Tracking Multiple Targets Using Blind Source Separation Algorithms

Ye Zhu and Anil Vikram
Electrical and Computer Engineering Department
Cleveland State University
Tech Report CSU-ECE-TR-08-02
May 23, 2008

Abstract

This paper describes an approach to track multiple targets using wireless sensor networks. In most of previously proposed approaches, tracking algorithms have access to the signal from *individual* target for tracking by assuming (a) there is only one target in a field, (b) signals from different targets can be differentiated, or (c) interference caused by signals from other targets is negligible because of attenuation. We propose a general tracking approach based on *blind source separation*, a statistical signal processing technique widely used to to recover individual signals from mixtures of signals. By applying blind source separation algorithms to *mixture* signals collected from sensors, signals from *individual* targets can be recovered. By correlating *individual* signals recovered from different sensors, the proposed approach can estimate tracks of multiple targets. To improve tracking performance, our approach utilizes both temporal information and spatial information available for tracking. We evaluate the proposed approach through extensive experiments. Experiment results show that the proposed approach can track multiple objects both accurately and precisely.

1 Introduction

Tracking moving targets with wireless sensors is one of prominent applications of wireless sensor networks. Sensors, also called as “smart dust” [1], are small devices known for their simplicity and low cost. Using a network of sensors with wireless communication capability enables both cost-effective and performance-effective approaches to track targets because of the availability of large amount of data collected by sensors for tracking targets. Depending on the applications, sensors with different sensing modalities such as acoustic, seismic, infrared, radio, and magnetic can be deployed for tracking different type of targets.

In general, data collected by sensors is *aggregate* data. In the signal processing language, signals received by sensors are generally *mixtures* of signals from *individual* targets. For example, an acoustic sensor in a field of interest may receive sound signals from more than one targets. Obviously tracking targets based on mixture signals can result in inaccurate results when interference from targets other than the one of interest is not negligible. Without loss of generality, we use the term *aggregate signal* to mean the signal received by sensor, i.e., data collected by sensors and *individual signal* to mean the signal transmitted from or caused by individual targets in the rest of the paper.

Aggregate signals collected by sensors networks pose a big challenge to target-tracking solutions. The problem space of the target-tracking problem is divided and special cases of the target-tracking problem have been well studied:

- Single-target case: In this case, it is assumed that only one target exists in a field of interest. So signals received by sensors are essentially *individual signals*.
- Negligible interference case: Some researches assume that interference from targets other than the one of interest is negligible. The assumption is legitimate for applications in which signal from a target attenuates dramatically when distance between the target and sensor increases.
- Distinguishable target case: Sensors can distinguish targets by tags embedded in signals or by having different targets to send signals using different channels such as using different frequency bands.

All these special cases assume that tracking algorithms can have access to individual signals. Singh et al. [2] propose a general approach to track multiple targets indistinguishable by sensors. The approach is based on binary proximity sensors which can only report whether or not there are targets in sensing area. The approach is based on simple device, the binary proximity sensors with the cost of limitation that it is only applicable to to track targets in smooth paths [2].

We propose an approach based on blind source separation, a methodology from statistical signal processing to recover unobserved ”source” signals from a set of observed mixtures of the signals. Blind source separation model was originally defined to solve *cocktail party problem*: The blind source separation algorithms can extract one persons voice signal given the mixtures of voices in a cocktail party. Blind source separation algorithms solve the problem based on the independence between voices from different persons. Similarly, in the target-tracking problem, it is generally safe to assume *individual signals* from different targets are independent. So we can use blind source separation algorithms to recover *individual signals* from *aggregate signals* collected by sensors. For the cases in which *individual signals* are dependent, blind source separation algorithms based on timing structure [3–5] of *individual signals* can be used.

The proposed algorithm utilizes both temporal information and spatial information available to track targets. Applying blind source separation algorithms on aggregate signals collected by sensors can recover *individual signals*. But the output of blind source separation algorithms includes not only recovered individual signals, but also noise signals, aggregate signals and partial signals which contain individual signals in different time durations. Clustering is used in our algorithm to pick out the individual signals from signals output by the blind source separation algorithms. A voting step based on spatial information is used to further improve the performance of the algorithm.

The contributions of this paper can be summarized as follows:

- We proposed a general approach to track multiple targets in a field. The approach can be applied in real-world applications where targets are indistinguishable and interference from other targets other than the one of interest is not negligible.
- We evaluate our approach with extensive experiments and analyze the effect of parameters used in the proposed approach experimentally and theoretically.
- We propose metrics to evaluate performance of target-tracking algorithms. The metrics originate from the general metrics used to evaluate performance of an estimator in statistics since essentially target tracking algorithms *estimate* the paths based on data collected from sensor networks.
- According to our knowledge, we are the first to apply blind source separation to process data collected from wireless sensor networks. Blind source separation algorithms are useful tools for processing data collected from wireless sensor networks since essentially data collected from sensors are all *aggregate* data. In this paper we focus on applying blind source separation in the target-tracking problem. The blind source separation algorithms can also be used to process data in other applications of wireless sensor networks such as location detection and factor analysis. For most applications of wireless sensor networks, analysis based on *individual* signals can yield more accurate results.

The rest of the paper is organized as follows. Section 3 explains about our network model and assumptions. In Section 5, we describe our approach in details. Section 6 theoretically analyzed the performance of our approach and effect of parameters used in our approaches. we report performance evaluation results of our approaches in Section 7. We conclude our paper in Section 9.

2 Related work

Wireless sensor networks have been proposed or deployed to track targets in various applications. The examples are tracking robots with infrared signal [6], tracking vehicles with infrared signals [7], tracking ground moving targets with seismic signals [8], tracking moving vehicles with acoustic sensors [9], tracking people with RF signals [10]. Location detection, equivalent as tracking static targets, have also been studied extensively. The topic has been investigated in different wireless networks such as wireless sensor networks [11, 12], wireless LANs [13], and wireless ad-hoc networks [14, 15].

