Rethinking of the Uncertainty: A Fault-Tolerant Target-Tracking Strategy Based on Unreliable Sensing in Wireless Sensor Networks

Yi Xie, Guoming Tang, Daifei Wang, Weidong Xiao, Daquan Tang and Jiuyang Tang
Science and Technology Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
[e-mail: xieyi@nudt.edu.cn]

Received March 9, 2012; revised May 7, 2012; accepted June 14, 2012;
published June 25, 2012

Abstract

Uncertainty is ubiquitous in target tracking wireless sensor networks due to environmental noise, randomness of target mobility and other factors. Sensing results are always unreliable. This paper considers unreliability as it occurs in wireless sensor networks and its impact on target-tracking accuracy. Firstly, we map intersection pairwise sensors' uncertain boundaries, which divides the monitor area into faces. Each face has a unique signature vector. For each target localization, a sampling vector is built after multiple grouping samplings determine whether the RSS (Received Signal Strength) for a pairwise nodes' is ordinal or flipped. A Fault-Tolerant Target-Tracking (FTTT) strategy is proposed, which transforms the tracking problem into a vector matching process that increases the tracking flexibility and accuracy while reducing the influence of in-the-filed factors. In addition, a heuristic matching algorithm is introduced to reduce the computational complexity. The fault tolerance of FTTT is also discussed. An extension of FTTT is then proposed by quantifying the pairwise uncertainty to further enhance robustness. Results show FTTT is more flexible, more robust and more accurate than parallel approaches.

Keywords: target-tracking; uncertainty; unreliable sensing; wireless sensor networks

A preliminary version of this paper is accepted by HPDIC Workshop in IEEE IPDPS 2012, May 21-25, Shanghai, China. This version includes a concrete description of the strategy design and gives quantitative extension of conducting sampling vectors. Besides, the outdoor system evaluation is appended to further validate our tracking strategy. The research is supported by the Universities Specialized Research Foundation for the Doctoral Program under Grant No.20114307110008, the National Natural Science Foundation of China under Grant No.60903225 & No.41001260, and the Graduate Student Innovation Foundation of the NUDT under Grant No. S100505.

http://dx.doi.org/10.3837/tiis.2012.06.002

1. Introduction

Mobile targets' tracking is a typical application of wireless sensor networks (WSNs), which plays a very important role in many military and civilian areas. Wireless sensors are always deployed randomly or deliberately in some area to detect intrusive target, obtain its real-time location and track its moving trace.

Although many excellent ideas have been proposed for target-tracking [1][2][3][4][5][6], most of existing works are based on the assumptions that the locations of sensor nodes are precise and the sensed data collected by sensor nodes is accurate. Actually, uncertainty, which worsens the tracking efficiency, is ubiquitous in sensor networks due to factors such as the impreciseness of positioning systems, environment noise and sensing irregularity. Hence, two main problems affected by the uncertainty are 1. for one-shot localization of the target, the error was serious, and 2. for continuous localizations (tracking mobile target), the returning results change back and forth instead of being smooth.

Considering most of existing tracking methods suffer from the ubiquitous uncertainty due to environmental noise and randomness of target mobility, several tracking strategies are designed to be tolerant of the uncertainty by calibrating the sensing result [7] or filtrating the low ranging-quality sensors [8] to improve the tracking efficiency. However, these tracking methods under uncertainty either lack flexibility in implementation, or lose sight of the fact that uncertain information can also promote the tracking efficiency. Consequently, this paper aims to propose a tracking strategy which can not only be tolerant of the uncertainty but also obtain and utilize it as additional helpful information of target tracking.

[9] and [10] have proclaimed unreliable sensing problem as well as the existence of pairwise sensors' uncertain area. Inspired by these previous works, the hypothesis and starting point of this paper are given. It is found out that the real distance between targets and sensor nodes has a bias with the generated distance from path-loss model. The real distance is independent of the position of the nodes/target as well as the received power. Hence for pairwise nodes, a uncertain area exists, in which the target signal's measurements of two nodes cannot be compared.

In this paper, a Fault-Tolerant Target-Tracking (FTTT) strategy is introduced which focuses on tracking mobile targets by utilizing the uncertain areas result from unreliable sensing. When the target is located in the uncertain areas of node pairs, the sampling reliability of each sensor and the comparison of their sensing results will be absolutely affected by the in-the-field factors, i.e., the comparison of Received Signal Strength (RSS) of pairwise nodes may be flipped, as shown in **Fig. 1**. What's more, a group of sampling is formed by multiple samplings during a localization, by analyzing which, a sampling vector can be generated to reflected the target's spatial information reliably. Besides, the monitored area can also be divided into several faces marked with a specific signature vector by the uncertain areas' boundaries,. Each face is considered to be a optional location of the target. Hence, by matching the two vectors to track the target, the uncertainty turns to be helpful information of target tracking instead of causation of great error. Additionally, the storage and time complexity is controlled effectively by designing a heuristic matching algorithm based neighbor face links. The fault tolerance is discussed to prevent impact on conducting of sampling vectors due to the loss of one-shot sensing. Elaborate performance evaluation and extensive simulation experiments show that our flexible FTTT can promote the tracking accuracy significantly with limited system cost.

We summarize the contributions made in this study as follows:

- To the best of our knowledge, this appears to be the first study that improves the efficiency of target tracking by using the unreliability or uncertainty of sensors' relative sensing relationship.
- A Fault-Tolerant Target-Tracking (FTTT) strategy is brought up. By conducting the sampling vector and the signature vector with uncertainty, the tracking process is turned into a vector matching problem, which makes FTTT flexible and robust.
- In order to consummate our tracking strategy, a series of methods, such as maximum likelihood matching model, heuristic matching algorithm and strategy fault tolerance, are proposed. These methods are used to increase the tracking accuracy, reduce the complexity and prompt the robustness. Thereafter, FTTT is extended by quantifying the pairwise uncertainty to conduct sampling vectors.
- Theoretical analysis is conducted to determine the grouping sampling times, while the worst-case tracking error is analyzed as well.
- Extensive simulations and an outdoor experimental system are given to validate the correctness and efficiency of FTTT strategy.

The rest of the paper is organized as follows: Section 2 reviews related works firstly. Then pairwise uncertainty problem is described and the uncertain boundaries are analyzed in Section 3. Section 4 introduced the Fault-Tolerant Target-Tracking strategy (FTTT) in detail and Section 5 presents its performance by analyzing the proper sampling times and the overall tracking error. Section 6 extends FTTT by quantifying the pairwise uncertainty. In Section7 the results of simulations and outdoor experiments are discussed. Finally, Section 8 concludes the paper.

2. Related Work

As range-based tracking methods [11][12][13] have to measure the distance between the target and the sensors or the angle of received signal, additional hardware is always needed [12], or requiring careful environment profiling [11][13]. Therefore, this kind of methods seem hard to be implemented. Another kind of prevalent tracking strategy is model-based, which is achieved by successively estimating the localization [2], velocity [4][5] and trace[14] of the target with target movement modeling, estimation[15] and filtering [16][17] (e.g., Kalman filter [18], Particle filters [19], Bayesian networks [20], Variational filter [21]). Most of the model-based methods use time-correlated measurements for localization, i.e., sensors may use the measurements of previous target positions to infer current target location. However, these methods are complex and inflexible which requires detailed assumptions of target mobility.

Recently, more flexible tracking schema was proposed by region division and sequence matching [22][23][24], in which divided faces/patches are regarded as the candidate locations of target tracking and reflect the relative distance relationship of sensors. These range-free and model-free tracking methods are based on pairwise relative distance, and turn tracking process into sequences matching problem without target mobility and accurate range-based localization, which seem to be generic, flexible, and compatible. However, these methods have their own problems as well. MSP and Sequence-based methods [23][24] address only stationary sensor node/target localization. Path-based MLE [22] requires assuming the maximum velocity of the target which is unfavorable in real system. Besides, these methods are not robust enough. Because they do not take consideration of the probable error from one-shot unreliable sensing, thus the detection sequence which reflects the relative distance from the target to the sensors is unconfident.

In order to deal with the uncertainty problem that most of existing tracking methods suffer from, CDL [7] use neighborhood hop-count sequence matching to implement local filtration to guarantee the high localizing ranging-quality. P*k*NN [8] propose a series of general models and provide an efficient tracking algorithm based on probabilistic *k*-nearest neighbors retrieval under uncertainty. Actually, these methods do not make full use of the unconfident and unreliable detection result, which can also provide useful information for improving tracking accuracy.

Since our tracking strategy is based on relative distance from the target to sensors and takes full consideration of the pairwise uncertainty, the strategy is proved to be robust and accurate in the unreliable sensing environment without increasing the overhead of system ultimately. Besides, compared to the constraint (such as the mobility or the maximum velocity of the target) of other methods, extra imposed conditions are not needed in our work.

3. The Uncertainty Problem Description

In this section, focusing on a given node pair, we first illustrate the existence of the uncertain area, where it is hard to determine which sensor node is nearer to the target by the RSS. Then the boundary of the uncertain area is analyzed.

3.1 Assumption of the Uncertain Area

In the real environment, the sensing results are ultimately unreliable, i.e., high tracking accuracy can only be achieved by best effort since the target's signal cannot be 100 percent accurate due to network delay, discrete sampling instances, sensing resolution and environment noise. Experiments are given to compare the RSS between the two nodes, $(node_1, node_2)$, and generate a detection sequence of the two RSS by descending sorting. As shown in **Fig. 1**, when the target is located near the perpendicular bisector of two sensor nodes, and is actually closer to one of them, the returning node sequences may flip at different time instants.

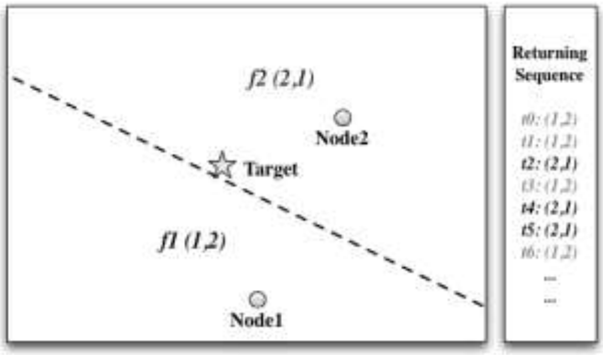


Fig. 1. Returning Node Sequence may Flip at Different Time Instant

Definition 1 (Uncertain Area): For any pair of nodes $n_1 : (x_1, y_1)$ and $n_2 : (x_2, y_2)$, there exists some area where it is hard to distinguish which sensor is closer to the target. The area is defined as *uncertain area*.

As a result, for each sampling time instant, the relative ordinal relationship of sensing intensity may not be obtained correctly. Then serious tracking error occurs in the methods that strictly rely on the certain detection node sequences [22][23][24] due to biased results.

