have a constraint on the number of active sensors that participate in the tracking task. In contrast, the data size of TCAT and CTCI is nearly constant because of the pre-defined sub-cluster size. As shown in Figure 16, the distribution of the average energy consumption is similar to the performance of the average data size. Observe that with a high network density (e.g., the number of sensor nodes = 500), the energy consumption of CTCI is larger than that of TCAT and lower than that of DCTTP.



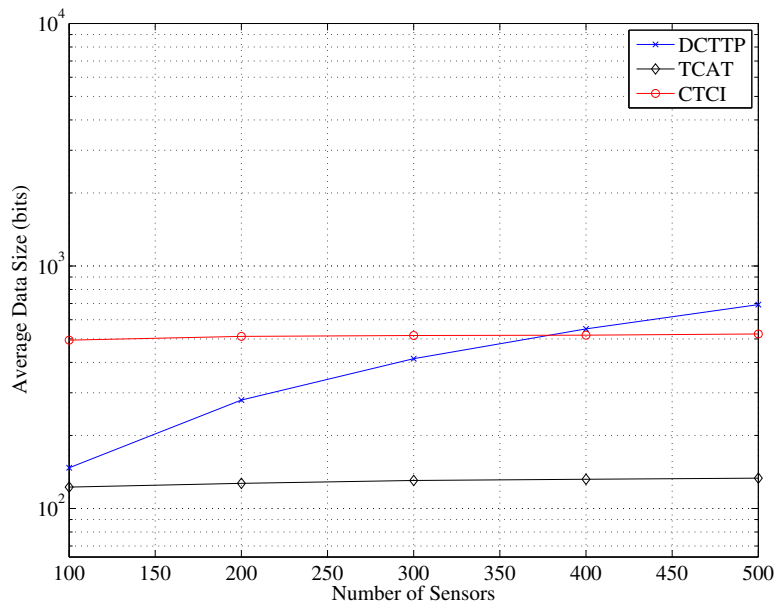


Figure 15. Impact of the number of sensor nodes on the average data size.

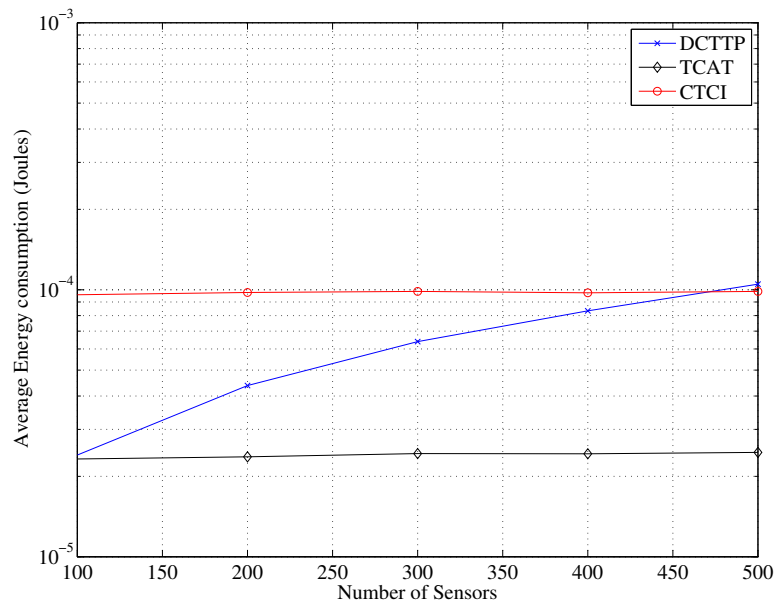


Figure 16. Impact of the number of sensor nodes on the average energy consumption.

### 5.3. Transmission Range

Figure 17 depicts that the error distance of TCAT and CTCI decreases, but the error distance of DCTTP increases rapidly as the transmission range increases. Because the location method of DCTTP is centroid positioning, the result of the positioning will be influenced by the position of the active sensor nodes. When the transmission range increases, there are more active sensor nodes that are far from the target. The result of the positioning is far away from the target; therefore, the error distance of DCTTP is larger.

