2011 31st International Conference on Distributed Computing Systems

Bubble Trace: Mobile Target Tracking under Insufficient Anchor Coverage

Pengpeng Chen*, Ziguo Zhong and Tian He
Computer Science and Engineering
University of Minnesota
200 Union Street SE, Minneapolis, MN, USA
Email: {pengpeng, zhong, tianhe}@cs.umn.edu

***Abstract*—As an essential requirement for surveillance systems, target tracking has been studied extensively. Most of the tracking schemes are based on trilateration, which requires each point in the monitoring area to be covered by at least three anchors. However, due to the inadequate deployment of costly anchors and environment constraints, the target might not always be detected by three or more anchors simultaneously, resulting in intermittent localization failures and performance degradation. To address this issue, this paper proposes a tracking method called Bubble Trace (BT) for insufficient anchor coverage and asynchronous networks. By fully extracting the location information embedded in dual, single and zero anchor coverage, we develop a bi-directional bounding algorithm to offer the bubble-shaped regions that indicate the possible locations of the target. Moreover, instead of separately estimating each position point of the target, we construct the trace by finding a maximum-likelihood path in a graph. The design is evaluated through extensive simulation and a test-bed experiment with 20 MicaZ nodes. Results show that the proposed scheme improves the tracking accuracy without using additional hardware under insufficient anchor coverage.**

I. Introduction

Target tracking, which can provide a spatio-temporal description of mobile objects, is one of the most important applications of wireless network. Nowadays lots of services benefit greatly from the availability of accurate tracking, in particular emergency rescue, hospital healthcare, aerospace systems and so on. However, due to costly hardware equipments and environment constraints, how to track the mobile target more accurately and reliably using finite system resources still remains to be one of the challenging issues.

Many excellent methods have been proposed for target tracking, which mainly involve the lower-layer signal processing and the upper-layer network protocol design. While these schemes are effective, they suffer from one or more constraints in the following. First, at least three anchors are required to cover any point in the service area for the trilateration-based schemes [1][2][3], which results in heavy infrastructure cost. If insufficient anchors are deployed, the tracking accuracy will be degraded seriously. Second, the model based tracking methods [4][5][6] like Kalman filter and Bayesian networks often bring a complex system design. And some movement or noise assumptions on the mobile target might be unrealistic in real applications. Third, additional hardware is required on the target for ranging purpose [7][8][9], which obviously increases the system cost.

Considering the limitations of existing work, instead of focusing on improving the tracking accuracy in dense anchor network, we are interested in performing target tracking in a more general environment where the anchor coverage is inadequate, the target can move arbitrarily, and no special hardware on the target is available. And we apply an asymmetrical network architecture in this paper, where the mobile target with limited resources (i.e., CPU, battery, etc) simply emits signals periodically and anchor nodes that are comparatively powerful and randomly deployed in the map are in charge of detecting signals from the target with high time resolution. In this context, existing trilateration-based approaches would suffer from severe performance degradation with the decreasing number of anchors. To deal with this problem, we propose a novel target tracking framework to make full use of the detection information for localization under insufficient anchor coverage. Specifically, our major contributions are as follows:

- To the best of our knowledge, we are the first to investigate the target tracking problem under insufficient anchor coverage, which is based on the arrival time measurements of the target signal.
- We introduce a generic tracking framework that eliminates additional ranging hardware at the target and accurate time synchronization among anchors. In this framework, a possible target area that looks like a bubble is firstly generated by a bi-directional bounding algorithm and then the tracking task is formulated as an optimal path searching problem in a graph.
- The performance of the proposed design is evaluated by extensive simulation studies. And a prototype system is implemented for verification with 20 MicaZ nodes.

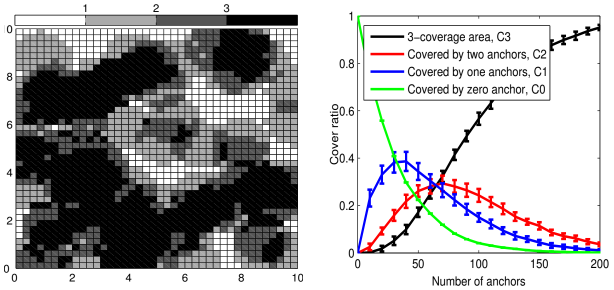
The rest of the paper is organized as follows. Section II describes the motivation behind the work. In Section III, we introduce the system overview. The detailed tracking algorithm is described in Section IV and V. Section VI presents an improvement which removes the limitation of time synchronization among anchors and discusses the tracking performance. The performance evaluation and a prototype system are given in Section VII. Finally, Section VIII surveys the related work and Section IX concludes the paper.

*Pengpeng Chen is a visiting student from Ocean University of China.

1063-6927/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCS.2011.56

II. Motivation

The nature of tracking determines that it is often not practical to add extra components such as accelerometers and magnetometers on the target for localization and movement determination. An alternative and more common approach is to utilize the time-difference-of-arrival (TDOA) of the targets' signal between pairs of anchors [10][11]. The main idea of traditional TDOA tracking method is as follows: a target emits a signal periodically and a number of anchors record the arriving time of the signal. Thus, one hyperbola can be computed by the arrival time measurement for each pair of anchors. With more than three anchors, more hyperbolas can be obtained. Then the target location is the intersection of multiple hyperbolas. Due to its simplicity and efficiency, the TDOA approach has widely been adopted for the wireless communication networks and wireless sensor networks.



(a) Area coverage with 80 anchors (b) Coverage ratio

Fig. 1. Anchor coverage.

A key requirement for the above method is 3-coverage of the monitoring field, which means each location point is covered by at least 3 anchors. Clearly the tracking quality is determined by the number of anchors covering each location in the field. Different coverage models and solutions are surveyed in [12][13]. In order to intuitively understand the needed number of anchors, we conduct the area coverage simulation in a $10r \times 10r$ rectangular region, in which anchors with detection radius r are randomly distributed. As shown in Fig. 1(a), the area coverage with 80 anchors is shown as an example, which displays the corresponding area covered by zero, one, two and more than three anchors, respectively. And Fig. 1(b) illustrates the coverage ratio under different numbers of anchors. We can see that 100 anchors can only achieve about 50% 3-coverage and 200 anchors are required in order to provide 96% 3-coverage of the whole field.

Obviously, a large number of anchors are needed to satisfy 3-coverage of a whole monitoring area, incurring high infrastructure cost. Besides, the deployment cost and the environmental limitations justify a partial 3-coverage of the monitoring filed. In this context, traditional trilateration methods suffer from severe performance degradation with the decreasing number of anchors. Hence the need for a new tracking design under insufficient anchor coverage arises, which can use the limited infrastructure resources to provide fairly location information. This forms the motivation for our work, that is to explore the position information embedded in dual, single or zero anchor coverage for improving tracking accuracy.