Most proposed approaches to track targets and detect location are based on characteristics of physical signals such as angle of arrival (AOA) [16–18], Time of Arrival (TOA) [19, 20], Time Difference of Arrival (TDOA) [21, 22] and Received Signal Strength (RSS) [23, 24]. Receiver signal strength is widely used in tracking targets with wireless sensor networks [25, 26]. Lots of work focuses on tracking a single target [27–29] or assume targets are distinguishable [30]. Tracking multiple targets in a field is a challenging problem in comparison with tracking single target. Various advanced techniques have been applied to solve the problem. The examples are signal processing techniques such as wavelet [31], FFT [7, 32], Kalmn filter [33, 34], statistical techniques such as principle component analysis [35].

A string of researches on tracking targets with wireless sensor networks are based on binary proximity sensors which can only report whether there are targets within sensing area. Initial work [27–29] on binary proximity sensors focus on tracking single target. Singh et al. [2] extended the approach to track multiple indistinguishable targets by applying particle filtering algorithms. Approaches based on binary proximity sensors have two obvious advantages: (a) The sensors are very simple since they only report binary information. (b) The approaches are robust since interference from other targets are essentially filtered out by an equivalent low-passed filter [29]. The cost of using these simple devices is loss of information which is helpful to accurately track targets due to the filtering effect. So approaches based on binary proximity sensors can not track target in path with high-frequency variations [29]. We propose a general approach to track multiple indistinguishable targets. The approach is based on blind source separation algorithms which can recover individual signals from aggregate signals. So the challenging problem of tracking multiple targets become a much easier problem equivalent as tracking single target. Since individual signals can be fully recovered, our approach can track targets following paths with high-frequency variations.

3 Network Model and Assumptions

A general model of tracking targets using wireless sensor networks is shown in Figure 1. Wireless sensors are randomly deployed in the field of interest. Generally a wireless sensor can receive individual signals from multiple sources. For example, suppose acoustic sensors are deployed in Figure 1, Sensor O_1 can receive audio signals from Target s_1, s_2, and s_3 during one time duration. Following are the assump-

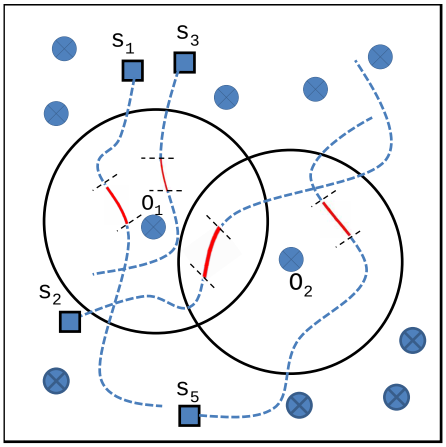


Figure 1. Network Model

tions made in this general model:

- Sensors have no capability to distinguish targets. This assumption is important for deploying sensors in uncooperative or hostile environment as tracking enemy soldiers with wireless sensor networks.
- The location of each sensor in the sensor network is known. Location information can be gathered in a variety of ways. For example, the sensors may be planted, and their location marked. Alternatively, sensors may have GPS capabilities. Finally, sensors may locate themselves through one of severals schemes that rely on sparsely located anchor sensor nodes [36].
- Aggregate signals collected by wireless sensors can be gathered for processing by a sink or gateway. Data compression or coding schemes designed for sensor networks such as ESPIHT [37, 38] can be used to reduce the data volume that is caused by remaining spatial redundancy across neighboring nodes or temporal redundancy at individual nodes.
- Targets are moving under a speed limit. Obviously it is impossible to track a high-speed target which only generates a small amount of data when passing the field of interest. We analyze the speed limit in Section 6.

4 Application of Blind Source Separation Algorithms in Tracking Target

In this section, we introduce blind source separation and rationale of applying blind source separation to the multiple target tracking problem using wireless sensor networks.

4.1 Blind Source Separation

Blind Source Separation (BSS) is a methodology in statistical signal processing to recover unobserved "source" signals from a set of observed mixtures of the signals. The separation is called *blind* to emphasize that the source signals are not observed and that the mixture is a black box to the observer. While no knowledge is available about the mixture, in many cases it can be safely assumed that source signals are independent. In its simplest form [39] , the blind source separation model assumes n independent signals $S_1(t)$, $\cdots$, $S_n(t)$ and n observations of mixture $O_1(t)$, $\cdots$, $O_n(t)$ where $O_i(t) = \sum_{j=1}^{n} a_{ij} S_j(t)$. The goal of BSS is to reconstruct the source signals $S_j(t)$ using only the observed data $O_i(t)$, the assumption of independence among the signals $S_j(t)$. Given the observations $O_i(t)$, BSS techniques estimate the signals $S_j(t)$ by maximizing the independence between the estimated signals. A very nice introduction to the statistical principles behind BSS is given in [39]. The common methods employed in blind source separation are minimization of mutual information [40, 41], maximization of nongaussianity [42, 43], maximization of likelihood [44, 45]. Timing-structure based algorithms [3–5] can be used to recover source signals when source signals are dependent.