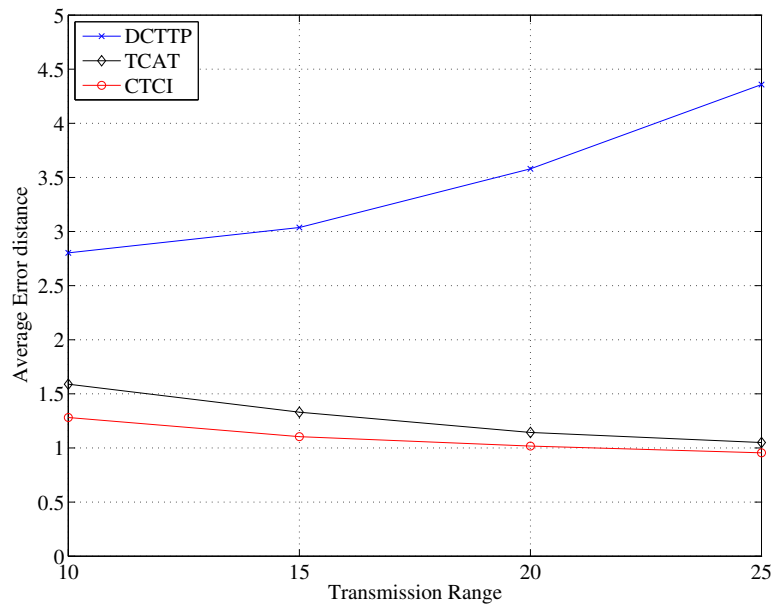


Figure 17. Impact of transmission range on the average error distance.

In TCAT, the cluster size increases as the transmission range increases. For position estimation of TCAT, a larger cluster size may result in a larger number of active sensor nodes in the cluster to which the leader belongs. Therefore, the error distance decreases as the bounding area for sampling is smaller. Similarly, there are more active sensor nodes participating in refining the bounding area in CTCI than in TCAT, so the error distance of CTCI is smaller. Figure 18 depicts that the data size and the energy consumption increase rapidly as the transmission range increases. Figure 19 shows that when the transmission range is larger than 25, the energy consumption of CTCI is larger than that of TCAT and smaller than that of DCTTP.

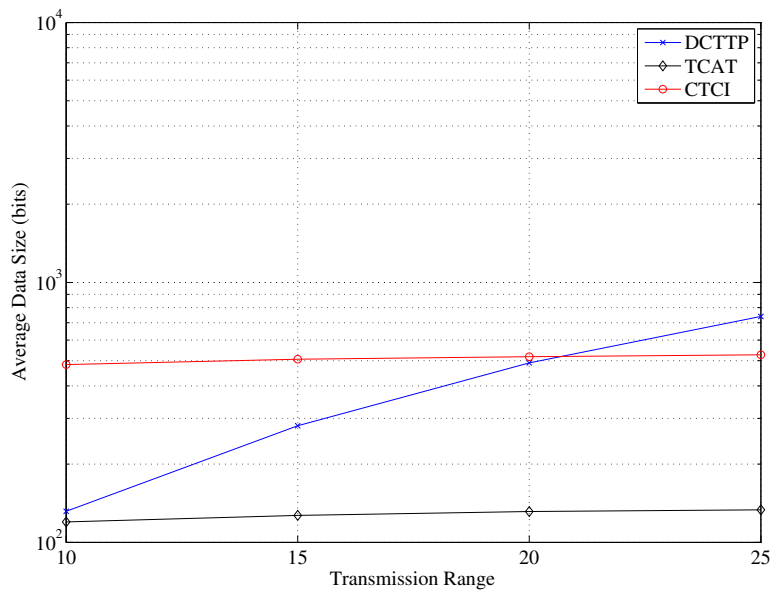


Figure 18. The impact of the transmission range on the average data size.