In addition, another requirement for the TDOA approach is that anchors must be precisely synchronized. Although many excellent ideas have been proposed for effective time synchronization in wireless networks (e.g., RBS [14], TPSN [15], FTSP[16], GTSP [17], etc), the synchronization performance is still not perfect due to the factors of delay, clock skew and drifting. The time synchronization errors will degrade the accuracy of the location estimation. Besides, synchronization among anchors is often not a disposable task and needed to be performed frequently during the whole tracking process, which introduces additional costs in many aspects. Hence it is necessary to design a method for releasing the synchronization requirement among anchors.

III. System Overview

A network model composed of mobile targets with unknown locations and static anchors is considered. The target emits signal each T seconds with a known and constant propagation speed c. We are unaware of the target's speed and direction, other than knowing the maximum speed v_{max}. The anchors which can accurately measure the signal arrival time are randomly deployed. And the number of anchors is not sufficient to achieve the 3-coverage of the whole monitoring area.

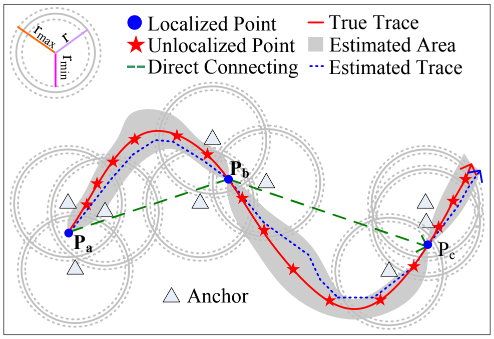


Fig. 2. A whole tracking framework.

Due to the sparse anchor support, the target can't always be localized during its movement. As shown in Fig. 2, the estimated trace by directly connecting localized positions under 3-coverage (i.e., P_a, P_b and P_c) fails to provide a sound tracking accuracy. Aiming at enhancing the tracking accuracy under the situation, we design a tracking framework which mainly includes two parts:

- Hyperbola-based Tracking with *Sufficient* Anchor Coverage. When the target is detected by three or more anchors during its movement, its locations (e.g., P_a, P_b and P_c in Fig. 2) are uniquely determined by the existing algorithms.
- Hyperbola-based Tracking with *Insufficient* Anchor Coverage. For the unlocated locations (pentagram points in Fig. 2) between two points P_a and P_b, a bi-directional bounding algorithm is developed to estimate a bubble region (grey area in Fig. 2) which bounds possible locations. Then the trajectory is constructed by finding a maximum-likelihood path (dotted line in Fig. 2) in the bubble region, instead of estimating each position separately.

Note that many existing works [18][19][20][21] for the target detection range assume a perfect disk model. However, this assumption does not hold well in practice due to ambient noise and heterogeneous signal attenuation patterns in realistic environments. To ensure that our design is as true to reality as possible, we assume a model with the upper and lower bounds on detection range, which is also adopted in [22]. Beyond the upper bound r_{max}, the anchor can't receive the signal from the target; and within the lower bound r_{min}, the target is guaranteed to be detected. Between two boundaries, the target is detected stochastically. When r_{max} equals r_{min}, our generic model degrades to the commonly used one.

In the following we first present the main design based on traditional TDOA measurements from synchronized anchors. The improvement of transforming arrival time measurements to distance differences without time synchronization among anchors is described later in Section VI. In addition, since the focus of the paper is to provide a tracking solution under insufficient anchor coverage, we only consider a single target for simplicity. Multiple targets can be tracked if anchors can separate the signals transmitted by each target.

IV. Tracking with Sufficient Anchor Coverage

In this section, we introduce the hyperbola-based tracking design when the target is detected by more than three anchors.

A hyperbola can be defined as the locus of points where the difference of the distances to the two foci is a constant. One of the hyperbola's applications is the location determination, using the time differences from two synchronized anchors. If the target is detected by n $(n \geq 4)$ anchors, we have $\binom{n}{2}$ distance differences and obtain $\binom{n}{2}$ hyperbolas:

$$\Delta d_{i,j} = (t_j - t_i) \cdot c, 1 \leq i < j \leq n \tag{1}$$

where t_i, t_j are the signal arrival time measured from anchors i and j, respectively. Then the target location is the intersection of these hyperbolas.

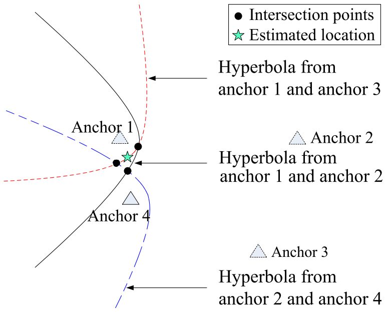

● Intersection points
☆ Estimated location
Hyperbola from anchor 1 and anchor 3
Anchor 1
Anchor 2
Hyperbola from anchor 1 and anchor 2
Anchor 4
Anchor 3
Hyperbola from anchor 2 and anchor 4

Fig. 3. Non-uniqueness of hyperbola intersections.

As shown in Fig. 3, three hyperbolic curves from four anchors often can't ideally intersect in one unique point, due to the TDOA measurement uncertainty. A simple solution for the target location determination is to calculate the weighted average of these points. However, it is a bit complicated to find the correct weight. Furthermore, each pair of hyperbola curves can have no, one or two intersections, the logic to find the correct one is non-trivial. In our tracking framework, given the anchor location (X_i, Y_i), we determine the target position (X, Y) by solving the following non-linear squares estimate.

$$(\hat{X}, \hat{Y}) = \arg\min_{X,Y} \sum_{i<j} (\Delta d_{i,j} - h(X, Y; X_i, Y_i, X_j, Y_j))^2 \tag{2}$$

where $h(X, Y; X_i, Y_i, X_j, Y_j)$ represents the distance difference from point (X, Y) to anchors i and j.

If the target is detected by three anchors, two hyperbolic functions can be obtained. In general, two hyperbolas may have four intersection points, but two hyperbolas that correspond to distance differences to a source have at most two intersections [23]. In this case, we can use past and new received information to eliminate the impossible location.

V. Tracking with Insufficient Anchor Coverage

Our main objective is to bound the possible area a target travels between two uniquely determined positions, and then to find a maximum-likelihood path from the bounding area. We first present an overview of a four-step design, followed by a detailed description of each step in the following sections.