3.2 Analysis of the Boundary of Uncertain Areas

A logarithmic attenuation noise model for signal-strength based detection was proposed in [25][26]. Hence, the received signal can be obtained by:

$$PL(d_k^i) = PL(d_0) + A - 10\beta\log(\frac{d_k^i}{d_0}) + X_k^i \text{, where } d_0 = 1 \text{ and } X_k^i \sim N(0,\sigma_X^2) \tag{1}$$

$PL(d_k^i)$ stands for the sensing results of node i at time k, where β is the path-loss exponent, d_k^i is the physical distance between node i and the target at time k and $PL(d_0)$ is an experimentally measured path loss at reference distance d_0. Path-loss exponent $\beta = 2$ corresponds to free space propagation, while $\beta = 3,4$ model environments with reflections and refractions. X_k^i is a random noise at time k for node i following a normal distribution with 0 mean and σ_X^2 variance.

Therefore, supposing there are two sensing nodes donated by m and n and the sensing resolution is ε, the limit of the RSS difference is equal to ε. It also stands for the maximum undistinguishable difference of the RSS of two sensor nodes. Then we get

$$\varepsilon = PL(d_k^n) - PL(d_k^m) = 10\beta\log(\frac{d_k^m}{d_k^n}) + (X_k^n - X_k^m) \tag{2}$$

Then, the ratio of the two distances is

$$\frac{d_k^m}{d_k^n} = 10^{\left(\frac{\varepsilon-(X_k^n - X_k^m)}{10\beta}\right)} = e^{\left(\frac{\ln 10\left[\varepsilon-(X_k^n - X_k^m)\right]}{10\beta}\right)}$$

For $X_k^n, X_k^m \sim N(0,\sigma_X^2)$, we can obtain that: $\varepsilon - (X_k^n - X_k^m) \sim N(\varepsilon, (\sqrt{2}\sigma_x)^2)$.
Taking the expectation of equation, we obtain

$$\frac{d_k^m}{d_k^n} = E(e^{\left(\frac{\ln 10\left[\varepsilon-(X_k^n - X_k^m)\right]}{10\beta}\right)}) = e^{\left(\frac{\ln 10}{10\beta}\varepsilon + \frac{1}{2}(\frac{\ln 10}{10\beta}\sqrt{2}\sigma_X)^2\right)} = C > 1 \tag{3}$$

We see that C is a constant for a given sensing resolution, and we derive that if two nodes cannot be compared with each other at time k, the upper bound of the distance between the target and the two nodes obeys formula(3). This means the trajectory equation of these points outlines the boundaries of the uncertain area.

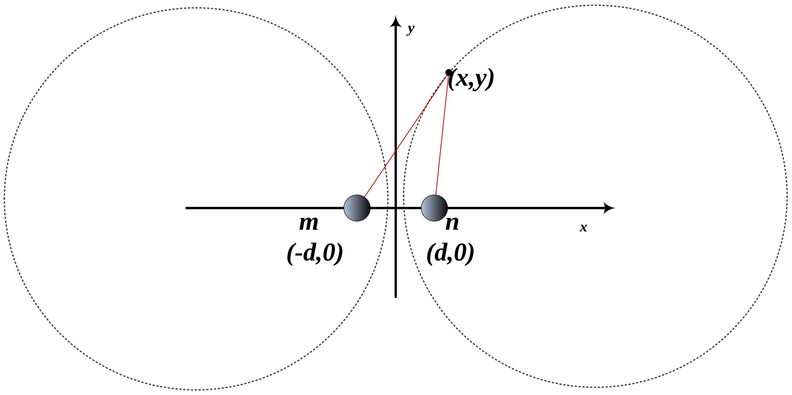


Fig. 2. Uncertain Boundaries of Node Pairs

As shown in Fig. 2., supposing two nodes is $2d$ away from each other with the coordinate

$(d,0)$ and $(-d,0)$, and considering points (x,y) on the uncertain boundaries obey the equation (3), we have

$$(x-\frac{C^2+1}{C^2-1}d)^2+y^2=\frac{4C^2d^2}{(C^2-1)^2} \tag{4}$$

This gives us a circle trajectory equation and the uncertain boundary can be defined accordingly.

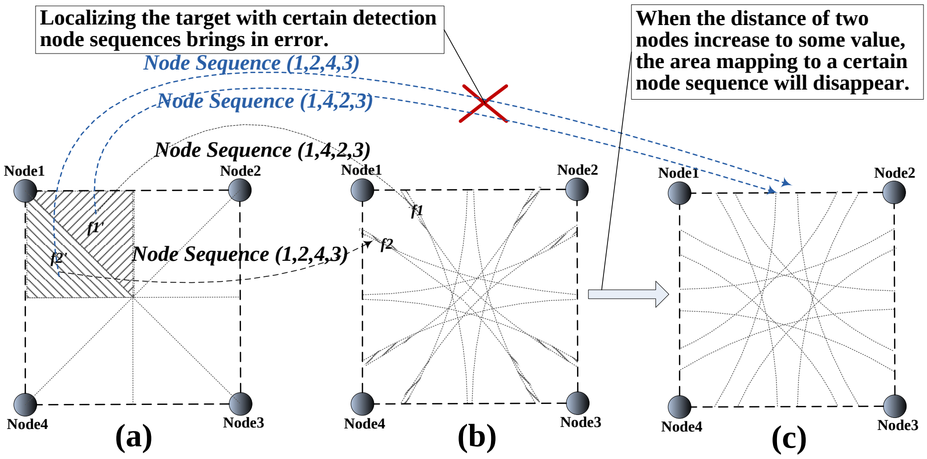

Localizing the target with certain detection node sequences brings in error.
When the distance of two nodes increase to some value, the area mapping to a certain node sequence will disappear.
Node Sequence (1,2,4,3)
Node Sequence (1,4,2,3)
Node Sequence (1,4,2,3)
Node Sequence (1,2,4,3)
Node1
Node2
Node3
Node4
f1'
f2'
f1
f2
(a)
(b)
(c)

Fig. 3. The Division of Area by Perpendicular Bisectors and Uncertain Boundaries of Node Pairs

Definition 2 (Uncertain Boundary): The trajectory of the two axisymmetric circles (Circles of Apollonius [27])is defined as the *uncertain boundary* of node pairs, whose axis is the perpendicular bisector of the two nodes, as shown in **Fig. 2**.

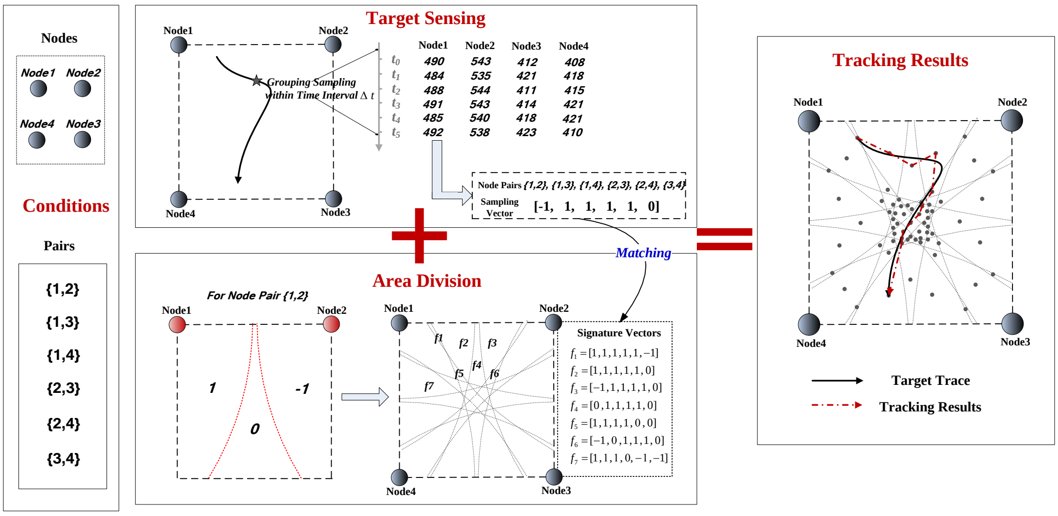
In [22] (tracking with certain node sequences), considering four sensors deployed as grid, the field should be divided into eight faces by the intersections of the perpendicular bisectors of each two nodes. This assumes all returned RSS had no error, as shown in **Fig. 3(a).** However, according to the analysis of sensing unreliability and the definition of uncertain area mentioned above, when d_k^m varies from 0 to ∞, uncertain boundaries are two axisymmetric circles, as shown in **Fig. 3(b)**. The area of the grid is divided into several faces by the intersections of pairwise uncertain boundaries of the four nodes. Thus, the eight faces with certain ordinal RSS of each node turn into tiny areas, as shown in **Fig. 3(a)** and **Fig. 3(b)**, e.g. $f_1 \to f_1', f_2 \to f_2'$. Hence, the certain detection sequences seem hard to be obtained, which means the target cannot be located into correct faces in majority of situations. Furthermore, when the distance between two nodes increases to some value, the faces in which we can get certain detection node sequences will no longer exist, as shown in **Fig. 3(c)**. Therefore, the certain sequence matching strategy will cause great tracking error of target one-shot localization or continuous tracking in the real environment, while the drawbacks can be overcome by the method based on uncertain area.

4. Fault-Tolerant Target-Tracking Strategy

In this section, the Fault-Tolerant Target Tracking (FTTT) strategy will be detailed discussed. Firstly, an overview of FTTT is given. Then, four subsections that support the FTTT, the construction of sampling vector, the division of monitor area, the heuristic matching algorithm and the strategy fault tolerance, will be introduced in turn.

4.1 Strategy Overview

The previous section shows that tracking by matching certain node sequences is full of holes due to the uncertain characteristic of the sensors' sampling. The One-shot non-confident sampling cannot reflect the real relative distant relationship because of the contingency. As a result, great tracking error exists. However, as shown in Fig. 4, a straight-forward method can be built by utilizing the unreliable information.



	Node1	Node2	Node3	Node4
t_0	490	543	412	408
t_1	484	535	421	418
t_2	488	544	411	415
t_3	491	543	414	421
t_4	485	540	418	421
t_5	492	538	423	410

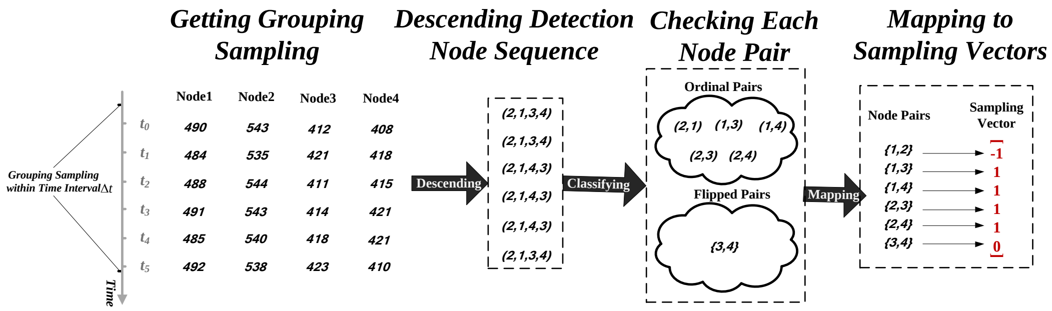
Fig. 4. Overview of Fault-Tolerant Target-Tracking Strategy

On the one hand, for each localization, each sensor samples multiple times in a very short time interval synchronously which forms a grouping sampling. For each pair of nodes, if the RSS order is flipped in different time instant, we assign value 0 to this pair. If the pairwise RSS order is ordinal all the time, 1 and -1 is assigned to it according to the order is ascending or descending. By enumerating all the pairs' values, a sampling vector can be obtained. On the other hand, for pairwise nodes, the uncertain boundaries divide the monitor area into three parts. We assign 0 to the uncertain area, 1 to the area nearer to the smaller node ID sensor, and -1 to the remaining one. Uncertain boundaries of node pairs intersect with each other and divide the area into several faces, each face has a unique signature vector by enumerating the assigned value according to the uncertain boundaries of each pair.