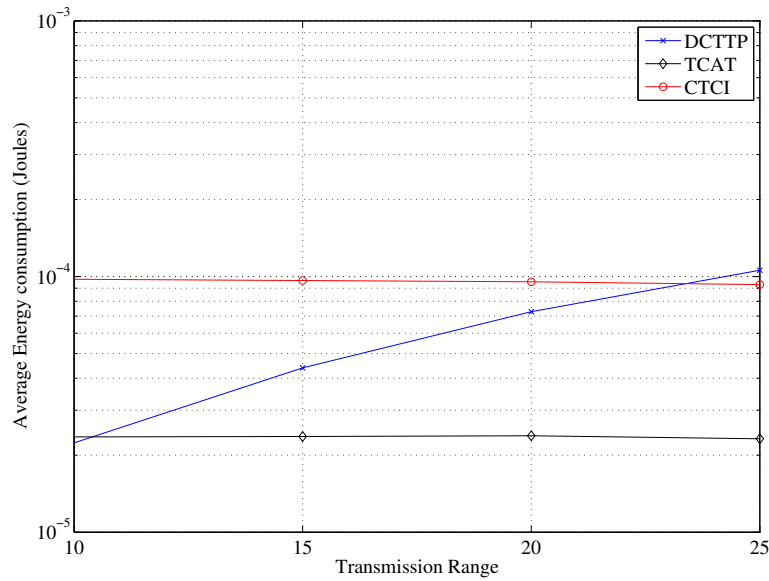


Figure 19. Impact of the transmission range on the average energy consumption.

#### 5.4. Number of Sample Points

Figure 20 depicts that the error distance of TCAT and CTCI decreases as the number of sample points increases. Observe that as the number of sample points is larger than 15, the error distance becomes stable. Figures 21 and 22 depict that the data size and energy consumption of CTCI are directly influenced by the number of the sample points. Therefore, there is a trade-off between the error distance and the energy consumption on the number of the sample points in CTCI.

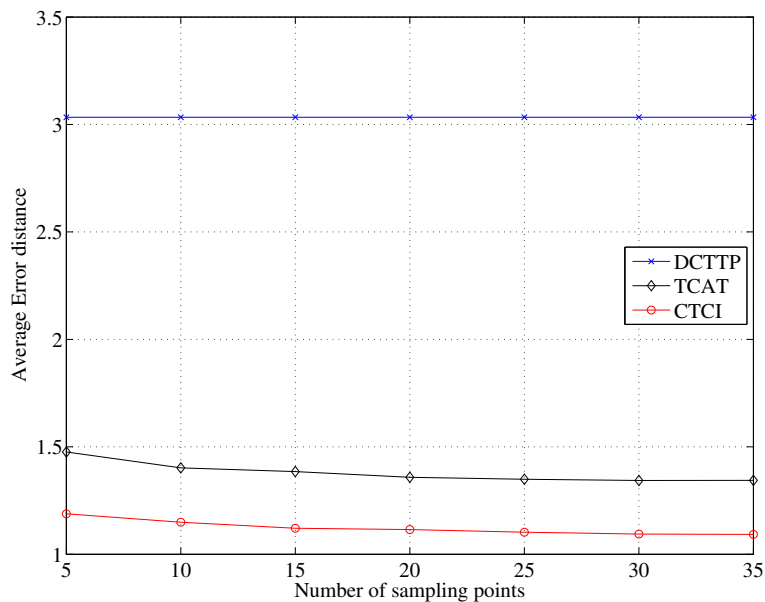


Figure 20. Impact of the sampling points on the average error distance.