4.2 Recover *Individual Signals* for Target-Tracking with Blind Source Separation Algorithms

In our tracking approach, blind source separation algorithms are used to recover *individual signals*, i.e., source signals as in Section 4.1 from *aggregate signals*, i.e., observations as in Section 4.1. Suppose acoustic sensors are deployed in the field shown in Figure 1, Sensor O_1 can receive audio signals from targets S_1, S_2, and S_3 and Sensor O_2 can receive audio signals from targets s_2 and s_5. If we represent the signal received by Sensor O_i as $O_i(t)$ and the signal from Target S_i as $S_i(t)$, we can have following two equations: $O_1(t) = S_1(t) + S_2(t) + S_3(t)$, $O_2(t) = S_2(t) + S_5(t)$. In general, for m neighboring sensors and n targets, we can rewrite the problem in vector-matrix notation,

$$\begin{pmatrix} O_1(t) \\ O_2(t) \\ \vdots \\ O_m(t) \end{pmatrix} = \mathbf{A}_{m\times n} \begin{pmatrix} S_1(t) \\ S_2(t) \\ \vdots \\ S_n(t) \end{pmatrix} \tag{1}$$

where $\mathbf{A}_{m\times n}$ is called *mixing matrix* in the BSS literature. Since the individual signals are independent from each other - they come from different targets - we can use any of the algorithms mentioned in Section 4.1 to recover individual signals $S_1(t), \cdots, S_n(t)$.

While the goal of BSS in this context is to re-construct the original signals $S_i(t)$, in practice the separated signals are sometimes only loosely related to the original signals. We categorize these separated signals into four types: In the first case, the separated signal is correlated to actual individual signals $S_i(t)$. The separated signal in this case may have a different sign than the original signal. We call this type of separated signal as individual separated signal. In the second case a separated signal may be correlated to an aggregate of signals from several targets. This happens when signals from more than two targets can be "heard" by all the sensors. In such a case, the BSS algorithm would not be able to fully separate the signal mixture into the individual separated signals. Rather, while some individual signals can be successfully separated, others remain aggregated. In the third case, separated signals may be correlated to one original signal in the beginning part and correlated to another original signal in the rest. We call this type of separated signal as partial separated signal. This happens when a target moves out of one sensing range and enter into another sensing range. In the fourth case, separated signals may represent noise signals. Noise in our case can be caused by attenuation because of the distance between sensor and targets. When a target is moving, the attenuation will change depending on the distance between the target and the sensor of interest. So two neighboring sensors may receive different individual signals from one target. The difference can be separated out as noise separated signals.

5 Blind Source Separation Algorithm

5.1 Tracking Algorithm

The tracking algorithms consists of six steps: In the first step, aggregate signals collected from sensors are grouped and segmented and these segmented signals are fed to the second step, blind source separation step. The output of blind source separation step are the separated signals. As described in Section 4, these separated signals contain individual separated signals, aggregate separated signals, noise separated signals, and partial separated signals. Clustering step will cluster these separated signals. Based on the cluster information, center selection step selects separated signals which are closest to actual individual signals. Intersection step estimates segments of paths based on separated signals selected from the previous step. Voting step outputs the estimated paths by voting on path segments generated in the intersection step. The details of these fives steps (preparation, separation, clustering, center selection, intersection, voting) are described below.

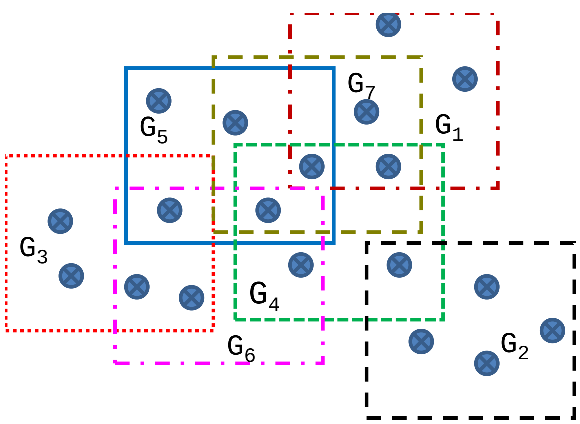


Figure 2. Grouping $(n_{group} = 5)$

5.2 Preparation Step

To fully utilize information collected from wireless sensor networks, aggregate signals are grouped spatially and segmented temporally. As shown in Figure 2, sensors in the field are grouped into sensor groups. Each group has n_{group} neighboring sensors. Aggregate signals collected from each sensor group are segmented according to time slots shown in Figure 3. Time slots are of length l_{seg}. The step size between two successive time slots is l_{step}. So two successive signal segments have a common part of length $l_{seg} - l_{step}$. A BSS algorithm will be applied on grouped aggregate signals sequentially, i.e., segment by segment in the next step.

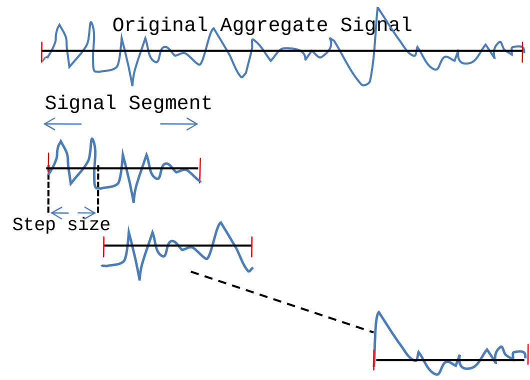


Figure 3. Signal Segments

We represent the segment group from ith sensor block and jth time slot as $OG_{i,j}$. The pth segment in the group is denoted as $O^p_{i,j}$. In set theory notation, $OG_{i,j} = \{O^p_{i,j} : p = 1, \cdots, n_{group}\}$. The output of the preparation step is segment groups $OG_{i,j}$.

Redundancy is created during grouping and segmenting. In grouping, a sensor can be grouped into more than one sensor groups. In segmenting, two successive time slots have overlap. The redundancy will be utilized in the following steps.

Since after the preparation step, signals are all in unit of segments. We use actual segments, individual segments, aggregate segments, partial segments, noise segments to mean segments of original individual signals, individual separated signals, aggregate separated signals, partial separated signals, and noise separated signals respectively.

5.3 Separation Step

The separation step applies a BSS algorithm on segments contained in $OG_{i,j}$. The output of the separation step is separated segments in groups denoted by $SG'_{i,j}$, i.e., the group of segments separated from $OG_{i,j}$.