A. *Design Overview*

Without loss of generality, we define a start point and an end point as P_s and P_e, which are uniquely localized positions of the target at time t_s and t_e by utilizing three or more anchors. During the interval between t_s and t_e, the location of the target can't be uniquely determined. We develop a bi-directional bounding algorithm to form a bubble region which consists of possible locations, and then identify a maximum-likelihood trace from the bubble region. The whole working process includes four steps:

- **Step 1:** Forward Tracking. Starting from point P_s at time t_s, the tracking algorithm recursively estimates the possible positions of the mobile target at each step based on signal detections and previous estimations.
- **Step 2:** Backward Tracking. When the target arrives at the point P_e, backward tracking is performed for reverse tracking. History information is employed to estimate the previous target locations until the start point P_s.
- **Step 3:** Bubble Formation. By intersecting the results obtained from the above two steps, we can obtain a bubble region, which contains possible target locations during the interval t_s and t_e.
- **Step 4:** Trace Identification. Instead of separately estimating each position of the target, we construct the trace by finding a maximum-likelihood path in the bubble region.

B. *Step 1: Forward Tracking*

Briefly, the forward tracking includes a prediction step and a filtering step. In the prediction step, the possible locations of target are predicted based on the previous locations and its movement. In the filtering step, we use new observations to eliminate the impossible predicted locations. We first describe the target tracking case in which a target is heard by two anchors, followed by the cases with one or zero anchor.

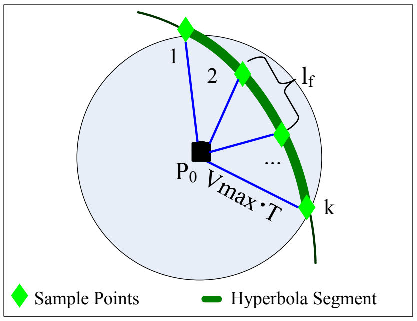


Fig. 4. Hyperbola dividing.

1) Tracking with Two Anchors: Without loss of generality, we assume that the target S is uniquely localized at a certain position covered by three anchors. As shown in Fig. 4, we use P_0 to represent the location at time t, where t denotes the discrete time. Based on this initial target position P_0, we are certain that the next position of target is contained in the circular region with origin P_0 and radius $v_{max} \cdot T$. If the target is heard only by two anchors, the measurement of time difference of arrival between these two anchors confines the possible locations of the target within a hyperbola curve. Therefore, at time $(t+1)$ we can confine the possible target locations within a hyperbola branch. As shown in Fig. 4, the final possible target locations should be located on the hyperbola segment which is contained in the circular region. To simplify computation, we choose k discrete points from the segment as the estimated target positions at time $(t+1)$. These discrete points are placed evenly along the segment with predefined fragment interval l_f. The length of fragment interval l_f can be determined based on accuracy requirement. Its value would not make a significant impact on the tracking performance, as indicated by our evaluation in the Section VII.

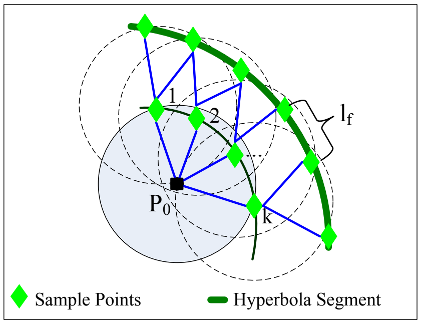


Fig. 5. Sample points selection.

Fig. 5 illustrates how we predict the position of the target S at time $(t+2)$. Clearly, we can conclude that the target position at time $(t+2)$ should lie in the k circle regions, each of which is centered at one possible point at time $(t+1)$. With a new TDOA measurement at time $(t+2)$, the possible target locations are decayed on the hyperbola segment contained within the k circular regions. In the same way, we select several points from the hyperbola segment according to the fragment length l_f, and a new group of possible target locations at time $(t{+}2)$ can be obtained. We use edges between the estimated points at time $(t{+}1)$ with the estimated points at time $(t{+}2)$ as the possible target traces, as long as the distance between them is smaller than $v_{max} \cdot T$. The above operation will be performed recursively for the subsequent time steps.

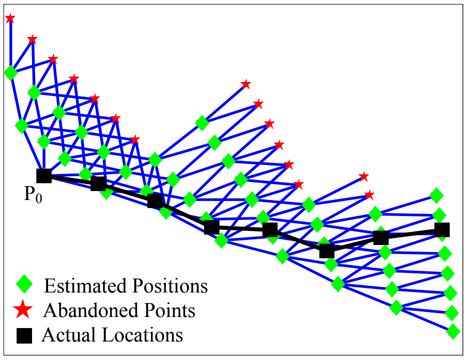


Fig. 6. All paths before cutting.

Fig. 6 illustrates a recursive seven-step tracking with $k = 5$ in the first step. A target location is said to be feasible at time $(t+j)$, if this location satisfies two constraints: (i) it is on the hyperbolas branch determined by the TODA measurement at time $(t+j)$, and (ii) the distance between this location and at least one of previous possible locations is less than $v_{max} \cdot T$. In Fig. 6, diamond-shape points are feasible points and star-shape points are infeasible points. A target path is said to be feasible at time $(t+j)$ if this path satisfies one condition: all points along the path up to time $(t+j)$ are feasible points.

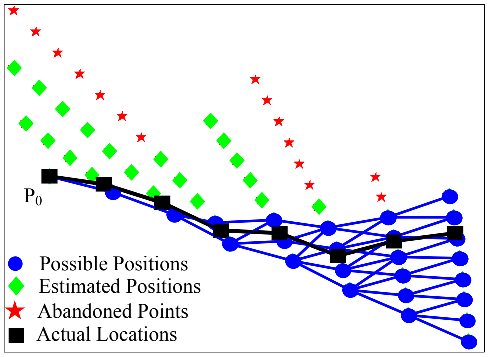
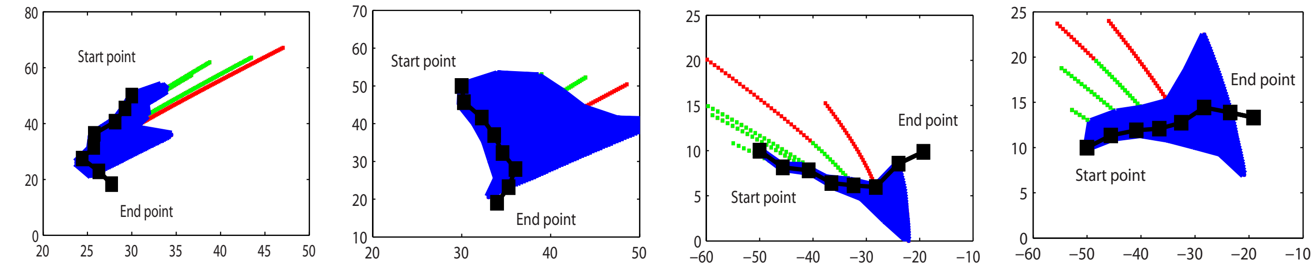


Fig. 7. Final possible paths after cutting.