Therefore, the target can be localized by matching the sampling vector with the signature vector of faces, and along the moving trace of the target, periodic grouping sensing results from related sensor nodes produce a series of sampling vector, which embed relative position relationships among the sensor nodes and mobile target, so we can track the target effectively.

4.2 Construct Grouping Sampling Vectors

Since different geographic distances between each sensor node and the target will lead to various sensing results at each sensor node, by sampling several times in a very short time interval, we can consider the target is relative stationary and a grouping sampling can be conducted.



	Node1	Node2	Node3	Node4
t_0	490	543	412	408
t_1	484	535	421	418
t_2	488	544	411	415
t_3	491	543	414	421
t_4	485	540	418	421
t_5	492	538	423	410

Fig. 5. Example of Constructing Sampling Vector

Definition 3 (Grouping Sampling): Supposing n sensors have been deployed in the monitor area, and in a very short time interval Δt, each sensor samples almost synchronously at different time instant e.g. from t_0 to t_{m-1}. All the sampling results can be organized as a grouping sampling, which can be denoted by a matrix as:

$$\begin{bmatrix} rss_{1,t_0} & rss_{2,t_0} & rss_{3,t_0} & \cdots & rss_{n,t_0} \\ rss_{1,t_1} & rss_{2,t_1} & rss_{3,t_1} & \cdots & rss_{n,t_1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ rss_{1,t_{m-1}} & rss_{2,t_{m-1}} & rss_{3,t_{m-1}} & \cdots & rss_{n,t_{m-1}} \end{bmatrix}$$

At each time instant, the sampling RSS of each sensor, which is a row of the matrix, can form a detection sequence by given a descending ordering of elements. For any sensor node n_i, its sampling results at different time instant forms a set, rss_i, which is also a column of the matrix. For any pair of nodes (n_1 and n_2) and their corresponding RSS (rss_1 is sampled by n_1 and rss_2 by n_2), there are three situations: 1. if all elements in rss_1 is greater than those in rss_2, defined $rss_1 > rss_2$; 2. if opposite, define $rss_1 < rss_2$; 3. if not all the elements in rss_1 is greater(smaller) than those in rss_2, define $rss_1 <> rss_2$. So their RSS relationship has three probable states according to the group of sampling, which can be formatted by two ordinal situations (n_1, n_2), (n_2, n_1) and one flipped situation $\{n_1, n_2\}$.

Definition 4 (Node Pair Value): For any pair of nodes, n_i and n_j, $i < j$, after comparing their sampling results (RSS), we derive the *node pair value*, denoted by V_{n_i,n_j}. $\forall_{i,j} V_{n_i,n_j} \in \{-1,0,1\}$. If $rss_1 > rss_2$, the value of this pair is 1; if $rss_1 < rss_2$, the value of this pair is -1; if $rss_1 <> rss_2$, the value is 0.

For the given deployed sensor nodes, all the values of node pairs can form a vector which can represent the information of each grouping samplings.

Definition 5 (Sampling Vector): Supposing there are n sensors deployed in the monitor field, there are $N = C_n^2$ pairs totally. By enumerating all the pairs in ascending order will get a sequence, formally, $(n_1,n_2),(n_1,n_3),...,(n_1,n_n),(n_2,n_3),(n_2,n_4),...,(n_2,n_n),......,(n_{n-1},n_n)$, we obtain the node pair values successively and a sampling vector $\vec{V}_d$ which is given by $\vec{V}_d = [v_1, v_2, ..., v_N{}_d]$, $v_i \in \{-1,0,1\}, i = 1,2,...,N$.

Algorithm 1

***Algorithm* 1** *Constructing Sampling Vector*

```
input:   Sampling results of sensors at different time instant, Sampling times $k$
output:  Sampling Vector $\overrightarrow{V_d}$
1:  while $count \neq k$ do
2:      $TempArray = [rss_{1,t_c}, rss_{2,t_c}, \ldots, rss_{n,t_c}]$;
3:      $SamplingArray =$ Descending($TempArray$);
4:      $Matrix[count] \leftarrow SamplingArray$;
5:      $count = count + 1$;
6:  end while;
7:  $count = 0$;
8:  for $i = 0$ to $n$ do
9:      for $j = i$ to $n$ do
10:         for $w = 0$ to $k$ do
11:             if $Matrix[i][k] \geq Matrix[j][k]$ then
12:                 if $V_d[count] == -1$ then
13:                     $V_d[count] = 0$;
14:                     break;
15:                 else
16:                     $V_d[count] = 1$;
17:                 end if;
18:             else if $Matrix[i][k] \leq Matrix[j][k]$ then
19:                 if $V_d[count] == 1$ then
20:                     $V_d[count] = 0$;
21:                     break;
22:                 else
23:                     $V_d[count] = -1$;
24:                 end if;
25:             end if;
26:         end for;
27:         $count = count + 1$;
28:     end for;
29: end for;
```

We summarize four steps to obtain the sampling vectors of each grouping sampling:

1. Aggregate all the sensors' sampling results at different time instant within the sampling unit (time interval Δt). Then the grouping sampling matrix is generated;

2. Descend each row of the grouping sampling matrix. Then for each sampling time instant, detection node sequences which reflects the RSS ordering can be found out;

3. Compare and classify each node pair through all the detection node sequences and analyze which pairs have been flipped and which are ordinal all the time;

4. Get the value of each node pair according to 3), and enumerate them all in ascending order to form the sampling vectors.

The algorithm constructing a sampling vector of n sensors from a grouping sampling is shown as **Algorithm 1**.

For example, as shown in **Fig. 5**, four sensors sampled almost synchronously at different time instant e.g. from t_0 to t_1 in a very short time interval Δt. The six sampling results at different time instant of four sensor nodes belong to a grouping sampling and form the grouping sampling matrix. After giving the descending order of the sampling results of each time instant (each row of the matrix), it is easy to find that only one pair $(3,4)$ flipped in this group results, and other pairs are ordinal as $(2,1),(1,3),(1,4),(2,3),(2,4)$ out of the $C_4^2 = 6$ pairs. Therefore, the sampling vector $[-1,1,1,1,1,0]$ is constructed.

The returned vector $\overrightarrow{V_d}$ is the sampling vector of this grouping sampling. Line 1 to 6 of **Algorithm 1** is performed to get grouping samplings matrix and descend each row, and line 8 to 29 is to build the sampling vector. It is easy to get the storage complexity of this algorithm, which is $O(n \cdot k + n(n-1)/2) = O(n^2)$ .While the time complexity of this algorithm should be $O(n^2 \cdot k)$. Since k is a quite limited constant (which will be detailed discussed in Section 5), the complexity is ought to be $O(n^2)$.

4.3 Construct the Signature Vector of Divided Faces

In order to localize and track the target, the monitor area is divided into several faces by the uncertain boundary of node pairs and each face is ready to be a candidate location of the target, as shown in **Fig. 7(a)**.

1) Area Dividing and Signature Vector Generating: For a given node pair, any point in the monitor area has three states: nearer to one node of the pair, nearer to the other, or in the uncertain area of this pair. This means that each location point in the area has unique node pair values of all the node pairs. Hence after combining all the node pair values of some location point, its signature vector can be get, which is used to describe the spatial information of the location point.

Definition 6 (Signature Vector): For n sensors deployed in the monitor field, there are $N = C_n^2$ pairs totally. By enumerating all the pairs in ascending order, we will get a sequence, formally, $(n_1, n_2), (n_1, n_3), ..., (n_1, n_n), (n_2, n_3), (n_2, n_4), ..., (n_2, n_n), ..., (n_{n-1}, n_n)$. Focusing on any point p in the monitor area, for each given pair $(n_i, n_j), i < j$, we assign a value to denote the three situations: p is nearer to n_i, the value is 1; p is nearer to n_j, the value is -1; p is in the uncertain area of this pair, the value is 0. Then we obtain the value of each node pair successively and a signature vector $\vec{V_s}$ of the point p is given by $\vec{V_s} = [v_1, v_2, ..., v_N\mathrm{d}]$, $v_i \in \{-1, 0, 1\}, i = 1, 2, ..., N$.

Lemma 1 (Uniqueness): All the points in the same face have the same signature vector and all the points that have the same signature vector belong to the same face. Formally, $\forall_{p_1, p_2} \overrightarrow{V_s}(p_1) = \overrightarrow{V_s}(p_2) \Leftrightarrow p_1, p_2 \in f$, where p_1, p_2 is two elements of the node pair set and f is a specific face.

This lemma can be proofed by contradiction easily, which indicated us that each face of the area should correspond to an ordinal pair set as well as a flipped pair set. Then each face can also be mapped to a specific signature vector.

According to our analysis, every group of sampling will return a sampling vector which represents the spatial information of the target, while every face divided by the uncertain boundaries has a unique signature vector. Then the location of the target can be determined by matching the two vectors, as shown in **Fig. 7(b)**.

2) Approximate Grid Division: The dividing of the area and computing of the signature vector of each face should be completed in the preprocess phase before starting the tracking the target, and information is real-time aggregated and stored in the base stations or in the cluster heads. The network model, topology, energy management and data synchronization of the target tracking sensor networks can be easily achieved according to [28].

Although real-time capability and precision of localization are very important to target tracking, the division of area involves very complex geometry problem, as shown in **Fig. 6(a)**. Therefore, an approximate division and centroid obtaining method are needed, which can divide the area easily and quickly, besides, the location error can be controlled dynamically and kept under a certain degree.