5.4 Clustering Step

Clustering step is used to eliminate noise segments, aggregate segments, and partial segments. The heuristic behind this step is: If a separated signal represents an individual signal, similar signals will be separated in at least similar forms by more than one neighboring sensor groups. In contrast, a separated signal that was generated because of attenuation or some interference has likely been generated by a single group only.

Based on the heuristic we identify cluster of similar separated signal segments by using the correlation coefficient as measure of similarity, and define the distance between two separated segments as follows:

$$D(S'^p_{i,j}, S'^q_{h,j}) = 1 - ||corr(S'^p_{i,j}, S'^q_{k,j})|| \quad , \tag{2}$$

where $S'^p_{i,j}$ denotes the pth segment in separated segment group $SG'_{i,j}$, and $corr(x,y)$ denotes the correlation coefficient of segments x and y. We use the absolute value of the correlation coefficient because the separated segments may be of different sign than the actual segment. Clustering will only cluster segments of same time slots as indicated in the distance measure defined in Equation 2.

The highly-correlated (similar) segments will cluster together. Figure 4 use a two-dimensional representation to further illustrate the rationale for the clustering approach in this step. As shown in this figure, the individual segments form clusters. The aggregate segments and partial segments on the other hand scatter in-between these clusters. The noise segments are distant both from each other and from the other segments.

In summary, the input of the clustering step is $SG'_{i,j}$ and the output of this step is clusters formed in each time slots. We use $Clst^i_j$ to denote the ith cluster formed in the jth time slot.

5.5 Center Selection Step

The goal of center selection step is to select center segments as shown in Figure 4 from clusters formed in the previous step. Center segments are the segments in the center of each cluster formed according to distance measure defined in Equation 2. The heuristic behind the step is: When a target is moving, neighboring sensor groups along the path can receive common part of the individual signal from the target. But for segments created from noise, the heuristic will not hold. For example, suppose the ith and kth sensor groups are neighboring sensor groups and segments $S'^{p}_{i,j}$ and $S'^{q}_{k,j+1}$ separated from these two sensor groups are related to the same individual signal from a target, the latter part of $S'^{p}_{i,j}$ should be very similar as the beginning part of $S'^{q}_{k,j+1}$. More specific, the partial segment $S'_{i,j}(l_{step}, l_{seg})$ should be highly correlated with the partial segment $S'_{k,j+1}(0, l_{seg} - l_{step})$. (Please note: we use $S'^{p}_{i,j}(x,y)$ to denote the the part of segment $S'^{p}_{i,j}$ from $t = x$ to $t = y$.) The heuristic still holds for segments separated from the same sensor block, i.e., $i = k$. Link correlation[1] $\rho(S'^{p}_{i,j}, S'^{q}_{k,j+1})$ is defined based on the heuristics as follows:

$$\rho_j(S'^{p}_{i,j}, S'^{q}_{k,j+1}) = ||corr(S'^{p}_{i,j}(l_{step}, l_{seg}), S'^{q}_{k,j+1}(0, l_{seg} - l_{step}))|| \quad (3)$$

where $\rho(S'^{p}_{i,j}, S'^{q}_{k,j+1})$ denotes the link correlation between segments $S'^{p}_{i,j}$ and $S'^{q}_{k,j+1}$. Absolute value is used in link correlation definition because the separated segments may be of different sign than the actual segments.

Center segments are selected based on the link correlation: The center segment in $Clst^i_j$ is the segment that has the highest link correlation with one of the center segments selected in $j+1$th time slot. To make center selection more robust against aggregate and partial segments, we use summation of link correlation between n_{slot} consecutive time slots, i.e., $\rho = \sum_{i=j}^{j+n_{slot}} \rho_i$.

The input to the center selection step is $Clst^i_j$ and the output is center segments C^i_j which denotes the ith center segment in the jth time slot. The number of center segments in each time slot is K which is the number of targets in the field. The value of K is either known a priori or can be estimated by using Principle Component Analysis.

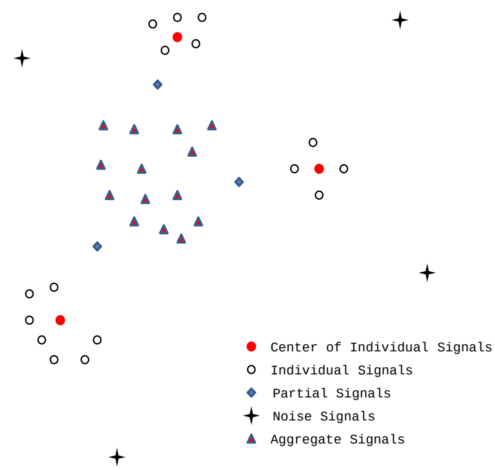


Figure 4. Clustering

5.6 Intersection Step

We estimate a segment of path by intersecting the sensing ranges of sensor groups that are likely to "hear" the target. For this we select sensor groups that have separated segments that are closely correlated with the likely individual segments from the targets in the area. The rationale is that for the sensors in a sensor group to hear a target, they must have sensed a signal that is at least similar to the signal generated by the target. This means that sensor groups with separated segments that correlate with any of the K center segments are likely to hear a target. We therefore determine the likely path segment taken by a target by geographically intersecting the sensing areas of the sensors in those sensor groups that have highly correlated separated segment with a center segment determined in the preceding center selection step.

The input of this step is center segments C^i_j. For each center segment, an intersection area $area^i_j$ is generated as output of this step. These generated areas are estimated path segments.

5.7 Voting Step

The voting step selects best estimated path segments for each time slot and concatenates the best estimated path segments to form an estimated path.