The filtering step is then straightforward. If a point is located only within infeasible paths, it shall be removed. Fig. 7 displays the final estimated traces composed of feasible location points (circle points). We note because our filter operation is done at the path-level instead of at individual points, the number of estimated target locations at each step grows moderately, leading to good tracking performance.

To further investigate the performance of our step-by-step tracking, Fig. 8 shows simulation results of four tracking cases with different types of target trajectories. In all these cases, the target generates signals with same intervals and moves at random moving speeds. The shaded region indicates the possible target positions, while the solid square represents the actual location of the target when it generates signals. From these four tracking cases and many other cases we studied, we have the following observation:



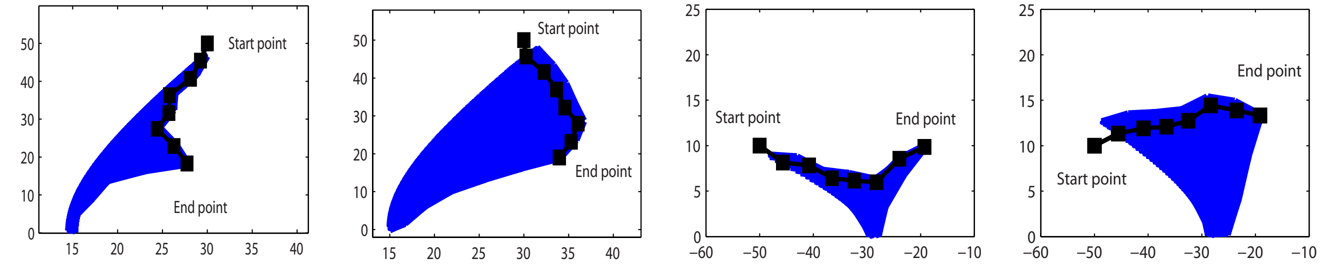
(a) Moving anti-clockwise from top to bottom

(b) Movig clockwise from top to bottom

(c) Moving anti-clockwise from left to right

(d) Moving clockwise from left to right

Fig. 8. Illustrations for forward tracking.



(a) Backward tracking from bottom to top

(b) Backward tracking from bottom to top

(c) Backward tracking from right to left

(d) Backward tracking from right to left

Fig. 9. Illustrations for backward tracking.

Observation 1. *During recursive tracking, the number of estimated target locations at each step doesn't always increase significantly. This is because our filter process removes infeasible points and all associated infeasible paths.*

Besides the above observation, the predicted locations which are inconsistent with new measurements will be eliminated at each time step. Let A_s be the set of all anchors, and A and B are the two anchors which detect the target. The filtering is performed based on the below facts:

- The target is simultaneously detected by anchors A and B. That is, $d_{sa} \leq r_{max}$ and $d_{sb} \leq r_{max}$, where d_{sa} and d_{sb} denote the distances from the target S to anchors A and B, respectively. If a predicted point can't satisfy the above condition, it should be filtered.
- The target isn't detected by any other anchor except A and B. $\forall$ anchor $i \in \{A_s - \{A, B\}\}$, $d_{si} > r_{min}$.
- The target lies on a branch of the hyperbola, which can be judged based on the sign of the difference $(d_{sa} - d_{sb})$.

When the target is no longer detected by three or more anchors after one unique localization, we can get a series of possible target positions for each time step by the forward tracking, which can reflect the basic target movement trend. Moreover, based on the obtained locations a real-time target trace can be constructed by finding a maximum-likelihood path, which will be described in the later subsection.

2) Tracking with One or Zero Anchor: Given the set of possible locations computed in the previous step, we can first predict that their next positions are contained in several circular regions with radius $v_{max} \cdot T$. Next we select position points uniformly from these circular regions. If there is only one anchor A detecting the target, the elimination of impossible sample points is based on the below facts:

- The target location is covered by anchor A. Namely, $d_{sa} \leq r_{max}$. A sample point is invalid if the distance between it and anchor A is larger than r_{max}.
- The target isn't heard by any other anchor except A. That is to say, $\forall$ anchor $i \in \{A_s - \{A\}\}$, $d_{si} > r_{min}$. If a point lies in a circular region whose origin belongs to $\{A_s - \{A\}\}$ and radius is r_{min}, it should be removed.

Similarly, the filtering condition is $\forall$ anchor $i \in A_s$, $d_{si} > r_{min}$, when the target can't be detected by any anchor.

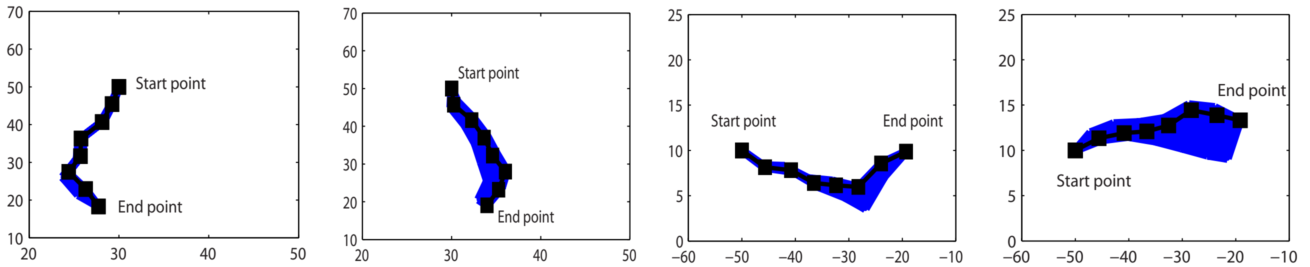
C. Step 2: Backward Tracking

If the target is uniquely localized at end point P_e after an interval, backward tracking is performed. It starts from point P_e, and applies the methods used in forward tracking to estimate the historical target positions. Similarly, we can also obtain a series of possible target positions for each emitting during the interval between t_s and t_e. Based on the same simulation parameters used in the Fig. 8, the corresponding backward tracking results are displayed in Fig. 9.

In practice, the target often can't be localized immediately when it just enters the monitoring area, due to the sparse anchor deployment in the boundary region. In this case, we can estimate the previous target locations by the backward tracking, once it is uniquely localized.

D. Step 3: Bubble Formation

For each emitting of a target, we can obtain two groups of possible locations by forward and backward tracking. The final possible target positions should be the common intersection of two groups. As shown in Fig. 10, the obtained final bubble regions for four cases are all deflated significantly. And these bubble regions indeed indicate the target moving trend. Thus, we have the second interesting observation, which can help us to decay the uncertainty of target locations further.