The area can be divided into a number of square grids, and the coordinates as well as the signature vector of each grid can be easily got. Make the center of the grid on the bottom left the origin, horizontal direction the X axis and vertical direction the Y axis, and define the center coordinate of each grid as its coordinate, as shown in **Fig. 6(b)**. Then scan all the grids in the tracking area and classify the grids which have the same signature vector. Supposing m grids in the approximate area has the same signature vector with face f, and their coordinates are respectively $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$, as shown in **Fig. 6(c)**. Then, as shown in **Fig. 6(d)** the approximate location or the centroid of the approximate of face f can be expressed as:

$$(x_G, y_G) = (\frac{1}{m}\sum_{i=1}^{m} x_i, \frac{1}{m}\sum_{i=1}^{m} y_i) \quad (5)$$

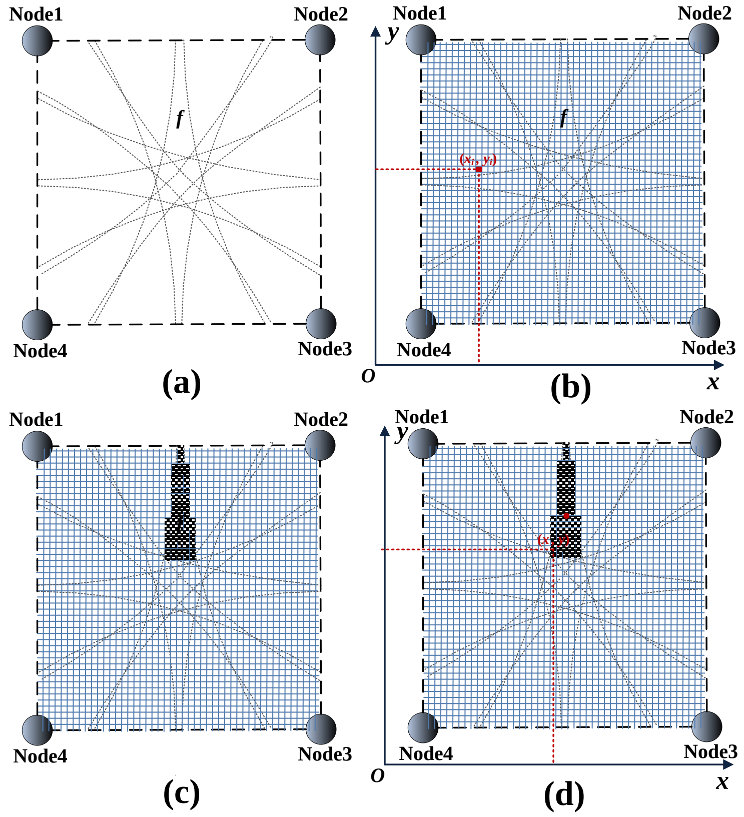


Fig.6. Example of Approximate Grid Division

An adaptive grid division algorithm in our previous work [29] can be used to simplify the face division pre-process of FTTT. However, the detailed discussion is beyond the capability of this paper.

4.4 Strategy Design in Detail

In ideal case, a sampling vector $\overrightarrow{V_d}$ should be identical with one and only one of the face's signature vector $\overrightarrow{V_s}$. However, in a real system, sensing at each sensor node could be irregular and affected by many factors. For n deployed sensor nodes there are only $O(n^4)$ divided faces (candidate locations). It is obvious that the number of different sampling vectors is much greater than the number of divided faces. Hence, sometimes there may not exist a face whose signature vector is equal to the sampling vector. For the no face matching situation, a maximum likelihood matching method is proposed to guarantee the target can be determined into a specific face.

1) Maximum Likelihood Matching: To be simple, we adopt Euclidean Distance to evaluate the similarity of two vectors.

Definition 7 (Similarity of two different vectors): For two given different vectors, $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ ($\overrightarrow{V_1} \neq \overrightarrow{V_2}$), their similarity $S_{\overrightarrow{V_1},\overrightarrow{V_2}}$ is define as $S_{\overrightarrow{V_1},\overrightarrow{V_2}} = 1/\left\|\overrightarrow{V_1} - \overrightarrow{V_2}\right\|$.

We can easily prove the reasonableness of the supposing similarity by the definition of sampling vectors and signature vectors. If one signature vector of a face has the maximum similarity to the sampling vector, it also means that the expecting location of the target is most likely in this face. Hence, the corresponding face of a sampling vector can be determined by selecting the face which has the maximum similarity with the sampling vector.

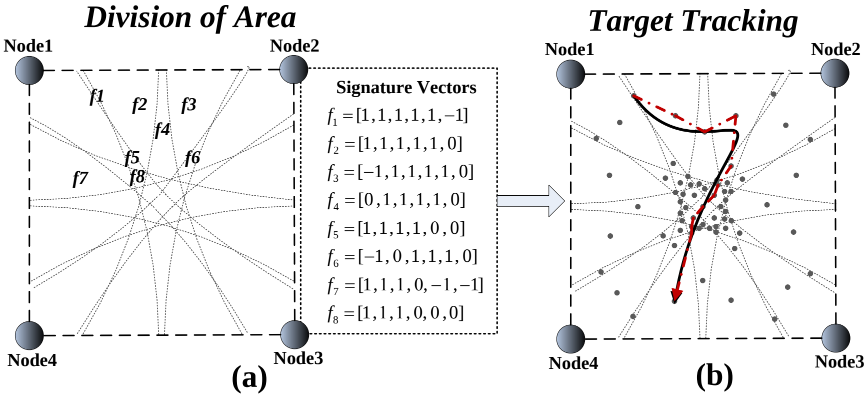


Fig. 7. Example of Divided Faces and Tracking Process

As shown in **Fig. 5**, the sampling vector $\overrightarrow{V_d} = [-1,1,1,1,1,0]$, and as shown in **Fig. 7 (a)**,the signature of f_3 is also $[-1,1,1,1,1,0]$. Hence, the target is localized in face f_3. However, if the sampling vector is appears to be $\overrightarrow{V_d} = [-1,1,1,1,1,1]$, there is no faces whose signature vector directly matches it. Therefore, the maximum likelihood matching method is requested to help localize the target. Computing the similarity between the sampling vector and the signature vector of each face, we can find out that the similarity between the sampling vector $\overrightarrow{V_d}$ and the signature vector of f_3 is $S_{\overrightarrow{V_d},\overrightarrow{V_s}(f_3)} = 1$, which is the maximum similarity. Hence, the target should be localized in face f_3 accordingly.

2) Heuristic Matching Algorithm

The target tracking problem is turned into a vector matching process. There are $O(n^4)$ divided faces totally if n sensor nodes can sense the target. Each divided face is stored as a record with face ID and its signature vector. Hence, the storage complexity is $O(n^4)$. Besides, matching a sampling vector with signature vectors will traversing all the candidate faces, which means the time complexity of the ergodic matching process is $O(n^4)$. However, we find that the divided faces are not isolated but have inherent correlations, thus neighbor face links can be built to describe the correlation. Firstly, we give the definition of neighbor face.

Definition 8 (Neighbor face links): For any given face f, if an adjacent face f' has one common edge with f, f' is a neighbor face of f. Edges which link neighbor faces are defined as *neighbor face links*. As shown in **Fig. 8(a)**, f_1, f_4 and f_5 are neighbor faces of f_2. The blue segments are the *neighbor face links*.

Theorem 1 (Relation between neighbor faces): Suppose f and f' are neighbor faces, the signature vector of f is $\overrightarrow{V_s}(f)$ and the signature vector of f' is $\overrightarrow{V_s}(f')$, $\overrightarrow{V_s}(f)$ and $\overrightarrow{V_s}(f')$ satisfies that $\left\|\overrightarrow{V_s}(f) - \overrightarrow{V_s}(f')\right\| = 1$.

Proof: For f and f' which are neighbor faces, they must have one common edge. For two points p and p' whose signature vector is denoted by $\overrightarrow{V_s}(p)$ and $\overrightarrow{V_s}(p')$, and they belong to f and f', respectively. Supposing p is very close to p' and only divided by the common edge, it is obvious that the edge is on some uncertain boundary of one node pair. Then, there is one and only one different component of the signature vectors of p and p', denoted by v_i and v_i', and $|v_i - v_i'| = 1$. As p is very close to p', there is no other uncertain boundaries crossing the two points. Hence the signature vectors of the two points satisfies $\left\|\vec{V_s}(f) - \vec{V_s}(f')\right\| = 1$. According to Lemma 1, this theorem is proved. ■

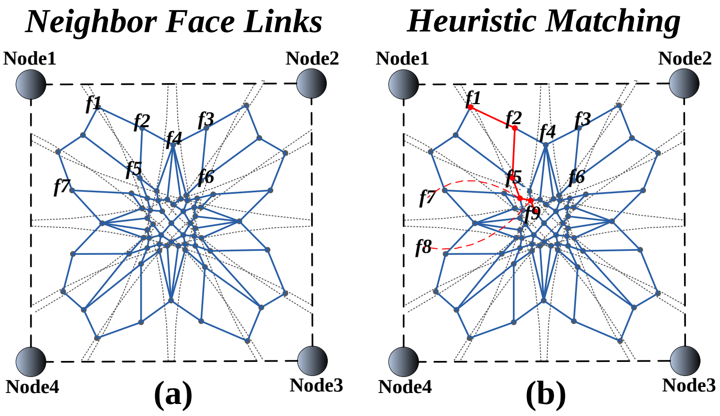


Fig.8. Example of Heuristic Matching with Neighbor Face Links

All the neighbor face links form a set, denoted as $L = \{(f, f') \mid \left\|\vec{V_s}(f) - \vec{V_s}(f')\right\| = 1\}$, which can describe the inherent correlations among divided faces very well. As shown in Fig. 8(b), with the information of the neighbor face links, the target tracking strategy has two advantages. **1**. The matching of any localization is separated into several tracking rounds, and within every round the neighbor face whose signature vector has the maximum similarity with sampling vector indicates the matching "direction" to assure the matching is more efficient. Then the original ergodic matching process has been changed into matching along with a convergent searching path. **2**. Consecutive tracking further reduces matching complexity. Considering the target tracking is a successive process, the face which is the previous target location is selected to be the next starting matching face. Referring to the previous localizing, each localization matching will end in very limited searching rounds.

Algorithm 2 illustrates the target tracking process. The number of edges link the neighbor relations is equal to the number of segments formed by inspection of uncertain boundary, which is $O(n^4)$ as well. We see that the storage complexity including the face ID, signature records and the neighbor face links, is still $O(n^4)$, which means building the neighbor face links will not ultimately increase the storage complexity. However, if we apply the correlation among these chaotic divided faces and utilize heuristic matching method based on neighbor relationship, the time complexity of the target tracking strategy will drop from $O(n^4)$ to $O(n^2)$. Taking the **Algorithm 1** into account, the final storage complexity is $O(n^4)$ and the time complexity is $O(n^2)$.

Algorithm 2

Algorithm 2 *Tracking with Vector Matching*

```
input:   FaceID-$V_s$ pairs $(f_1,V_s(f_1)),(f_2,V_s(f_2)),...,(f_k,V_s(f_k))$
         Neighbor face links set $L$
         Sampling Vectors $V_{d,1},V_{d,2},...,V_{d,m}$
output:  Target-tracking trace $P$

for $i=0$ to $m$ do
    if $f_0 == null$ then
        $f_0 =$ Initialization ();
    end if;
    $f_n = f_0$;
    while $S_{neighbor} > S_{current}$ do
        $f_c = f_n$;
        $S_{current} = S_{V_{d,i},V_s(f_c)}$;
        $S_{neighbor} = 0$;
        while $f_c.next\_neighbor \neq null$ do
            if $S_{neighbor} < S_{V_{d,i},V_s(f_c.next\_neighbor)}$ then
                $S_{neighbor} = S_{V_{d,i},V_s(f_c.next\_neighbor)}$;
                $f_n = f_c.next\_neighbor$;
            end if;
        end while;
    end while;
    $f_0 = P[i] = f_c$;
    $S_{current} = -1$;
end for;
```

3) Strategy Fault Tolerance: In a real system, one-shot sensing at each sensor node could be irregular and affected by environment noise or obstacles, but the dimension of signature vectors is inherent and fixed, which only relies on the density of the sensor networks and the sensing range of sensors. Hence, the dimension of the sampling vectors may be less than the inherent dimension of signature vectors, because of the loss of some one-shot sensing results. Then the fault tolerance problem should be discussed.