The criteria used in selecting best estimated path segments are based on the distance between path segments, i.e., intersection areas. The distance between two intersection areas is defined as the distance between two closest points from each area. If two areas overlap, their distance is zero. In our algorithm, the best estimated segments in one time slot have the smallest total distance with all the estimated segments in current time slot, previous four time slots, and following four time slots.

6 Theoretical Analysis

In this section, we analyze the effect of signal attenuation, tracking resolution, effect of moving speed, and complexity of the algorithm.

[1]We use the term link correlation because it is used to "link" segments in two *successive* time slots.

input : $Clst_j^i$ - ith cluster formed in jth time slot ($j = 1, \cdots, v$), K - number of center segments in each time slot, v is number of time slots available.
output: C_j^k - kth center segment selected from jth time slot ($j = 1, \cdots, v$)

Initialize $ClstFlag_j^i = 0$ for all i and j;
Initialize $C_j^k = 0$ for all k and j;
for $r \leftarrow 1$ **to** $v - 1$ **do**
  foreach *separated segment* S_a' *in rth time slot* **do**
    foreach *separated segment* S_b' *in* $(r+1)$*th time slot* **do**
      calculate link correlation $\rho_r(S_a', S_b')$ according to Equation 3;
    end
  end
  include $\rho_r(S_a', S_b')$ into the set ρ_r;
end
for $s \leftarrow 1$ **to** K **do**
  for $r \leftarrow 1$ **to** v **do**
    if $r == 1$ **then**
      while $C_r^s == 0$ **do**
        find segments S_a' and S_b' which have the largest link correlation $\rho_r(S_a', S_b')$ in the set ρ_r;
        find $Clst_r^x$ so that $S_a' \in Clst_r^x$; /* S_a' belongs to rth time slot. */
        find $Clst_{r+1}^y$ so that $S_b' \in Clst_{r+1}^y$;
        if $ClstLabel_r^x == 0 \&\& ClstLabel_{r+1}^y == 0$ **then**
          $C_r^s = S_a'$;
          $C_{r+1}^s = S_b'$;
          remove $\rho_r(S_a', S_b')$ from the set ρ_r;
          Set $ClstLabel_r^s = 1$;
        end
        else
          remove $\rho_r(S_a', S_b')$ from the set ρ_r;
          Continue;
        end
      end
    end
    if $r > 1$ **then**
      while $C_r^s == 0$ **do**
        find segments S_m' in the $(r+1)$th time slot which has the largest link correlation with C_r^s in the set ρ_r;
        find $Clst_{r+1}^z$ so that **if** $ClstLabel_{r+1}^z == 0$ **then**
          $C_r^s = S_b'$;
          $C_{r+1}^s = S_m'$;
          remove $\rho_r(C_r^s, S_m')$ from the set ρ_r;
          Set $clulabel_{r+1}^z = 1$;
        end
        else
          remove $\rho_r(C_r^s, S_m')$ from the set ρ_r;
          Continue;
        end
      end
    end
  end
end

Algorithm 1: Select Center Segments from First Time

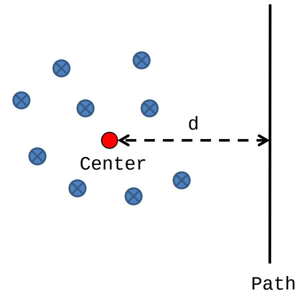


Figure 5. Setup for Experiments on Signal Attenuation

6.1 Signal Attenuation

Signal attenuation is a natural consequence of signal transmission over long distances. It is a function of transmission distance.

When tracking static targets, signal attenuation will not affect tracking performance. Since targets are static, the distance between targets and sensors does not change over time. So the attenuation can be modeled as a constant. For the same individual signal from a target, different sensors will observe different attenuation because of different transmission distance. So individual signals received by different sensors from the same target can be different by a scaling factor. The difference because of a scaling factor can be absorbed by the mixing matrix defined in the BSS model as Equation 1. So attenuation does not affect tracking static targets by our approach.

When tracking moving targets, signal attenuation may cause noise signals in the output of separation step. When targets are moving, the difference between individual signals received by different sensors is not just a scaling factor. Because when a target is moving, the attenuation changes with the transmission distance between the target and a specific sensor. So the difference can not be absorbed by the mixing matrix. The consequences of the difference are: (a) Noise segments can be generated during separation because of the difference (b) Separated individual signals are less correlated with original individual signals. Clustering step, center selection step and voting step are designed with consideration of these consequences.

We did a simple experiment to show the effect of signal attenuation on the separation performance when targets are moving. The setup is as shown in Figure 5: Ten randomly placed sensors form a sensor group. Three targets are moving in the sensing range of the sensor group. We fixed the path of two targets in our experiment and increase the vertical distance between center of the sensor group and the path taken by the target of interest. Figure 6 shows the maximum correlation between separated signals and the actual individual signal from the target of interest. As we can observe that when the vertical distance increases, the correlation with original individual signal is higher. So the separation performance is better when vertical distance increases. The reason is: When the vertical distance increase, the range of attenuation change decreases. In turn, attenuated individual signals received by different sensors are less different. From this experiment, we can also infer the effect of sensor density. More sensor deployed in a field, more likely to have a sensor group both covering the path segment of interest and distant from the target at the same time. So higher sensor density can lead to better separation performance.

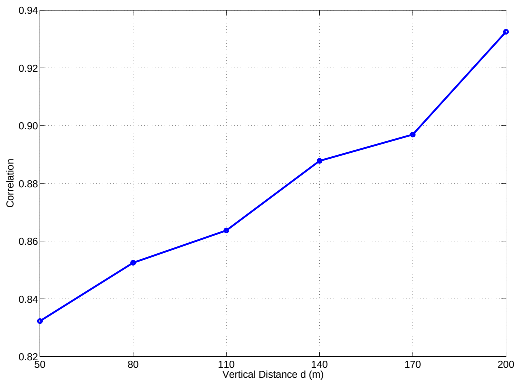


Figure 6. Effect of Attenuation

6.2 Tracking Resolution

We analyze tracking resolution of the algorithm in this section. The purpose of analysis is to estimate achievable performance of the proposed tracking algorithm. We focus on the intersection step in the analysis.