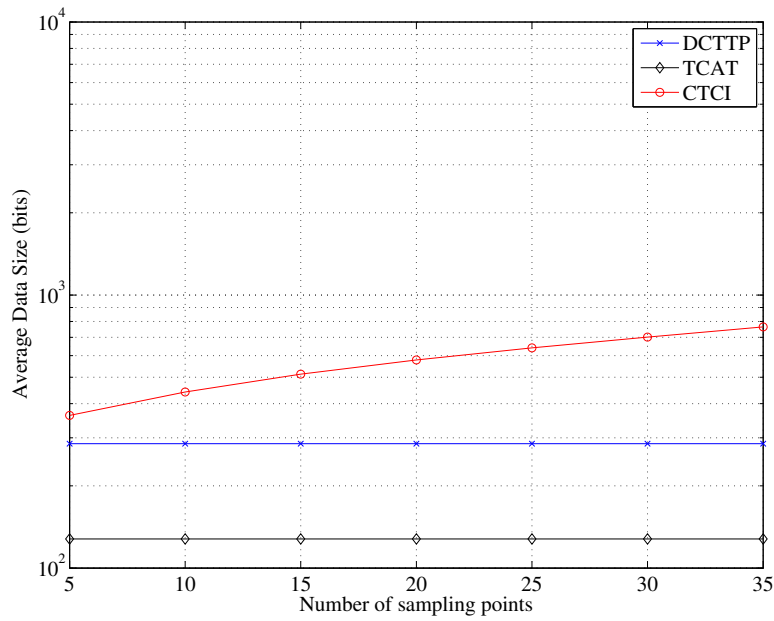


Figure 21. Impact of the sampling points on the average data size.

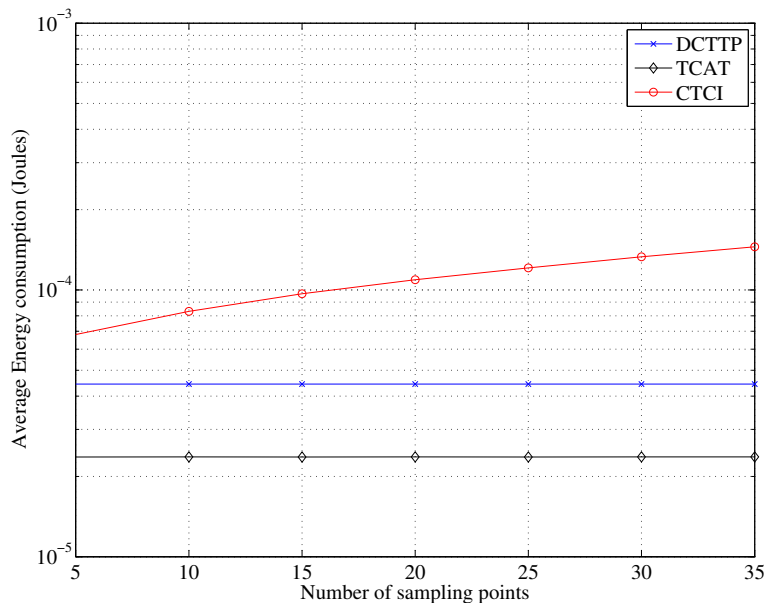


Figure 22. Impact of the sampling points on the average energy consumption.

5.5. Noise

Because of the limited capability of a sensor and a extremely small sample size for particle filtering, the fundamental estimation problems are to estimate the distance between the reference sensors and the target and to determine the property of the signal (e.g., the signal arrival angle). In a nutshell, accurate tracking is highly related to precise TOA/AOA measurements and the signal processing capability. In this set of experiments, we explore the impact of measurement noise on system performance. With the variance of angle estimation  $\sigma_\theta = 1$  and with varying the uncertainty of distance estimation  $\sigma_d$ , as shown in Figure 23, the increment of the measurement noise leads to a larger

error distance of TCAT and CTCI. Observe that with an increasing measurement uncertainty  $\sigma_d$ , the performance of the proposed CTCI method tends to be close to the CRLB.

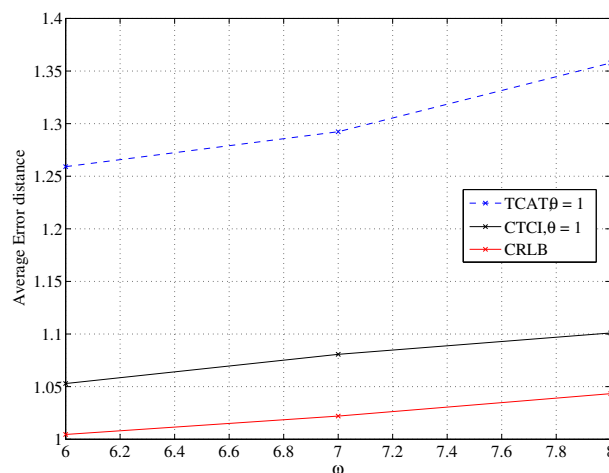


Figure 23. Impact of noise on the average error distance.