Actually, the dimension of the sampling vectors is relatively dynamic, which cannot guarantee obtaining all the nodes' sampling results. For a group sampling, all the sensors are in the node set N_s. However, for some reason such as breakdown of sensors or fault occurrence, some of the sensors do not return sensing results, which are in the node set $\overline{N_r}$, and sensors which return results are in the set N_r. Obviously, $N_r \cup \overline{N_r} = N_s$.

Thus, one node pair value can only have three situations: 1. If both nodes are in the set N_r, the node pair value can be got by comparing the two sensing results; 2. If both nodes are in the set $\overline{N_r}$, We define '$*$' to be the node pair value, because the sensors in $\overline{N_r}$ do not actually participate in tracking, so that we cannot determine the relative RSS relationship; 3. If only one of the sensors is in $\overline{N_r}$ and the other is in N_r, the node pair value is determined by the principle that sensing results in $\overline{N_r}$ are always smaller than those in N_r. Hence, each node pair value of sampling vector can be conducted accordingly.

$$\textit{node pair value} \text{ of } n_i \text{ and } n_j \ (i<j) = \begin{cases} \text{According to } \textit{definition } 4 & n_i \in N_r \text{ and } n_j \in N_r \\ 1 & n_i \in N_r \text{ and } n_j \in \overline{N_r} \\ -1 & n_i \in \overline{N_r} \text{ and } n_j \in N_r \\ * & n_i \in \overline{N_r} \text{ and } n_j \in \overline{N_r} \end{cases} \tag{6}$$

As a result, even if the fault occurs, the sampling vector $\overrightarrow{V_d}$ can also be filled with the equal length to the signature vector of divided faces. Considering the factor of fault occurrence and the extended node pair value, we have to redefine the difference of vectors.

Definition 8 (difference of two different vectors): For two given different N-dimensional vectors (N denote the number of node pairs), $\overrightarrow{V_1} = [v_{1,1}, v_{1,2}, ..., v_{1,N}]$ and $\overrightarrow{V_2} = [v_{2,1}, v_{2,2}, ..., v_{2,\mathrm{N}}]$, ($\overrightarrow{V_1} \neq \overrightarrow{V_2}$), their difference $\overrightarrow{V_1} - \overrightarrow{V_2}$ is define as:

$$\overrightarrow{V_1}-\overrightarrow{V_2}=[(v_{1,1}-v_{2,1}),(v_{1,2}-v_{2,2}),\dots,(v_{1,\mathrm{Nd}}-v_{2,\mathrm{Nd}})]$$

The difference of corresponding components of the two vectors obeys equation (7).

$$v_{1,i}-v_{2,i}=\begin{cases} v_{1,i}-v_{2,i} & v_{1,i}\neq * \text{ and } v_{2,i}\neq * \\ 0 & else \end{cases} \tag{7}$$

For instance, as shown in **Fig. 5**, four sensing results are expected to be returned. However, there are only results from n_1 and n_3 which satisfy $rss_{n_1} > rss_{n_3}$.

As the discussion above, the sampling values of node pair (n_1,n_2), (n_1,n_3), (n_1,n_4) and (n_3,n_4) is 1 . Analogously, the sampling value of (n_2,n_3) is -1 and the sampling value of (n_2,n_4) is $*$. Hence, the sampling vector is conducted to be $\overrightarrow{V_d}=[1,1,1,-1,*,1]$. After finding the face whose signature vector has the maximum similarity with this sampling vector, the target should be located in the face f_8 (whose signature vector $\overrightarrow{V_s}(f_8)=[1,1,1,0,0,0]$) with the similarity $S_{\overrightarrow{V_d},\overrightarrow{V_s}(f_8)}=1/2$.

5. Performance Discussion

We can see that FTTT is based on the grouping sampling which means each sensor should sense several times in a very short time interval. A proper sampling times k should be determined, which can guarantee high confidence of target's spatial information. Another key metics we must concern about is the tracking error, which directly reflects the efficiency of a tracking strategy. We should analyze the parameters related to the tracking error as well.

5.1 Determination of Grouping Sampling Times

For any pair of nodes, n_1 and n_2, e.g., denoted as pair P_1, the probable order is twofold: one is sequential order $O_s=(1,2)$, the other is reverse order $O_r=(2,1)$. When the target located in the uncertain area of n_1 and n_2, the pair is ought to be flipped. However, the number of grouping sampling times is limited, so the grouping sampling may only contain O_s or O_r, which means we cannot guarantee getting the expected flipped pair $\{1,2\}$. For k continuous grouping samplings, when the target appears in the uncertain area, we suppose the probability that one time sampling result appear to be sequential or reverse order are both $1/2$. Then the grouping sampling cannot obtain the information of expected flipped pair

$$f=f_{\overline{P_1}}=\left(\frac{1}{2}\right)^k+\left(\frac{1}{2}\right)^k=\left(\frac{1}{2}\right)^{k-1}.$$

Hence, the probability that grouping sampling getting the expected flipped pair is

$$f_1=1-f=1-\left(\frac{1}{2}\right)^{k-1}$$

For N excepted flipped pairs, $P_1,P_2,\dots,P_N$, it is obvious that $f_{\overline{P_1}}=f_{\overline{P_2}}=\dots=f_{\overline{P_N}}=f$, the probability that a grouping sampling can get all the flipped information is

$$f_N=\sum_{M=0}^{N}(-1)^M\cdot C_N^M\cdot f^M \tag{8}$$

After obtaining the general term of the recurrent sequence above (according to Appendix I), we can derive the probability that a grouping sampling can get all flipped information of N node pairs as $f_N=(1-f)^{N-1}$, where $f=\left(\frac{1}{2}\right)^{k-1}$, k is the sampling times. Suppose the probability is expected to be larger than λ, formally, $f_N=(1-\left(\frac{1}{2}\right)^{k-1})^{N-1}>\lambda$, we can get

$k > 1 - \log_2(1-\lambda^{\left(\frac{1}{N-1}\right)})$,where $\log_2(1-\lambda^{\left(\frac{1}{N-1}\right)}) < 0$.

Logarithmic dependence instructs that even if the probability is very high and the number of node pairs is very large, the request sampling times is limited. For instance, there are 20 sensor nodes sensing the target simultaneously (very dense) and the expected probability is 99% (very high), limited sampling times $k=16$ can satisfy the user's probability threshold.

5.2 Analysis of Tracking Error

We see that the tracking error contains two parts. One is caused by the distance between the located face of FTTT and the actual face that the target located in, which is called *inter-face error*. The other results come from the area of the faces, which is called *intra-face error*. As the number of the divided faces is $O(n^4)$, where n is the number of the nodes that participate in tracking, the tracking error mainly comes from the former factor. According to analysis in Section 4, the vector distance between two faces is approximately proportion to their geographical distance, formally, $GD(f_1, f_2) \propto VD(\vec{V}_s(f_1), \vec{V}_s(f_2))$. By reviewing the similarity between two vectors, it is obvious that the error between the located face and the actual face is corresponding to the similarity between the detecting vector and the signature vector. In order to simplify the analysis, the approximate vector distance is used to evaluate the inter-face error in the following part instead of geography distance.

As we have discussed that for only a pair of nodes, P_1, detecting the target, the probability that a grouping sampling can obtain all the flipped pairs and generate the correct sampling vector is $f_1 = 1 - f = 1 - \left(\frac{1}{2}\right)^{k-1}$. Then we can obtain the error caused by localizing the target to the incorrect faces.

Supposing the target appeared out of the uncertain area, the sampling vector is constant according to our assumption, and the inter-face error is 0. While the target is in the uncertain area, if we do not get the flipped pair from the sampling vector, the target may be localized into a wrong face which brings in error. For only one pair P_1 detecting the target, the probability has the property $f_{P_1} + f_{\overline{P_1}} = 1$, the expectation of the error E of the localization due to locating the target to a wrong face is $E = f_{P_1} \cdot 1 + f_{\overline{P_1}} \cdot 0 = \left(\frac{1}{2}\right)^{k-1} = f$.

Supposing the target appeared in the intersection of the uncertain area of N node pairs, formally, $P_1, P_2, ..., P_N$c. Then, for any node pair P_i, the returned information should be one of the above-mentioned two situations, P_i or $\overline{P_i}$. The sum of all the probabilities of possible sampling results is 1, formally

$$\begin{aligned}&(f_{P_1\wedge P_2\wedge P_3\wedge\cdots\wedge P_N} + f_{\overline{P_1}\wedge P_2\wedge P_3\wedge\cdots\wedge P_N} + f_{P_1\wedge\overline{P_2}\wedge P_3\wedge\cdots\wedge P_N} + f_{P_1\wedge P_2\wedge\overline{P_3}\wedge\cdots\wedge P_N} + \cdots + f_{P_1\wedge P_2\wedge P_3\wedge\cdots\wedge\overline{P_N}})\\&+(f_{\overline{P_1}\wedge\overline{P_2}\wedge P_3\wedge\cdots\wedge P_N} + f_{P_1\wedge\overline{P_2}\wedge\overline{P_3}\wedge\cdots\wedge P_N} + \cdots + f_{P_1\wedge P_2\wedge P_3\wedge\cdots\wedge\overline{P_{N-1}}\wedge\overline{P_N}}) + \cdots\cdots + (f_{\overline{P_1}\wedge\overline{P_2}\wedge\overline{P_3}\wedge\cdots\wedge\overline{P_N}}) = 1\end{aligned} \tag{9}$$

Considering all the node pairs are coequal, we only need to concern about how many node pairs report incorrect information. In other words, we conclude that the probability of all the situations that there are M node pairs reflecting incorrect information are equal, which is denoted by $(1-f)^{N-M} \cdot f^M$, besides, there are C_N^M different situations, which makes the value of the error to be M . Hence, the expectation of the error is: $E_N = \sum_{M=0}^{N} M \cdot C_N^M \cdot (1-f)^{N-M} \cdot f^M$.

The calculating process of the equation above can refer to Appendix II and the vector distance error is given by $E_N = N \cdot f$.

Focusing on an area when the target appears in, there are totally n nodes can sense the target and thus $N = C_n^2$ node pairs. Then the area can be divided into K faces by the uncertain boundaries of pairwise nodes pairs. We define K_i is the number of faces that is the inspection of the uncertain boundaries of i node pairs. It is obvious that $K = \sum_{i=1}^{N} K_i$.

The expectation of the inter-face error is presented as $Ev = \sum_{i=1}^{N} E_i \cdot \frac{K_i}{K}$. Then considering $E_i = i \cdot f$, the error is $Ev = \sum_{i=1}^{N} i \cdot f \cdot \frac{K_i}{K} = \frac{f}{K} \sum_{i=1}^{N} i \cdot K_i$.

Hence, an inequality can be conducted to represent the error

$$Ev = \frac{f}{K} \sum_{i=1}^{N} i \cdot K_i < N \cdot \frac{f}{K} \sum_{i=1}^{N} K_i = N \cdot f = C_n^2 \cdot f = O(f \cdot n^2)$$

At any time instant the target can be sensed by $n = \pi R^2 \cdot \rho$ sensor nodes, where R is the sensing range of sensors and ρ is the density of the sensor deployment. Consequently, there are C_n^2 node pairs. The πR^2 area (in which the sensors can sense the target) can be divided into $O(n^4)$ faces. Considering the influence of intra-face error, in worst-case the final tracking error is

$$E = \sqrt{\frac{Ev \cdot \pi R^2}{\xi n^4}} < \sqrt{\frac{C_{\pi R^2 \cdot \rho}^2 \cdot f \cdot \pi R^2}{\xi (\pi R^2 \cdot \rho)^4}} = O\left(\frac{1}{2^{(k-1)/2} \cdot \rho \cdot R} \right) \tag{10}$$

where ξ is a constant.