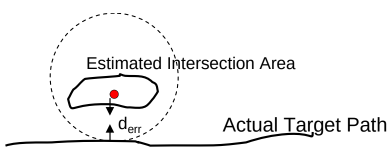


Figure 7. An Example

First, we define *error distance* as follows:

Definition The error distance between a point in one intersection area and the path segment of interest is the minimal distance between the point and any point on the path segment.

Tracking resolution is defined based on the error distance definition:

Definition Tracking resolution is defined as the average of error distance between all the points inside an intersection area and a path segment of interest.

As can be seen in Figure 7 actual target path and estimated intersection area. Error distance d_{err} is the minimum distance between point inside estimated intersection area (represented with dot) and points on the path segment of interest. Tracking resolution is average error distance of all the points inside an estimated intersection area.

We focus on linear path segments in theoretical analysis for the following reasons: (a) Any path can be formed with linear segments. (b) In practice the segment size used in the proposed algorithm is small so that path segments are close to linear. To simplify the analysis of tracking resolution we assume the path segment of interest fit inside the intersection area and perpendicular to the line joining of center of two sensor groups. We assume the sensors are uniformly distributed over the field. So sensor groups are also uniformly distributed over the field.

We assume N sensors are deployed in an field of size a meter by b meter and sensing range of each sensor is R. Both average tracking resolution and the finest tracking resolution are analyzed below.

6.2.1 *Finest Tracking Resolution*

Finest tracking resolution is defined as the achievable minimal mean error distance.

We assume sensor groups are located within a circle of radius r on average. So we have

$$\begin{aligned} \text{Sensor Density} &= \frac{N}{a \times b} \\ &= \frac{n_{group}}{\pi r^2} \end{aligned}$$

where n_{group} is the number of sensors in each sensor group. Thus the average radius r is

$$r = \sqrt{\frac{n_{group} ab}{\pi N}} \tag{4}$$

THEOREM 6.1. *Finest tracking resolution of tracking a linear path segment of length l is $\int_0^{\frac{l}{2}} \{(\frac{y}{tan\theta_2}) - (\frac{y}{tan\theta_1})\} dy$ where $\theta_1 = tan^{-1}(\frac{y}{\sqrt{(R+r)^2 - (\frac{l}{2})^2}})$ and $\theta_2 = sin^{-1}(\frac{y}{R+r})$.*

PROOF. The finest tracking resolution is achieved when the path segment of interest fits exactly into the intersection area of two sensing ranges as shown in Figure 8. In Figure 8, line segment AC is the linear path segment of length l. So the path segment of interest is perpendicular to the line joining of center of sensor groups. The distance d_{err} is the distance between sample point on the path denoted with G and the point on the perimeter of the sensing range denoted with F. Since d_{err} is the shortest distance from F to any points on the path segment, d_{err} is also the error distance between point F and the path segment.

Suppose in Figure.8 the distance between centers of two neighbor sensor groups is $2x$. The value of x can be derived as follows. $\triangle OAB$ is a right angle triangle, $OA = R + r$ which is the sensing radius of sensor group and AB is $\frac{l}{2}$ which is half of the segment length l. From $\triangle OAB$

$$\begin{aligned} (R+r)^2 &= x^2 + (\frac{l}{2})^2 \\ x &= \sqrt{(R+r)^2 - (\frac{l}{2})^2} \end{aligned} \tag{5}$$

Thus distance between neighbor sensor groups $2x = 2\sqrt{(R+r)^2 - (\frac{l}{2})^2}$.

The error distance d_{err} can be derived as follows:
From $\triangle DGE$ as shown in Figure.8.

$$\begin{aligned} tan\theta_1 &= \frac{y}{x_2} \\ x_2 &= \frac{y}{tan\theta_1} \end{aligned} \tag{6}$$

Now from $\triangle FDE$, $FD = R + r$ which is sensing radius of sensor group and $DE = y$ as can be seen in Figure.8. Where

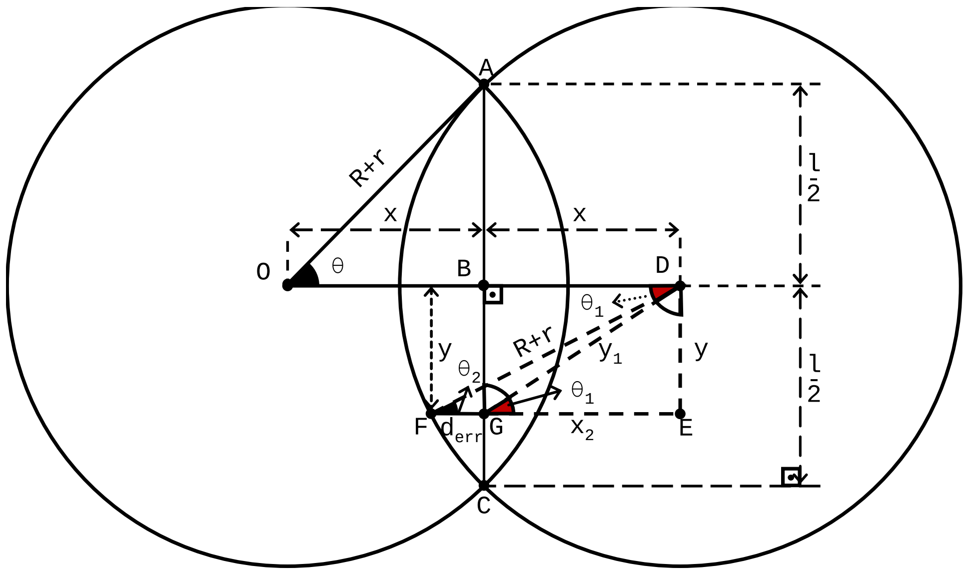


Figure 8. Finest Tracking Resolution

y is the distance between the points G and B as shown in Figure.8.