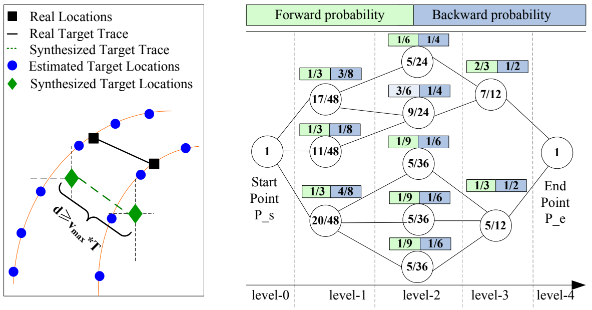
(a) Common intersection for the first case (b) Common intersection for the second case (c) Common intersection for the third case (d) Common intersection for the fourth case

Fig. 10. Illustrations for bubble formation.

Observation 2. *Through performing the above three steps, the region which indicates the possible target locations shrinks quickly and shapes a bubble when the target arrives at the end point. The positions in the bubble region which are close to P_s and P_e have higher certainty than others.*

Using the bi-directional bounding algorithm, we can get a bubble-shaped region that indicates the possible locations of the target. In next step, we are interested in constructing an approximate trace by finding a maximum-likelihood path in a graph, which will be described in the next subsection.

E. ***Step 4:*** *Trace Identification*



(a) Examples for location estimation (b) The computation of arrival probability

Fig. 11. The points synthesis and arrival probability computation.

Given a group of estimated target locations at time t, the straightforward approach to determine the target position is using the following formulas:

$$\begin{cases} X_t = \frac{1}{n}\sum_{i=1}^{n} x_t(i) \\ Y_t = \frac{1}{n}\sum_{i=1}^{n} y_t(i) \end{cases} \tag{3}$$

where n is the number of the estimated position points at time t and $(x_t(i), y_t(i))$ represents coordinates of the ith point along a hyperbola branch. Obviously this method leads to low accuracy. For example, if the target is detected by two anchors, it should lie on the hyperbola. But the synthesized point may be far from it, as shown in Fig. 11(a). Second, the distance between two adjacent synthesized points may be larger than $v_{max} \cdot T$, which is inconsistent with the maximum target speed assumption. Thus, instead of estimating the position separately, we estimate the target trace by finding a maximum-likelihood path in a graph, which is described in the following.

A graph G can be built based on the estimated positions between P_s and P_e, as shown in Fig. 11(b). The vertex set $V(G)$ is composed of all estimated positions, and the edge set $E(G)$ denotes the possible target movements. The length of any edge in $E(G)$ is smaller than $v_{max} \cdot T$. A point is said to be in the level-l point set, if it is l steps away from P_s and we assume there are m levels in total. The target trace is determined by finding a maximum-likelihood path in the graph, which is implemented as follows: we first compute the arrival probability of each vertex, and then define each vertex's weight by its tracking error. Finally we find a path from P_s to P_e in the graph G, which has a minimum accumulated error.

1) Arrival Probability Computation: We compute the arrival probability $p_l(i)$ for each point i in the level-l set. The computation of forward arrival probability starts from the level-0 point P_s and spreads throughout the graph step by step. Fig. 11(b) shows an example of the arrival probability computation process. Initially, the point P_s is uniquely localized and its probability is 100%. Since the target only reaches three possible locations at next level, the forward probability of each point in level-1 is calculated as $100\% * \frac{1}{3}$. Similarly, the forward probability of each point in level-2 is calculated based on its adjacent vertices' forward probability in level-1. In short, given the forward probability $p_l(i)$ and the outdegree $c_l(i)$ for any point i in level-l, we calculate the forward probability of point j in level-$(l+1)$ by the following recursive formula:

$$p_{l+1}(j) = \sum_{q=1}^{k} p_l(v_q) \cdot \frac{1}{c_l(v_q)} \tag{4}$$

where the number of adjacent vertices in level-l for point j is k and point v_q denotes the adjacent vertex. The forward probability of each point can be obtained from its adjacent vertices' probability in last level recursively, with initial known location. Similarly, as shown in Fig. 11(b), we calculate the backward probability of each point, which spreads from end point P_e to start point P_s using the above method. For each vertex in the graph, we take the average of its forward probability and backward probability as the final arrival probability .

2) Vertex Weight Definition: In the step, we calculate the error weight of each vertex in the graph. Refer to the weighted least squares principle, the weight of vertex can be defined as:

$$W_l(i) = \sum_{j=1, j\neq i}^{n} p_l(j) h(x_l(i), y_l(i); x_l(j), y_l(j))^2 \tag{5}$$

where $W_l(i)$ represents the error weight of point i in level-l, $h(x_l(i), y_l(i); x_l(j), y_l(j))$ denotes the distance between points i and j, and n is the number of points in level-l.

3) Optimal Path Matching: The optimal target trace with overall maximum likelihood in the graph can be obtained by choosing a point $f(l)$ from each level, which minimizes the accumulated vertex weight. Namely,

$$\begin{aligned} \text{minimize} \quad & \sum_{l=1}^{m} W_l(f(l)) \\ \text{subject to} \quad & f(l) \in V(G), \\ & \forall l, edge(f(l), f(l+1)) \in E(G) \end{aligned} \tag{6}$$

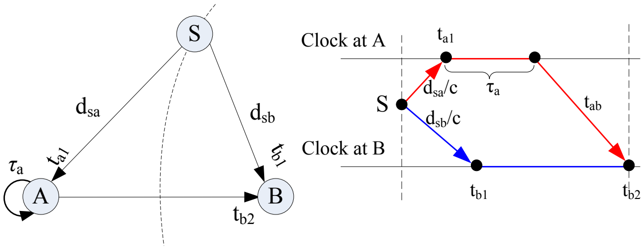
Let the accumulated vertex weight of a path in G be the length of the path, the problem turns into a shortest path problem, which can be solved by Dijkstra algorithm [24].

Here are two points to be noted. First, before the target is uniquely localized again, we only compute the forward probability and find a shortest path from the start point to current estimated positions for providing a real-time trace. Second, the target might not be localized immediately when it enters the boundary region of the monitoring area. In this case, we compute the backward probability and offer a historical trace until the target is uniquely localized.

VI. Discussion

A. Hyperbola Determination without Time Synchronization

In the above design, time synchronization is needed among anchors for constructing the hyperbolic function. Here we describe an improved solution, which can transform the arrival time measurements into hyperbolic functions without time synchronization among anchors.



(a) Hyperbola determination

(b) Time lines of anchors A and B

Fig. 12. Hyperbola determination and time lines.