Therefore, we find out that the tracking error is dependent on the deployment density, the sensing range of sensor nodes, and the sampling times. Increasing sampling times and deployment density will reduce the tracking error. However, too dense deployment will worsen the communication ability of the sensor networks as well as the delay. Anyhow, high tracking performance can be easily achieved by setting proper and undemanding sampling times.

6. Strategy Extension

As we discussed before, when there is not any face whose signature vector is equal to the vector, the maximum likelihood matching method is proposed. However, sometimes we may find out that there exists more than one face with the maximum likelihood. For example, if the sampling vector is $[0,1,1,1,1,-1]$, $S_{\vec{V_d}, \vec{V_s}(f_1)} = S_{\vec{V_d}, \vec{V_s}(f_4)} = 1$. Hence, the face f_1 and f_4 both have the maximum similarity, as shown in Fig. 7. The target should be located to the location which is the mean value of all the candidate faces which have the maximum similarity.

However, it is obvious that the sampling information is not sufficiently used. Supposing one grouping sampling which contains k times continuous samplings, for a given pair of nodes n_1 and n_2, we can count the number of sample in which the RSS of n_1 is greater than n_2, $N_{(n_1,n_2)}$, and the number of sample in which the RSS of n_1 is smaller than n_2, $N_{(n_2,n_1)}$. It is easy to see that the sampling times k is equal to the sum of the two numbers, $k = N_{(n_1,n_2)} + N_{(n_2,n_1)}$. Then the probability of the ordinal pair appearing to be $(1,2)$ is given by $P_{(n_1,n_2)} = N_{(n_1,n_2)} / k$, and similarly $P_{(n_2,n_1)} = N_{(n_2,n_1)} / k$.

The node pair value should also be extended with the extra information of the probability. We define the extended node pair value as the probabilities of the sequential order of this pair subtract the probability of reverse order of this pair.

Definition 10 (Extended Node Pair Value): For any pair of nodes, n_i and n_j, $i < j$, supposing the sampling times is m. After comparing the strength of their sampling results, we derive the *Extended Node Pair Value* V'_{n_i,n_j} by $\forall_{i,j} V_{n_i,n_j} \in [-1,1]$ and $V_{n_i,n_j} = P_{(n_i,n_j)} - P_{(n_j,n_i)}$.

In this way, the sampling vector of different grouping sampling results are different. Additionally, when we calculate the similarity between the sampling vector and the signature vector of some face, each component (the extended node pair value) of the sampling vector is no longer qualitative, {-1, 0, 1}, but quantitative (distributing in the interval [-1, 1]). It not only contains the information that whether this pair of nodes have flipped in the grouping sampling but shows the flipped extent that how much the sequential order is more (or less) than the reverse order. Then we mainly eliminate the problem that the maximum similarity between the sampling vector and the signature vectors is not unique.

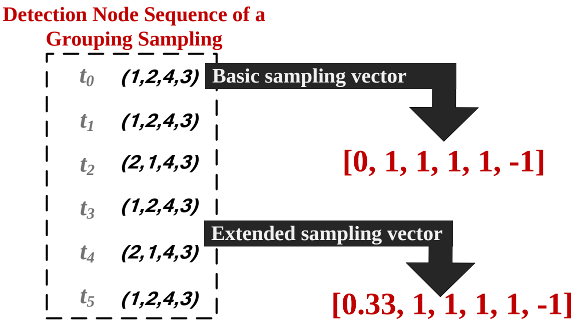


Fig. 9. Comparion Between Basic Sampling Vector and Extended Sampling Vector

For example, as shown in **Fig. 9**, this grouping sampling has six continuous samplings. For the node pair n_1 and n_2, there are four sequential orders, $(1,2)$, and two reverse orders, $(2,1)$. Hence, the basic sampling vector formed by basic node pair values is $[0,1,1,1,1,-1]$, while the extended sampling vector is turned into $[0.33,1,1,1,1,-1]$.

Therefore, if the extended sampling vector is used, we will get the similarity: $S_{\vec{V_d},\vec{V_s}(f_1)}=1.5$, $S_{\vec{V_d},\vec{V_s}(f_2)}=1/\sqrt{(2/3)^2+1}\approx 0.832$, $S_{\vec{V_d},\vec{V_s}(f_3)}=1/\sqrt{(4/3)^2+1}=0.6$, $S_{\vec{V_d},\vec{V_s}(f_4)}=1/\sqrt{(1/3)^2+1}$ ≈ 0.949, $S_{\vec{V_d},\vec{V_s}(f_5)}=1/\sqrt{(2/3)^2+1+1}\approx 0.640$ and $S_{\vec{V_d},\vec{V_s}(f_6)}=1/\sqrt{(4/3)^2+1+1}\approx 0.514$.

As shown in the results, we find out that only face f_1 has the maximum similarity ($S_{\vec{V_d},\vec{V_s}(f_1)}=1.5$), so the target should be located in this face. After adopting the extended sampling vector, the similarities are various from each other, i.e., the similarities between any extended sampling vector and signature vectors of different faces are various. Hence, we can easily select the only face with the maximum similarity to determine the correct location of the target.

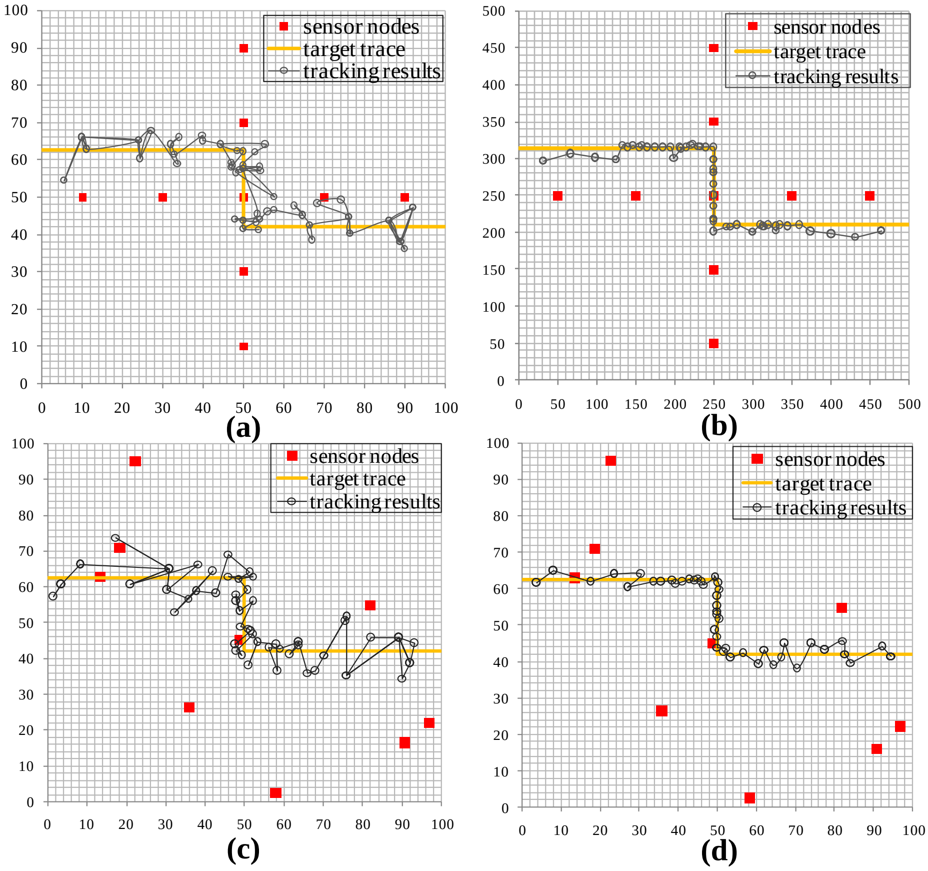


Fig. 10. Tracking by FTTT and PM under regularly and randomly sensor deployment. (k =5, ε =1)

Table 1. System Parameters and Settings

Parameter	Settings
Filed Size	$100 \times 100m^2$
Noise Model Parameter	$\beta = 4, \sigma_X = 6$
Number of Sensor Nodes (n)	$5 \sim 40$
Sensing Range (R)	$40m$
Sensing Resolution (ε)	$0.5 \sim 3dBm$
Sampling Rate (λ)	$10Hz$
Target Velocity	$1 \sim 5m/s$
Sampling Times	$3 \sim 9times$

7. Performance Evaluation

We conduct simulation experiments to compare the proposed Fault-Tolerant Target-Tracking (FTTT) strategy with Direct maximum likelihood estimation (Direct MLE) [24] tracking scheme and the optimal path matching strategy with MLE (PM) proposed in [22]. In the simulation, we divide the monitor area by grid division according to our previous work [29].

A trace of the mobile target is generated by the random waypoint mobility model [30]. Tracking error at one point in the trace is the geographic distance between the estimated position and the true position. The mean tracking error is average of the errors of all the points in the trace. Each tracking simulation lasts 60s. **Table 1** illustrates the detailed parameters settings.

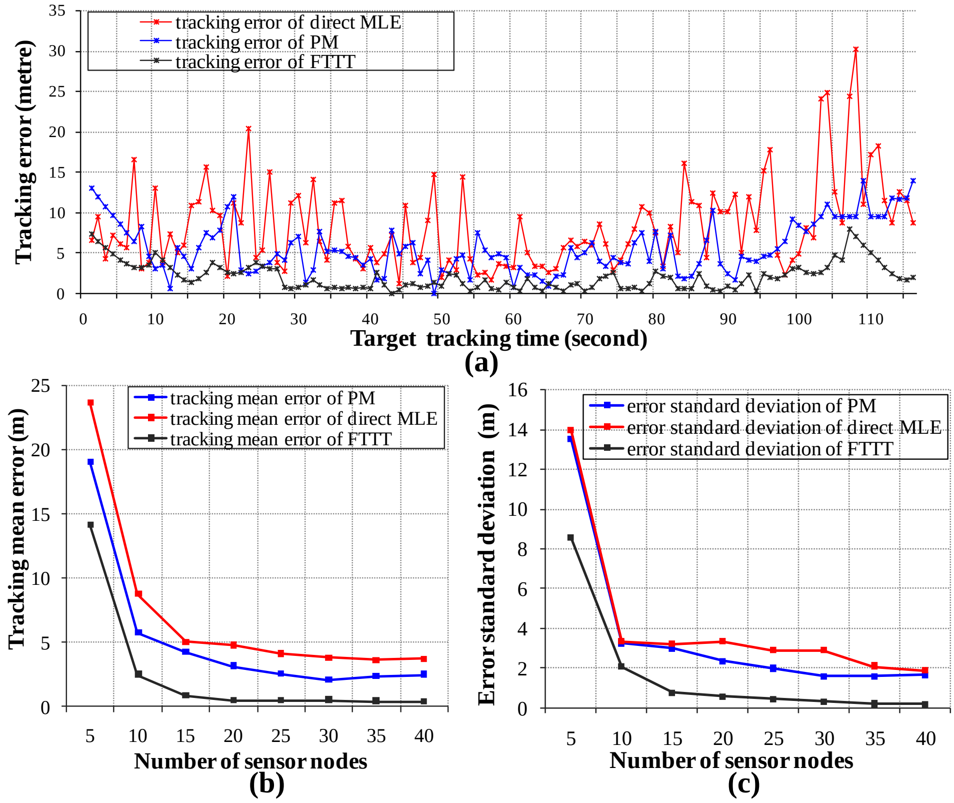


Fig. 11. (a) Dynamic Tracking Error Along Time Series (k =5, ε =1 and n =10). (b) and (c) Tracking Mean Error and Standard Deviation Varies with the Number of Sensors (k =5, ε = 1).