$$\begin{aligned} sin\theta_2 &= \frac{y}{R+r} \\ \theta_2 &= sin^{-1}(\frac{y}{R+r}) \end{aligned} \tag{7}$$

$$\begin{aligned} tan\theta_2 &= \frac{y}{d_{err}+x_2} \\ d_{err} &= \frac{y}{tan\theta_2} - x_2 \end{aligned} \tag{8}$$

Where x_2 is from Equation6 and θ_2 is from Equation7.

$$d_{err} = \frac{y}{tan\theta_2} - \frac{y}{tan\theta_1} \tag{9}$$

For all the points on the line segment FG, the average error distance is $\frac{d_{err}}{2}$.

To calculate average of error distance for all the points within the intersection area, we integrate $\frac{d_{err}}{2}$ with y between the limits 0 and $\frac{l}{2}$.

Thus the finest tracking resolution is $\frac{1}{2}\int_0^{l/2}\{(\frac{y}{tan\theta_2}) - (\frac{y}{tan\theta_1})\}dy$ Where $\theta_2 = sin^{-1}(\frac{y}{R+r})$ and $\theta_1 = tan^{-1}(\frac{y}{\sqrt{(R+r)^2-(\frac{l}{2})^2}})$. □

COROLLARY 6.2. *When finest tracking resolution is achieved, the distance between the two neighboring sensor blocks is* $2\sqrt{(R+r)^2-(\frac{l}{2})^2}$

The proof of Corollary is contained in the proof of Theorem .

6.2.2 *Average Tracking Resolution*

Average tracking resolution predicts the average tracking accuracy achievable by the proposed tracking algorithm. It is the mean error distance averaged over all the possible cases.

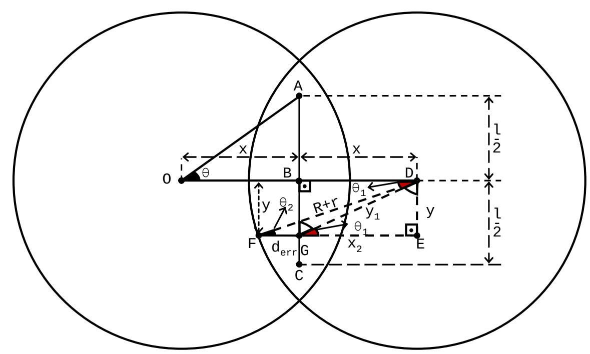
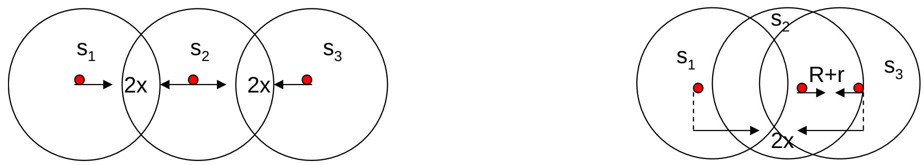


Figure 9. Average Tracking Resolution

THEOREM 6.3. *Average tracking resolution of tracking a linear path segment of length l is* $\int_{R+r}^{Z}\int_0^{\frac{l}{2}}\{(\frac{y}{tan\theta_2}) - (\frac{y}{tan\theta_1})\}dydx$ *where* $Z = 2\times\sqrt{(R+r)^2-(\frac{l}{2})^2}$ *which is the distance between centers of neighbor sensor groups in case of the finest tracking resolution,* $R+r$ *is the distance between centers of neighbor sensor groups in case of the worst case tracking resolution,* $\theta_2 = sin^{-1}(\frac{y}{r+R})$ *and* $\theta_1 = tan^{-1}(\frac{y}{x})$.

PROOF. As shown in Figure 8, suppose the distance between the center of two sensor groups is $2x$. Average tracking resolution can be derived by integral of mean error distance over possible x where x is half of the distance between the center of two sensor groups. Lower limit of x value is from the worst tracking resolution and upper limit is from the finest tracking resolution. The worst case tracking resolution is explained as follows:

Sensor groups are uniformly distributed over the sensor field. Let us say three sensor groups s_1, s_2 and s_3 are placed along a line as shown in Figure10(a), where s_1,s_2 and s_2,s_3



(a) In case of Finest Tracking resolution (b) In case of Worst Tracking Resolution

Figure 10. An Example

have coverage intersection respectively. And let us say the distance between centers of two neighbor sensor groups is same as in case of finest tracking resolution. Now vary the distance between s_2 and s_3 by moving s_3 closer to s_2 as can be seen in Figure10(b). We achieve the worst case when the distance between centers of s_2 and s_3 is $R+r$, since s1 and s3 begin to have intersection in their coverage.
We derived the error distance d_{err} in section 6.2.1.

$$d_{err} = \frac{y}{tan\theta_2} - \frac{y}{tan\theta_1} \tag{10}$$

where $\theta_1 = tan^{-1}(\frac{y}{x})$ and $\theta_2 = sin^{-1}(\frac{y}{r+R})$. So the mean error distance is $\frac{1}{2}\int_0^{l/2}\{(y/\tan\theta_2)-(y/\tan\theta_1)\}dy$.