As shown in Fig. 12(a), given the locations of anchors A and B, we are going to estimate the location (x, y) of target S that emits a signal every T seconds. Let d_{ab} be the distance between anchors A and B. Let d_{sa}, d_{sb} be the unknown distances from S to A and B, respectively. Fig. 12(b) illustrates the hyperbola function generation scheme: when the target emits a signal, anchors A and B both record the signal at their local time t_{a1} and t_{b1}. Then the anchor A send a signal to the anchor B after time τ_a. This signal will reach anchor B at its local time t_{b2}. Through the analysis of timelines of anchors A and B, we can obtain the following equation:

$$\frac{d_{sa}}{c} + \tau_a + t_{ab} = \frac{d_{sb}}{c} + t_{b2} - t_{b1} \tag{7}$$

Let $\Delta t = t_{b2} - t_{b1} - \tau_a - t_{ab}$, which gives

$$\Delta d = d_{sa} - d_{sb} = (t_{b2} - t_{b1} - \tau_a - t_{ab}) \cdot c = \Delta t \cdot c \tag{8}$$

The communication between anchors A and B can be wired, wireless or any other form. And the transmission time t_{ab} from anchor A to B can be easily solved or profiled, since the locations of A and B are known and fixed. Then a hyperbolic function can be obtained from Eq.8, whose foci are A and B. Note τ_a is measured based on anchor A's local timer in this scheme, while t_{b2} and t_{b1} are based on anchor B's local timer. All anchors measure the arrival time of signals and schedule their transmissions using a local clock. Therefore, it doesn't require time synchronization between the target and anchors. Also the time synchronization is not needed among anchors.

B. Factors Affecting Tracking Performance

In our design anchors are required to detect signal arrival time accurately. Unlike the hyperbola curve assumed in previous design, it is an uncertain hyperbolic area in practice. The farther away along the asymptotes, the larger absolute uncertainty in position. However, the time measurement noises don't lead to serious performance degradation, because the uncertainty can be bounded by the detection range of the anchors and the trace is estimated in an uncertain area instead of estimating the position separately. One interesting observation we have for bubble trace is that the actual target speed can affect the shape of the bubble region. If we significantly over-estimate the maximum speed, the bubble region becomes larger, leading to less accurate tracking performance. Our evaluation has confirm this observation.

VII. Evaluation and Experiment

A. A Whole Simulation Example

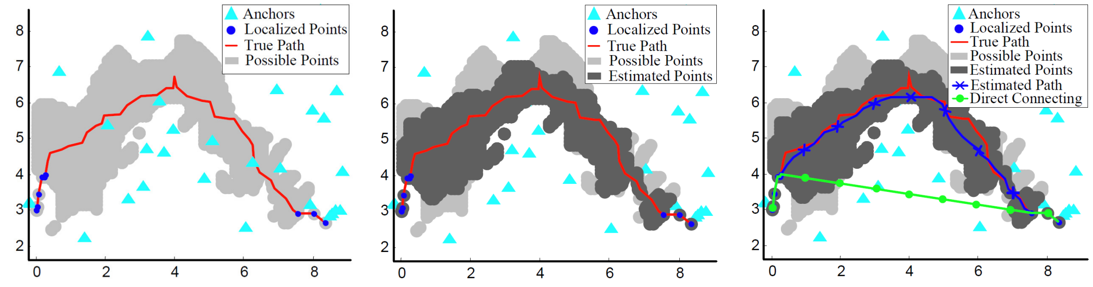
This subsection gives an intuitive tracking process through a simulation example. Detailed performance analysis is provided in later subsections. Fig. 13(a) displays all possible position points estimated in the forward tracking process and Fig. 13(b) shows the final positions after bi-directional bounding. Two traces are given in Fig. 13(c), which are generated by path matching and direct connecting, repetitively. It is clear that the former trace is much more closely around the true one.

B. Simulation Setup

In the simulation, an estimation error at one point in the trace is defined as the geographic offset between the estimated position and corresponding true position. The mean tracking error is defined as average error of points in the trace. We conduct two types of simulations in which the target emitting period is one second. In the first scenario where multiple anchors are deployed randomly and the target may be detected by zero, one, two or more than three anchors during its movement, we compare the average errors for four different methods. In the second scenario, we evaluate the performance when the target can only be detected by two anchors.

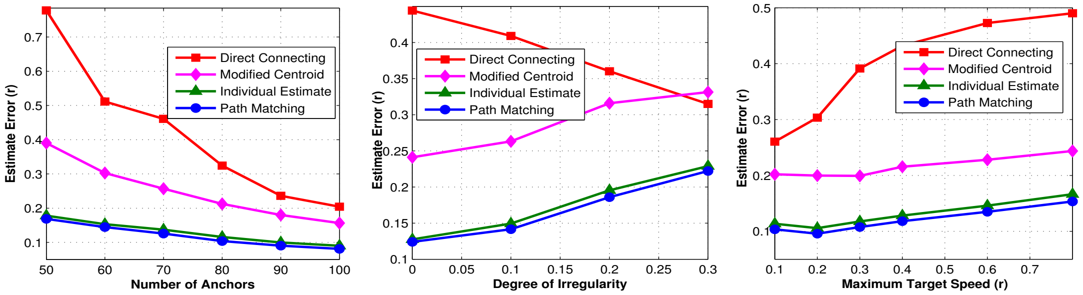
In the first scenario, we vary the following parameters:

- Number of anchors. The number of anchors varies from 50 to 100. And they are randomly distributed in the $10r \times 10r$ rectangular region.



(a) Positions estimated in forward tracking (b) Positions estimated after bounding (c) Optimal path matching

Fig. 13. Visualized simulation example.



(a) Impact of number of anchors (b) Impact of degree of irregularity (c) Impact of maximum speed

Fig. 14. Performance evaluation for multiple anchors.

- Degree of irregularity. We use degree of irregularity (DOI) [25] to denote the signal detection range variation of anchors, which varies from 0 to $0.3r$.
- Speed of target. We adopt the random way point mobility model [26] for the target. It is one of the most commonly used mobility models for mobile ad hoc networks. Here the target's speed is randomly chosen from [0, v_{max}].

We compare the performances for the following methods:

- Direct connecting. The target trace is given by directly connecting points which are localized uniquely.
- Modified centroid. At each step we can obtain a set of possible positions based on the maximum speed condition and previous locations. We simply calculate the centroid for each step and connect them to form a trace.
- Individual estimate. After bi-directional bounding, we estimate each position separately by Eq.3.
- Path matching. After the bi-directional bounding, the trace is obtained by finding a maximum-likelihood path.

In second scenario, we mainly study the impact of time measurement error, hyperbola segment dividing and actual target speed. In this simulation, two anchors are deployed at $(-30m, 0)$ and $(30m, 0)$, respectively. The start point and end point of target are assumed to be known and the maximum speed of the target is specified as $5m/s$.