7.1 Tracking Simulation

This subsection gives an intuitive comparison between FTTT and PM strategy of a tracking example. **Fig. 10(a)** and **Fig. 10(b)** show all the position points estimated by PM and FTTT under the tracking example in which sensors are deployed in grid, respectively. **Fig. 10(c)** and **Fig. 10(d)** is the tracking results of PM and FTTT when the sensors are randomly deployed under uniform distribution.

7.2 Performance Simulation

Fig. 11(a) shows the dynamic tracking errors of FTTT, PM and Direct MLE along with time series, which further proves the conclusion that the tracking performance of FTTT is much greater than the other two. The following simulations illustrate the impact of tracking error.

1) Tracking Performance Comparison with Different Number of Sensor Nodes: We compare the FTTT strategy with PM and Direct MLE methods under different number of random deployed sensor nodes, which varies from 5 to 40. **Fig. 11(b)** and **Fig. 1 Fig. 11(c)** show the mean tracking error and standard deviation of error, respectively. Simulation depicts that 1. by increasing number of deployed sensor nodes, the tracking error and its standard deviation are both reduced, and when sensor nodes is less than 10, the reducing degree is great; 2. the FTTT strategy performs much better than PM and Direct MLE.

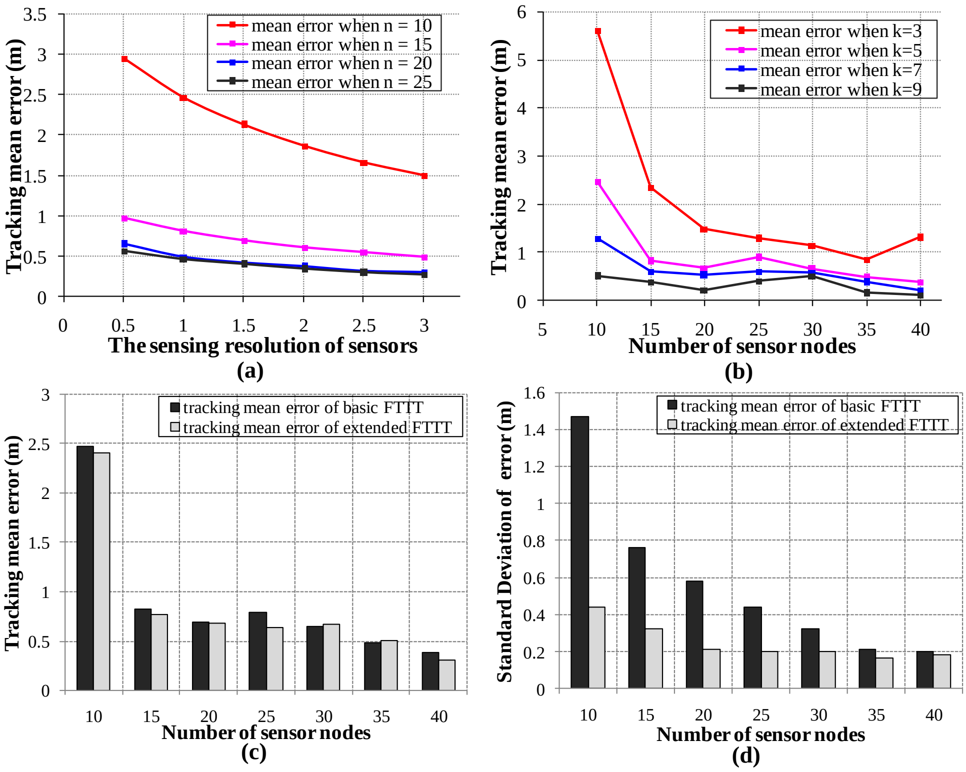


Fig. 12. (a) Impact of Sensing Resolution (k =5); (b) Impact of Sampling Times (ε =1); (c) and (d) Comparison of Mean Error and Standard Deviation between basic and extended FTTT (k =5, ε =1).

2) Impact of Sensing Resolution: **Fig. 12(a)** shows the mean tracking error of FTTT changes with the sensing resolution ε when randomly deploying 10,15,20 and 25 sensor nodes. It shows that 1. under the same noise model parameter (β and σ_X), lower sensing resolution leads to higher tracking accuracy; 2. for a given sampling times (k = 5), the less sensors deployed, the more quickly the tracking error is reduced along with the reduction of sensing resolution. Besides, when the number of sensor nodes is larger than 20 (when sampling times), the tracking errors have not yet been sensitive with the sensing resolution.

3) Impact of Sampling Times: **Fig. 12(b)** shows the mean error of FTTT strategy when the number of sensor nodes varies from 10 to 40 under different sampling times k = 3, 5, 7, 9. Results indicate 1. increasing the sampling times k will reduce the tracking error; 2. when the number of sampling times is very limited, but the number of sensor nodes is great, the tracking error will be increased along with the number of sensors. This is because when there are many sensors participanting in tracking, the information of uncertain area cannot be obtained effectively under limited sampling times.

4) Effectiveness of Strategy Extension: When the sensing resolution ε = 1 and the sampling times k =5, **Fig. 12(c)** compares the mean error of basic FTTT with the extended

FTTT and Fig. 12(d) compares their standard deviation of tracking errors. It is obvious that although the extended FTTT does not ultimately reduce the tracking error, it reduces the error deviation (when $n=10$, the standard deviation of extended FTTT is 79% less than the basic FTTT), which makes returning trajectory much smoother and the tracking process more robust. This is ultimately consistent with our analysis in Section 6, and the quantifying the sampling vector's uncertainty in extended FTTT improves the tracking efficiency.

7.3 System Evaluation

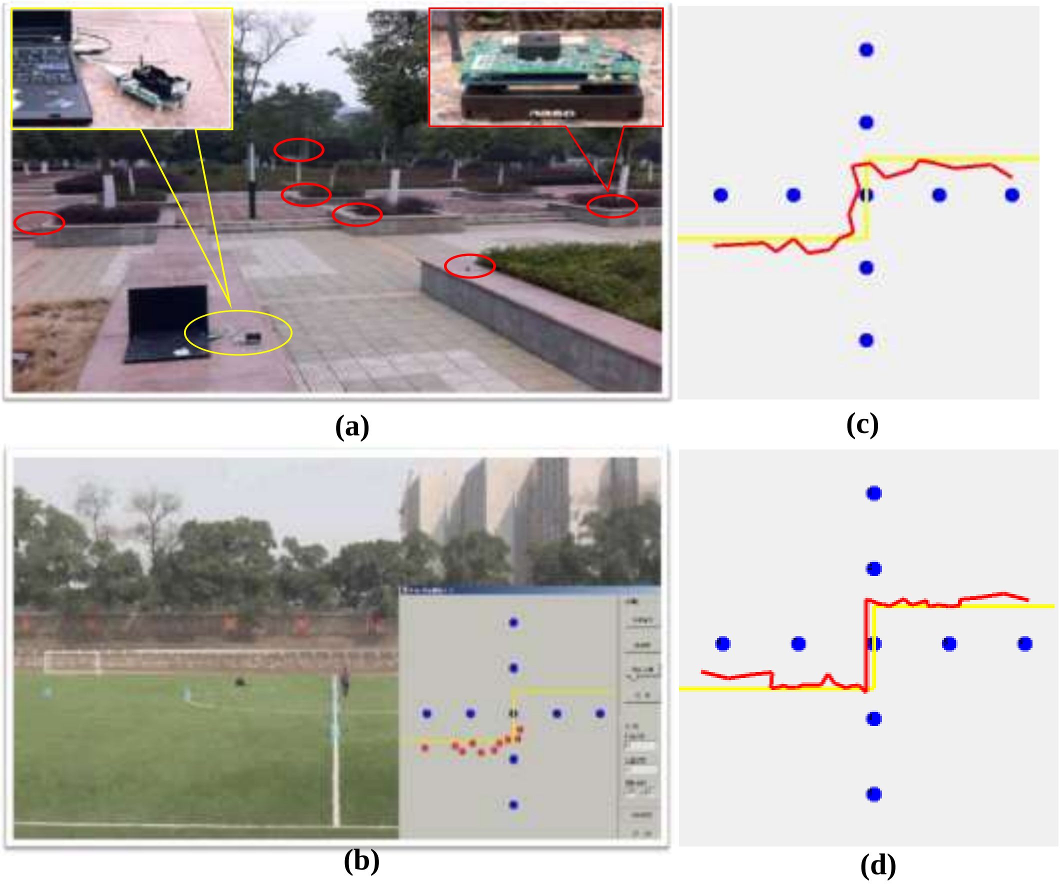


Fig. 13. Outdoor System Evaluation

FTTT is implemented by using the Crossbow IRIS (XM2110) motes with the MTS300 sensor boards outdoor, as shown in Fig. 13(a). 10 IRIS motes are used and 9 of them are deployed as a cross “ + ” shape in the playground of our campus to be the ***sensor nodes*** for tracking. One person carries the remaining mote to simulate a ***mobile target***. The target moves along a “_┌‾” shape trace at changeable velocity in 1~5m/s, and its 4 kHz fixed frequency piezoelectric resonator continuously send signal. The deployed sensors receive the signal strength and send them to base station via MIB520 USB interface board. Fig. 13(b) shows the target trajectory is real-time generated alone with the target moving. Fig. 13(c) and Fig. 13(d) illustrate the tracking results of basic and extended FTTT, respectively.

We can see from Fig. 13(c) and Fig. 13(d) that both the basic and extended FTTT perform target tracking very well. For the basic FTTT, since the target velocity is optional instead of programmed and the noise always exists, the tracking error is in-and-out. However, even the maximum tracking error is acceptable. For the extended FTTT, it makes the tracking trajectory much smoother than the basic FTTT, especially at the turning corner where the tracking error should be greater. From the outdoor real environment evaluation, we can find out that FTTT is robust and tracks the moving target accurately. FTTT not only overcomes uncertainty of the environment and the target, but makes full use of the uncertain information to improve the tracking efficiency.

8. Conclusion

To the best of our knowledge, this appear to be the first paper to use unreliability to improve the efficiency of target tracking by including uncertain area boundaries of node pairs to demark monitor area into faces and use multiple times grouping samplings to obtain relative relationship of pairwise nodes' RSS. The strategy proposed in this paper transforms the tracking problem into a vector matching process which improves tracking flexibility while reducing the influence of in-the-field factors and results in improved tracking accuracy and robustness Our evaluation of theoretical and experimental results demonstrates that this strategy outperforms other related methods.