### 5.6. Compressing Bound

Figure 24 depicts that the error distance of DCTTP increases as the compressing bound increases. Because the sensed data of each active sensor node is compressed through the bitmap process, the distance between the active sensor node and the target may be represented by the average distance. Therefore, not only distortion may occur, but also the compressed data may influence the result of target tracking. The compressed data influences the weight of each active sensor node, so the reference value of the weight may decrease. Notice that in TCAT and CTCI, since the results of the positioning are not compressed through the bitmap process, the error distance is not influenced by the compressing bound. The bitmap process of the CTCI scheme is to obtain the information of measurement uncertainties in the original measurement data.

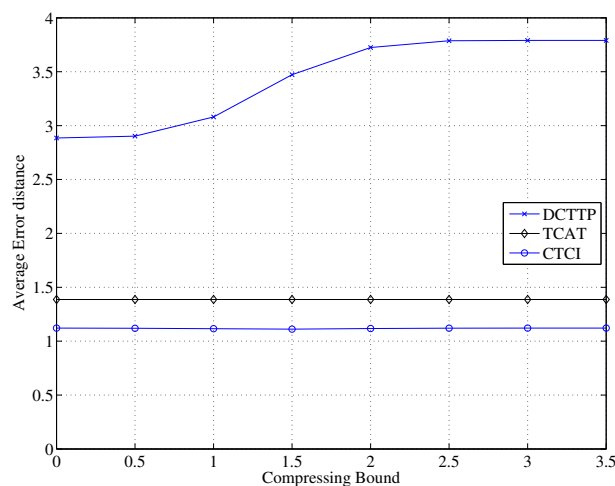


Figure 24. Impact of the compressing bound on the average error distance.

Observe that in Figure 25 when the compressing bound is zero, the data of DCTTP and CTCI will be reserved completely. As expected, the data size of DCTTP and CTCI decreases as the compressing bound increases. When the compressing bound is larger than two, the data are almost fully represented

by the average value, and the error distance of DCTTP is saturated to a large estimation error. Define the distortion ratio as the portion of original sample points, which are replaced by the average value (*i.e.*, using the compressing bound to control the number of the remaining sample points). As shown in Figure 26, the distortion ratio of CTCI increases as the compressing bound increase. Figure 27 shows that if sensors with DCTTP and CTCI transmit their original data that are not compressed through the bitmap process, the cost of the energy consumption will become considerable. As depicted in Figures 25 and 27, with a large compression bound, the TCAT scheme can be considered as a special case of CTCI, which compresses the samples to the mean.

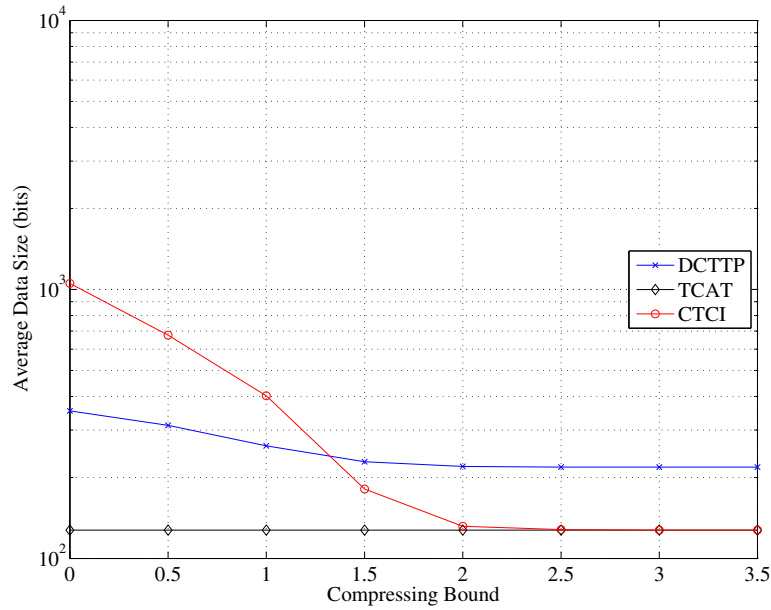


Figure 25. Impact of the compressing bound on the average data size.