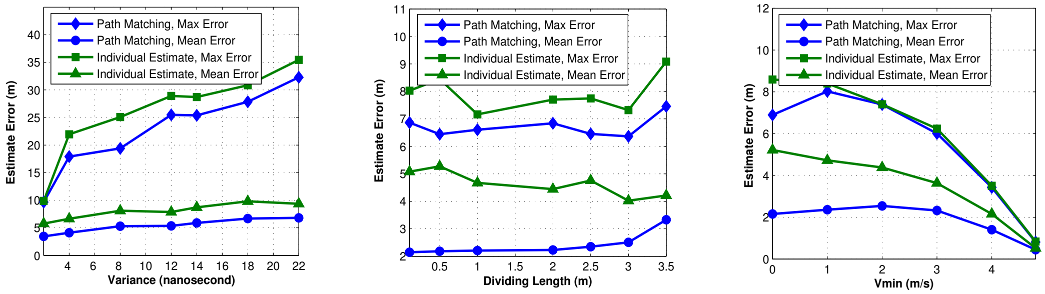
C. Accuracy

1) Impact of the Number of Anchors: Increasing the number of anchors makes localization easier, but increases network and deployment costs. Fig. 14(a) shows the average error for different tracking algorithms when the number of anchors varies. The accuracy of both direct connecting and modified centroid improves as the anchor density increases, since the target can be detected by more anchors. Path matching algorithm performs adequately even for low anchor densities and outperforms the other techniques. With 50 anchors, the estimate error is $0.77r$ for the direct connecting method, while the error in the path matching method is reduced to $0.18r$.

2) Impact of Irregularity: Unlike the perfect circles, the reception range of anchors can vary substantially with environmental conditions and antenna irregularities. Fig. 14(b) shows the impact of DOI on estimate error when 80 anchors are deployed. The average estimate error of the direct connecting decreases as DOI increases. This is because the increasing of DOI enlarges the detection radius of anchors at a certain extent. Hence the target can be detected by more anchors with greater chance and then localized uniquely. For other methods, the number of estimated possible points will increase drastically with the increasing of detection radius, leading to the performance degradation.

3) Impact of Maximum Target Speed: Varying target speed is similar to varying the emitting periods. If transmissions are more frequent, the tracking is more accurate but computation overhead increases. Fig. 14(c) shows the impact of target speed on the error as the maximum speed increases from $0.1r$ to $0.8r$ with 80 deployed anchors. The increased maximum speed makes the tracking less accurate. That's because the next possible locations fall into a larger uncertain region.

4) Impact of Time Detection Error: The impact of TDOA measurement errors for multiple anchors was extensively studied, we evaluate their impact with two deployed anchors. The errors of time measurement Δt are assumed to be normally distributed according to $N(0, \delta^2)$. Fig. 15(a) shows that the maximum position error depends on the variance of measurements. However, the variance doesn't have a great influence on mean tracking errors for path matching algorithm.



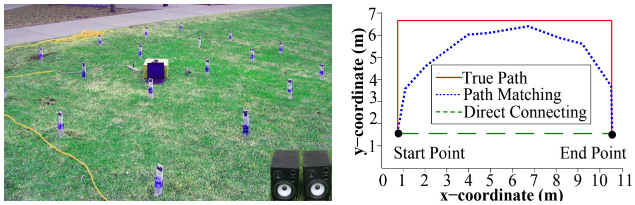
(a) Impact of time detection error (b) Impact of segment dividing length (c) Impact of minimum target speed

Fig. 15. Performance evaluation for two anchors.

5) Impact of Dividing Hyperbola Segment: In the tracking with two anchors, the hyperbola segment is divided for obtaining the sample points. More division will increase the computational complexity. However, it doesn't bring significant performance improvement. As shown in Fig. 15(b), the tracking accuracy is affected slightly by the dividing length. In this simulation, the hyperbola segment can be divided into several fragments with the length of 2 m, which has little accuracy loss and low computation complexity.

6) Impact of Minimum Target Speed: The tracking performance is considerably affected by the minimum target speed. In this simulation, we vary the minimum target speed from $0m/s$ to $4.8m/s$, and the actual target speed is randomly chosen from $[v_{min}, v_{max}]$. As shown in Fig. 15(c), with the increasing of the minimum speed, the tracking accuracy becomes more and more precise. And the estimate errors drop fast as the minimum target speed increases from $3m/s$.

D. Experiment



(a) System evaluation (b) Tracking results

Fig. 16. A prototype system.

A simple prototype system of the design is implemented. A speaker which generates 4.7Khz acoustic signals is used as the target. And 20 MicaZ nodes which are equipped with microphones are deployed in (4×5) grids. In order to make the speaker and the nodes lie in the same horizontal plane, each node is left 8 inches above the ground as shown in Fig. 16(a). In the prototype system, the locations where the speaker emits signals form a "⊓" shape trace, in which the start point and end point are known. And the MicaZ nodes are responsible for detecting the speaker's signal and recording the arrival timestamps. During the speaker's movement, it may be detected by more than two nodes. In order to verify the effectiveness of the proposed design, we only use the timestamps measured from two nodes which are close to the speaker. After collecting all the arrival time data, we process them off-line. Fig. 16(b) illustrates the results where the dotted line is our estimated trace and the long dashed line denotes the trace generated by direct connecting method.

VIII. Related Work

A. Location Determination Approaches

Existing localization schemes are mainly categorized into two classes: range-free and range based. Range-free solutions like APIT [25], Centroid method [27], DV-HOP [28] don't need to measure accurate distances or angles. Instead, seeds or controlled-event distributions are used for node localization. In order to remove the impact of holes in the network, a Rendered Path (REP) technique is proposed for anisotropic sensor networks in [29]. Range-based approaches like Received Signal Strength (RSS) [30], Time of Arrival (TOA) [31], Angle of Arrival (AOA) [3] assume that nodes can measure the distance or the relative directions of neighbor nodes. In addition, there is another localization method also named "TDOA" [32], using ultrasound and Radio-frequency (RF) signals simultaneously on the sender. When a receiver records the signal arrival time, the difference of two timestamps can be used to calculate the sender-receiver distance.

B. Target Tracking Techniques

Many techniques have been proposed for target tracking so far [33] [34] [35] [36] [37] [38]. In most tracking approaches the target is often localized based on trilateration, which needs dense infrastructure support and incurs heavy cost. They will suffer from severe performance degradation with the decreasing number of seeds. The dead reckoning approach estimates the current location by adding estimated displacement to the previous location estimation [39]. Although it is less dependent on the infrastructure, specialized hardware on target like accelerometer and digital compass are required, which limits the practical aspects. Different from previous work, we provide a tracking framework for an asynchronous network with insufficient anchor deployments which can provide fair tracking accuracy.

Another key idea in tracking mobile targets is to include a dynamic model for predicting the position at the next time step. The widely used filtering techniques are Bayesian networks [40] and Kalman filter [41]. However, the model based methods not only bring about a complex system design, but also require some maneuver-related assumptions about the mobile target. In contrast, we impose the maximum target speed as the only constraint in this work.