Appendix

Appendix I: Proof of the Probability Equation

If there are totally N node pairs in the grouping sampling, We can know from the equation (8) that the probability that the grouping sampling can get all the flipped information in basic approximate target-tracking strategy is given by $f_N = \sum_{M=0}^{N} (-1)^M \cdot C_N^M \cdot f^M$. We conduct the equation $f_N + f \cdot f_{N-1} = \sum_{M=0}^{N} (-1)^M \cdot C_N^M \cdot f^M + f \cdot \sum_{M=0}^{N-1} (-1)^M \cdot C_{N-1}^M \cdot f^M = \sum_{M=0}^{N-1} (-1)^M \cdot C_{N-1}^M \cdot f^M$.

We can see that $f_N + f \cdot f_{N-1} = f_{N-1}$, then $f_N = (1-f) \cdot f_{N-1}$, where $0 < f < 1$. Besides, it easy to get that $f_1 = 1 - f$. Hence, we can get our probability equation $f_N = (1-f)^N$. ■

Appendix II: Proof of the Localization Error Equation.

Since when the target appear in the intersection of N node pairs' uncertain area, the expectation of the error is

$$E_N = \sum_{M=0}^{N} M \cdot C_N^M \cdot (1-f)^{N-M} \cdot f^M \text{ and } E_{N-1} = \sum_{M=0}^{N-1} M \cdot C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M$$

Considering $C_N^M - C_{N-1}^M = C_{N-1}^{M-1}$ and $C_N^N = C_{N-1}^{N-1} = 1$, we can construct the equation

$$\begin{aligned}
& E_N - E_{N-1} = \left[E_N - (1-f) \cdot E_{N-1}\right] - f \cdot E_{N-1} \\
&= \left[\sum_{M=0}^{N} M \cdot C_N^M \cdot (1-f)^{N-M} \cdot f^M - (1-f) \cdot \sum_{M=0}^{N-1} M \cdot C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M\right] - f \cdot \sum_{M=0}^{N-1} M \cdot C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M \\
&= \left[\sum_{M=0}^{N-2} (M+1) \cdot C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^{M+1} + (C_{N-1}^{N-1} \cdot N \cdot f^N)\right] - f \cdot \sum_{M=0}^{N-1} M \cdot C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M \\
&= f \cdot \sum_{M=0}^{N-1} (M+1) \cdot C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M - f \cdot \sum_{M=0}^{N-1} M \cdot C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M \\
&= f \cdot \sum_{M=0}^{N-1} C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M
\end{aligned}$$

According to equation (9), the sum of all probabilities is 1, i.e., $\sum_{M=0}^{N-1} C_{N-1}^M \cdot (1-f)^{N-M-1} \cdot f^M = 1$.

We obtain that $E_N - E_{N-1} = f$ .And according to above-mentioned $E_1 = f$, it is obvious that $E_N = N \cdot f$. ■

References

[1] D. Li, K. Wong, Y.H. Hu and A. Sayeed, "detection. classification and tracking of targets in distributed sensor networks, " *IEEE Signal Processing Magazine*, vol.19, no.2, 2002. Article (CrossRef Link)

[2] A. Smith, H. Balakrishnan, M. Goraczkoet and N. Priyantha, "tracking moving devices with the cricket location system," *MobiSys'04* , pp.190-202, 2004. Article (CrossRef Link)

[3] F. Gustaffsson and F. Gunnarsson, "mobile positioning using wireless network: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol.22, no.4, 2005. Article (CrossRef Link)

[4] B. Kusy, A. Ledeczi and X. Koutsoukos, "Tracking mobile nodes using RF Doppler Shifts," *Sensys'07*, pp.29-42, 2007. Article (CrossRef Link)

[5] S. Mohanty, "VEPSD: A novel velocity estimation algorithm for next- generation wireless systems," *IEEE Trans. on Wireless Com.*, vol.4, no.6, 2005. Article (CrossRef Link)

[6] A. Terzis, A. Anandarajah, K. More and I.J. Wang, "Slip surface localization in wireless sensor networks for landslide prediction," *IPSN'06*, pp.109-116, 2006. Article (CrossRef Link)

[7] W. Xi, Y. He, Y.H. Liu, J.Z. Zhao, L.F. Mo, Z. Yang, J.L. Wang and X.Y. Li, "Locating sensors in the wild: pursuit of ranging quality," *Sensys'10*, pp.295-308, 2010. Article (CrossRef Link)

[8] Q.Q. Ren, J.Z. Li and S.Y. Cheng, "target tracking under uncertainty in wireless sensor networks," *IEEE ACM Trans. Networking*, MASS'11, pp.430–439, 2011. Article (CrossRef Link)

[9] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks," *RealWSN'08*, pp.1-5, 2008. Article (CrossRef Link)

[10] W. Wang, V. Srinivasan, B. Wang and K.C. Chua, "coverage for target localization in wireless sensor networks," *IEEE Trans. on Wireless Comm.*, vol.7, no.2, pp.667-676, 2008. Article (CrossRef Link)

[11] B. Gaddi, H. Bracha, A. Tal, D. Danny and R. Boris, "continuous close-proximity rssi-based tracking in wireless sensor networks," in *Proc. of 2010 Int'l Conf. on Body Sensor Networks*, pp.234-239, 2010. Article (CrossRef Link)

[12] D. Moore, J. Leonard, D. Rus and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. of SenSys'04*, 2004. Article (CrossRef Link)

[13] K. Whitehouse, C. Karlof and D. Culler, "A Practical Evaluation of Radio Signal Strength for Ranging-based Localization," *SigMobile'07*, vol.11, no.1, 2007. Article (CrossRef Link)

[14] Z.J. Wang, E. Bulut and B.K. Szymanski, "distributed energy-efficient target tracking with binary sensor networks," *TOSN'10*, vol.6, no.4, 2010. Article (CrossRef Link)

[15] Z. Abbasi1, A. Farahi and H. H. S. Javadi, "An improvement in maximum likelihood estimation algorithm in sensor networks," *International Journal of Computer Science & Engineering Survey*, vol.2, no.1, Feb.2011. Article (CrossRef Link)

[16] X.R. Li and V.P. Jilkov, "A survey of maneuvering target tracking: approximation techniques for nonlinear filtering," *SPIE'04*, 2004. Article (CrossRef Link)

[17] P. Zhang and M. Martonosi, "LOCALE: collaborative localization estimation for sparse mobile sensor networks," *IPSN'08*, 2008. Article (CrossRef Link)

[18] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Journal of Basic Engineering*, 1960. Article (CrossRef Link)

[19] B. Ristic, S. Arulampalam and N. Gordon, "Beyond the Kalman Filter: Particle Filters for Tracking Applications," 2004. Article (CrossRef Link)

[20] J. Liu, J. Reich and F. Zhao, "Collaborative in-network processing for target tracking," *Journal on Applied Signal Processing*, Mar.2003. Article (CrossRef Link)

[21] J. Ting, H. Snoussi and C. Richard, "Decentralized variational filtering for target tracking in binary sensor networks," *IEEE Trans. on Mobile Computing*, vol.9, no.10, pp.1465-1477, 2010. Article (CrossRef Link)

[22] Z. Zhong, T. Zhu, D. Wang and T. He, "Tracking with Unreliable Node Sequences," *IEEE InfoCom'09*, pp.1215-1223, April 2009. Article (CrossRef Link)

[23] Z. Zhong and T. He, "MSP: Multi-Sequence Positioning of Wireless Sensor Nodes," *Sensys'07*, 2007. Article (CrossRef Link)

[24] K. Yedavalli and B. Krishnamachari, "Sequence-Based Localization in Wireless Sensor Networks," *IEEE Trans. on Mobile Computing*, vol.7, no.1, pp.81-94, 2008. Article (CrossRef Link)

[25] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks," *IEEE Trans. on Mobile Computing*, 1996. Article (CrossRef Link)

[26] H. Lord, W.S. Gatley and H.A. Evensen, "Noise Control For Engineers", *McGraw Hill Book Co*., 1980. Article (CrossRef Link)

[27] C.S.Ogilvy, "Excursions in Geometry", *Dover*, 1990. Article (CrossRef Link)

[28] D. M. OFRIM, D. I. SĂCĂLEANU, B. A. OFRIM and R. STOIAN, "Implementation of an Adaptive Synchronizing Protocol for Energy Saving in Wireless Sensor Networks," International Journal of Communications, vol.4, no.4, 2010. Article (CrossRef Link)

[29] G.M. Tang, G.X. Zhou, Y. Xie, D.Q. Tang and J.Y. Tang, "Target Localization Based on Double-level Grid Division in Wireless Sensor Networks," *Computer Science*, vol.39, no.6, pp.25-29, 2012. Article (CrossRef Link)

[30] D.B. Johnson and D.A. Maltz," Dynamic Source Routing in Ad hoc Wireless Networks", *IEEE Trans. on Mobile Computing*, 1996. Article (CrossRef Link)

Yi Xie received the B.S. and M.S. degree from National University of Defense Technology (NUDT) in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree in the Science and Technology Information Systems Engineering Laboratory of NUDT, Changsha, China. He is a professional member of ACM and Chinese Computer Federation (CCF). His current research interests are in target-tracking wireless sensor networks, data management in sensor networks, mobile sensor networks and information resource management in wireless networks.

Guoming Tang Guoming Tang received the B.S. and M.S. degree from National University of Defense Technology (NUDT) in 2010 and 2012, respectively. Recently, he has earned the funding supporting from China Scholarship Council (CSC), and is going to pursue his Ph.D. degree in Department of Computer Science, Faculty of Engineering, University of Victoria, Canada. His current research interest is target tracking strategies in wireless sensor networks.

Daifei Wang received the B.S. degree from Central South University in 2011. She is currently pursuing the M.S. degree in the Science and Technology Information Systems Engineering Laboratory of National University of Defense Technology, Changsha, China. Her current research are mainly on target-tracking wireless sensor networks, and information resource management in wireless networks.

Weidong Xiao received the B.S. and M.S. degree from National University of Defense Technology (NUDT) in 1990 and 1998, respectively and obtained his Ph.D. there in 2004. He is currently an Professor with Information resource management and also a supervisor of Ph.D. candidates in the Science and Technology Information Systems Engineering Laboratory of NUDT. He is a director of Chinese Institue of Manament Science and Engineering. His current research interests are in data management in sensor networks, information management in ubiquitous networks and Internet of Things (IoTs).

Daquan Tang received the B.S. and M.S. degree from National University of Defense Technology (NUDT) in 1994 and 2002, respectively. He is currently an Professor at College of Information System and Management of NUDT, where he obtained his Ph.D. in 2009. He is the senior member of the Chinese Institue of Electronics. His current research interests are in data management in sensor networks, spatio-temporal data management and target-tracking wireless sensor networks.

Jiuyang Tang received the B.S. degree from National University of Defense Technology (NUDT) in 2000. He is currently an Associate Professor at College of Information System and Management of NUDT, where he obtained his Ph.D. in 2006. His current research interests are in topology control in sensor networks, P2P, ubiquitous networks and Internet of Things (IoTs).