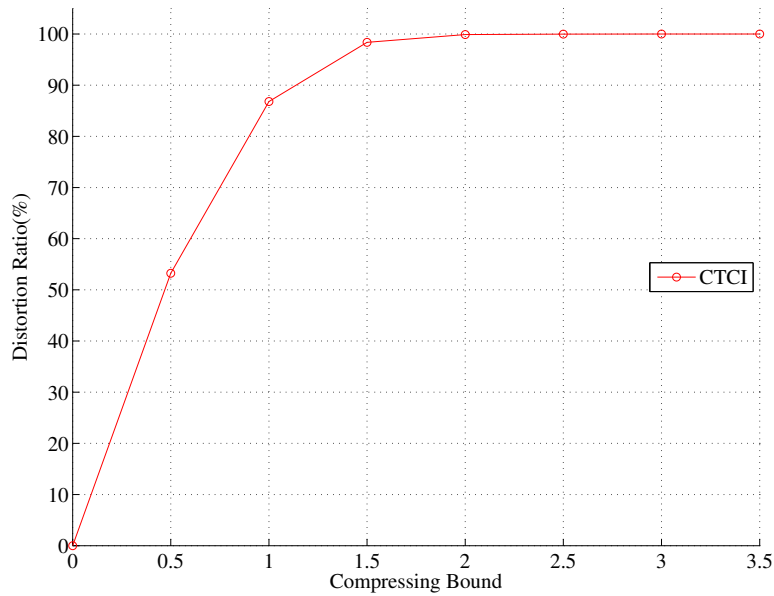


Figure 26. Impact of the compressing bound on the distortion ratio.

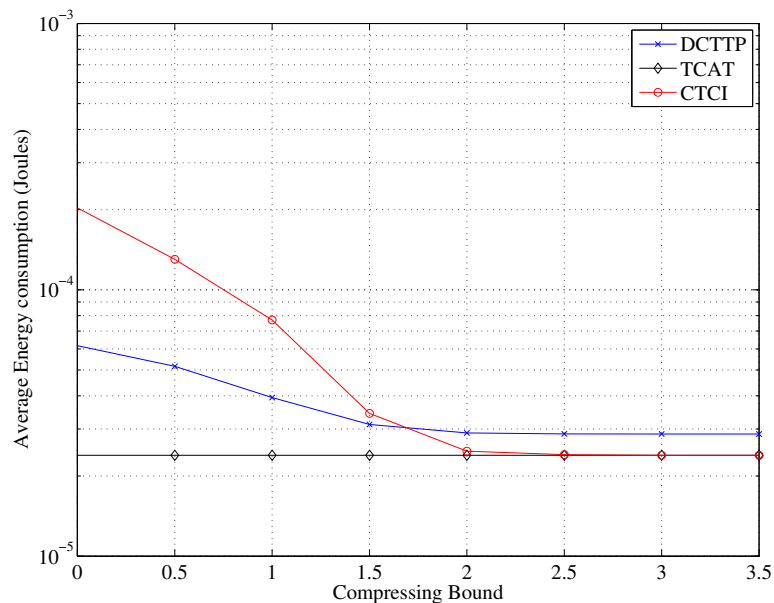


Figure 27. Impact of the compressing bound on the average energy consumption.

## 6. Conclusions

This paper presents an information compression scheme for target tracking. Comparing the proposed CTCI method to the TCAT and DCTTP methods, the simulation results show that the proposed CTCI scheme can adaptively control the estimation performance and closely reflect the relationship between the distortion of tracking results and energy consumption. Considering the system parameters (e.g., transmission range, the number of sensor nodes, compressing bound, the number of sub-cluster size and the number of sampling points), at the slight cost in data size and energy consumption, the tracking accuracy of CTCI is superior to those of TCAT (improved by 20%) and DCTTP (improved by 100%). Therefore, many trade-offs between the system parameters need to be considered in each scheme. In future work, we will explore a more effective and robust compression scheme, evaluate the compression complexity and quality and adjust the tactics of the two-level clustering to further reduce the data size, maintain the desired estimation quality and achieve energy conservation.

**Acknowledgments:** This research is supported by the Ministry of Science and Technology of Taiwan under Grant No. MOST-104-2221-E-005-034.