IX. Conclusions

This paper presents the work for target tracking under sparse anchor deployment, which can improve the tracking accuracy compared to the traditional TDOA method. Without assumptions of time synchronization among anchors, we first determine the possible target region by the bi-directional bounding algorithm and then find a maximum-likelihood path from a graph. Our simulations and experiment reveal that the proposed technique can provide fairly accuracy, especially in the sparse anchor deployment environment.

Nevertheless, many issues still remain to be explored. Our ongoing work are: (1) study the optimal anchor placement problem, since the error is related to the anchor placement; (2) integrate the bubble generation step with the formation of particle filters to provide a predictable performance; (3) extend the proposed algorithm with multiple targets tracking, and see if we can find some new methods for joint targets tracking.

Acknowledgment

This work was supported in part by NSF grants CNS-0917097, CNS-0845994, and CNS-0626609. We also received partial support from the National Natural Science Foundation of China grant (60873248) and the Key Program of National Natural Science Foundation of China grant (60933011).

References

[1] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, vol. 2, pp. 775-784, Mar. 2000.

[2] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "TPS: A time-based positioning scheme for outdoor wireless sensor networks," in *Proc. IEEE INFOCOM*, vol. 4, pp. 2685-2696, Mar. 2004.

[3] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," in *Proc. ACM MobiCom*, pp. 166-179, Jul. 2001.

[4] P. Zhang and M. Martonosi, "LOCALE: Collaborative localization estimation for sparse mobile sensor networks," in *Proc. IPSN*, pp. 195-206, Apr. 2008.

[5] B. Kusy, A. Ledeczi, and X. Koutsoukos, "Tracking mobile nodes using RF doppler shifts," in *Proc. ACM SenSys*, pp. 29-42, Nov. 2007.

[6] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *J. on Applied Signal Processing*, vol. 2003, no. 4, pp. 378-391, Mar. 2003.

[7] C. Randell, C. Djiallis, and H. Muller, "Personal position measurement using dead reckoning," in *Proc. IEEE ISWC*, pp. 166-173, Oct. 2005.

[8] H.-J. Jang, J. W. Kim, and D.-H. Hwang, "Robust step detection method for pedestrian navigation systems," *IEEE Electronics Letters*, vol. 43, no. 14, pp. 749-751, Jul. 2007.

[9] R. Jirawimut, P. Ptasinski, V. Garaj, F. Cecelja, and W. Balachandran, "A method for dead reckoning parameter correction in pedestrian navigation system," *IEEE Trans. Instrumentation and Measurement*, vol. 52, no. 1, pp. 209-215, Feb. 2003.

[10] G. Mellen, M. Pachter, and J. Raquet, "Closed-form solution for determining emitter location using time difference of arrival measurements," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 1056-1058, Jul. 2003.

[11] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. on Signal Processing*, vol. 42, no. 3, pp. 1905-1915, 1994.

[12] M. Cardei and J. Wu, *Handbook of Sensor Networks, chapter Coverage in Wireless Sensor Networks*. CRC Press, 2004.

[13] S. Kumar, T. H. Lai, and J. Balogh, "On K-coverage in a mostly sleeping sensor network," in *Proc. ACM MobiCom*, pp. 144-158, Sep. 2004.

[14] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. OSDI*, pp. 147-163, Dec. 2002.

[15] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. ACM SenSys*, pp. 138-149, Nov. 2003.

[16] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proc. ACM SenSys*, pp. 39-49, Nov. 2004.

[17] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," in *Proc. IPSN*, pp. 37-48, Apr. 2009.

[18] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. IEEE INFOCOM*, vol. 3, pp. 1380-1387, Apr. 2001.

[19] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proc. ACM Mobihoc*, pp. 131-142, May 2006.

[20] X. Bai, Z. Yun, D. Xuan, T. H. Lai, and W. Jia, "Deploying four-connectivity and full-coverage wireless sensor networks," in *Proc. IEEE INFOCOM* , pp. 906-914, Apr. 2008.

[21] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proc. ACM MobiCom*, pp. 284-298, Aug. 2005.

[22] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," in *Proc. ACM MobiCom*, pp. 45-57, Sep. 2004.

[23] X. Xu, S. Sahni, and N. S. V. Rao, "On basic properties of localization using distance-difference measurements," in *Proc. FUSION*, Jul. 2008.

[24] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[25] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. ACM MobiCom*, pp. 81-95, Sep. 2003.

[26] D. B. Johnson and D. A. Maltz, "Dynamic source routing in Ad hoc wireless networks," *Mobile Computing*, pp. 153-181, 1996.

[27] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28-34. Oct. 2000.

[28] D. Niculescu and B. Nath, "DV based positioning in Ad Hoc networks," *Journal of Telecommunication Systems*, vol. 22, no. 1-4, pp. 267-280, 2003.

[29] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," in *Proc. ACM MobiCom*, pp. 51-62, Sep. 2007.

[30] J. Hightower, G. Borriello, and R. Want, "SpotON: An indoor 3D localization sensing technology based on RF signal strength," *Technical Report #2000-02-02, University of Washington*, Feb. 18, 2000.

[31] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, "Global Positioning System: Theory and Practice," Fourth Edition, Springer Wien, New York, 1997.

[32] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. ACM MobiCom*, pp. 32-43, Aug. 2000.

[33] 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, pp. 17-29, Mar. 2002.

[34] B. Kusy, J. Sallai, G. Balogh, A. Ledeczi, V. Protopopescu, J. Tolliver, F. Denap, and M. Parang, "Radio interferometric tracking of mobile wireless nodes," in *Proc. MobiSys*, pp. 139-151, Jun. 2007.

[35] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," *IEEE Transactions on Wireless Communication*, vol. 3, no. 5, pp. 1689 - 1701, Sep. 2004.

[36] A. Smith, H. Balakrishnan, M. Goraczkoet, and N. Priyantha, "Tracking moving devices with the cricket location system," in *Proc. MobiSys*, pp. 190-202, Jun. 2004.

[37] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "PinPoint: An asynchronous time-based location determination system," in *Proc. ACM MobiSys*, pp. 165-176, Jun. 2006.

[38] Z. Zhong, T. Zhu, D. Wang, and T. He, "Tracking with unreliable node sequences," in *Proc. IEEE INFOCOM*, pp. 1215-1223, Apr. 2009.

[39] L. Fang, P. J. Antsaklis, L. A. Montestruque, M. B. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie, "Design of a wireless assisted pedestrian dead reckoning system -The NavMote experience," *IEEE Trans. Instrumentation and Measurement*, vol. 54, no. 6, pp. 2342-2358, Dec. 2005.

[40] S. Mohanty, "VEPSD: A novel velocity estimation algorithm for next-generation wireless systems," *IEEE Trans. on Wireless Com.*, vol. 4, no. 6, pp. 2655-2660, Nov. 2005.

[41] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. D, pp. 35-45, Mar. 1960.