**Author Contributions:** Shi-Kuan Liao and Chih-Yu Wen conceived of and designed the experiments. Shi-Kuan Liao performed the experiments. Hsiao-Ping Tsai and Chih-Yu Wen analyzed the data. Kai-Jay Lai and Hsiao-Ping Tsai contributed reagents/materials/analysis tools. Shi-Kuan Liao and Chih-Yu Wen wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bhatti, S.; Xu, J. Survey of Target Tracking Protocols using Wireless Sensor Network. In Proceedings of the 2009 Fifth International Conference on Wireless and Mobile Communications, Cannes, France, 23–29 August 2009.
2. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188.
3. Kuo, Y.-C.; Hsiao, M.-Y.; Wen, C.-Y. An Integrated Mobile Sensor Platform for Collaborative Indoor Self-Positioning Applications. In Proceedings of the 2013 IEEE TENCON Spring Conference, Sydney, Australia, 17–19 April 2013; pp. 496–500.

4. Hightower, J.; Borriello, G. Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In Proceedings of the Sixth International Conference on Ubiquitous Computing, Nottingham, UK, 7–10 September 2004; pp. 88–106.
5. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In Proceedings of the IEEE Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000.
6. Chen, W.-P. A Data-quality Driven Framework for Data Dissemination in Wireless Sensor Networks. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 2004.
7. Al Aghbari, Z.; Kamel, I.; Elbaroni, W. Energy-efficient distributed wireless sensor network scheme for cluster detection. *Int. J. Parallel Emergent Distrib. Syst.* **2013**, *28*, 1–28.
8. Kim, W.; Mechtov, K.; Choi, J.Y.; Ham, S. On target tracking with binary proximity sensors. In Proceedings of the Information Processing in Sensor Networks (IPSN), Los Angeles, CA, USA, 15 April 2005.
9. Shrivastava, N.; Madhow, R.M.U.; Suri, S. Target tracking with binary proximity sensors: Fundamental limits, minimal descriptions, and algorithms. In Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems, New York, NY, USA, 1–3 November 2006.
10. Singh, J.; Madhow, U.; Kumar, R.; Suri, S.; Cagley, R. Tracking multiple targets using binary proximity sensors. In Proceedings of the 6th ACM International Conference on Information Processing in Sensor Networks, Cambridge, MA, USA, 25–27 April 2007.
11. Wang, Z.; Bulut, E.; Szymanski, B.K. Distributed energy-efficient target tracking with binary sensor networks. *ACM Trans. Sens. Netw.* **2010**, *6*, 32.
12. Lin, J.; Ren, Q. Compressing Information of Target Tracking in Wireless Sensor Networks. *Wirel. Sens. Netw.* **2011**, *3*, 73–81.
13. Chen, Y.C.; Wen, C.Y. Decentralized Cooperative TOA/AOA Target Tracking for Hierarchical Wireless Sensor Networks. *Sensors* **2012**, *12*, 15308–15337.
14. Gordon, N.J.; Salmond, D.J.; Smith, A.F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Proc. Radar Signal Process.* **1993**, *140*, 107–113.
15. Xu, Z.D.E.; Dasgupta, S. Target tracking and mobile sensor navigation in wireless sensor networks. *IEEE Trans. Mob. Comput.* **2013**, *12*, 177–186.
16. Okaie, Y.; Nakano, T.; Hara, T.; Nishio, S. Distributing nanomachines for minimizing mean residence time of molecular signals in bionanosensor networks. *IEEE Sens. J.* **2014**, *14*, 218–227.
17. Tseng, P.-H.; Ding, Z.; Feng, K.T. Cooperative Self-Navigation in a Mixed LOS and NLOS Environment. *IEEE Trans. Mob. Comput.* **2014**, *13*, 350–363.
18. Wen, C.Y.; Sethares, W.A. Automatic decentralized clustering for wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2005**, *5*, 686–697.
19. So, H.C.; Shiu, E.M.K. Performance of TOA-AOA hybrid mobile location. *IEICE Trans. Fundam.* **2003**, *E86-A*, 2136–2138.
20. Bravos, G.; Kanatas, A.G. Energy consumption and trade-offs on wireless sensor networks. In Proceedings of the 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'05), Berlin, Germany, 11–14 September 2005; Volume 2, p. 1279.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).