From Corollary 6.2.1, we know the distance between the center of two sensor groups when the finest tracking resolution is achieved. The worst-case tracking resolution is achieved when the distance between the centers of two sensor groups is $R+r$. The average tracking resolution can be derived by integral of mean error distance over possible x, the half of the distance between the centers of two sensor groups. So the average tracking resolution is $\frac{1}{2}\int_{R+r}^{Z}\int_0^{\frac{l}{2}}\{(y/\tan\theta_2)-(y/\tan\theta_1)\}dydx$ where $Z = 2\times\sqrt{(R+r)^2-(\frac{l}{2})^2}$, $\theta_1 = tan^{-1}(\frac{y}{x})$, and $\theta_2 = sin^{-1}(\frac{y}{r+R})$ $\square$

6.3 Effect of Moving Speed

Targets' moving speed affects performance of tracking algorithm. Tracking algorithms track moving targets by observing changes in sensing signals collected from sensors. If a target moving through a sensor-deployed field with infinity speed, then sensors are not able to observe enough change in sensing signals for tracking. On the other hand, signals reported by sensors are digitized, i.e., sampled from original sensing signals. If a sensor can sample sensing signals with a high sampling rate, the sensing data collected from sensors can possibly capture enough changes for tracking fast-moving targets. To make moving speed discussed in this paper independent from the sampling rate, we use meter per sample interval as the unit for speed.

The effect of moving speed on the performance of the proposed tracking algorithm is twofold: (a) The algorithm processes collected signals in signal segments. When moving speed is low, the signal segments will be long for time slots of same length. In the separation step, the separation performance for longer signal segments is generally better than for shorter signal segments. So in turn, low moving speed can lead to better tracking performance. (b) Long signal segments correspond to long path segment finished by targets. Longer path segments has less chance to be covered entirely in the sensing range of one sensor block. So it is less likely to recover the whole signal segment for longer signal segments. In our experiments, we find separation performance and overall tracking performance is satisfactory when signal segments have 100 data samples. Obviously the longest path segment which can be fitted into a sensing range with radius $(R+r)$ is of length $2(R+r)$. So the proposed algorithm can generate satisfactory result when moving speed is below $\frac{R+r}{50}$ meter per sample interval.

6.4 Complexity vs. Tracking Performance

The complexity of the algorithm is largely determined by the step size shown in Figure 3. To decrease complexity, we can increase the step size. The cost will be less reliable link correlation because of shorter common part between two successive time slots. In the mean time, when step size increases, the estimated path may not be continuous since (a) An estimated path is composed by a series of intersection areas estimated based on signal segments. (b) These intersection areas may not have overlap when the step size is large.

7 Performance Evaluation

We evaluate the performance of the proposed tracking algorithm with extensive simulations with Matlab. We assume acoustic sensors are deployed in the field of interest for tracking purpose.

7.1 Experiment Setup

In the following experiments, the simulated field is a 1600m $\times$ 1600m square area. Sensors are randomly deployed in the field. The movement of targets is restricted to a 1000m $\times$ 1000m *center area* to eliminate boundary effects. The signals used for tracking are real bird signals downloaded from website of Florida Museum of Natural History [46]. The attenuation of sound signals is according to atmospheric sound absorption model [47, 48]. The sensing range of sensors is 230m. Paths followed by targets are generated randomly if not mentioned. The number of sensors in each sensor group n_{group} is 10 if not specified. In the following experiments, targets are moving at a speed below 0.15 meter per sample interval.

7.2 Performance Metrics

As described in Section 5.6 and 5.7, the output of the target-tracking algorithm is the concatenated intersection area. To evaluate the performance according to the concatenated intersection area, we quantize the whole area using 10m $\times$ 10m tiles. The intersection area is represented by a set of points inside the area, each point representing the corner of the corresponding tile. Two metrics are used to evaluate the area: One is the *mean error distance*. It is based on the error distance defined in Section 6.2. The mean error distance is the mean of error distance between all points inside concatenated intersection areas and the actual path taken by a target. The other is the *standard deviation of the error distance* between the points inside the concatenated intersection area and the actual path taken by a target. The first one measures the accuracy of the tracking algorithm and the second measures the precision of the tracking algorithm. If we cast the evaluation of the estimation algorithm in terms of evaluating a statistical estimator, the accuracy corresponds to the bias of the estimator and the precision corresponds to the variance of the estimator.

As described in Section 6.4, step size can affect both tracking performance and computational complexity. A big step size can reduce computation time with the cost of having gaps between concatenated intersection areas. We use *percentage of coverage* to measure the discontinuity in estimated paths. It is equal to one minus the ratio between sum of distance between neighboring intersection areas and length of the actual path. The distance between two intersection areas is defined as in Section 5.7: It is the distance between two closest points in each intersection area. If two intersection areas have overlap, the distance is zero.

7.3 A Typical Example

An example of typical results of our tracking algorithm is shown in Figure 11. The paths taken by these targets are shown in Figure 11(a). We include a zigzag path in this example since zigzag path is one kind of path with high frequency variation. Figure 11(b) shows paths estimated by our algorithm. The estimated paths are drawn in red dots.

7.4 Sensor Density vs Performance

As analyzed in Section 6, sensor density can greatly affect tracking performance. In this series of experiments, we increase the number of sensors in the field from 100 to 1000. The length of signal segment used in this set of experiments is 100 samples. The step size is 10 samples. We will use these algorithm parameters in the following experiments if not explicitly specified.

Figure 12(a) and 12(b) shows the tracking performance under different sensor density. From Figure 12(a), we can observe: (a) The tracking algorithm can both accurately and precisely track targets even when the sensor density is not high. (b) When sensor density increases, the error distance decreases. This is because of two reasons: (a) When sensor density increases, more sensor groups can sense the target of interest. So intersecting sensing areas of more sensor groups can lead to smaller error distance. (b) When sensor density is high, better separation is possible as analyzed in Section 6.1. Figure 12(b) shows that percentage of coverage decreases when sensor density increases. In other words, when sensor density increases, more gaps exist in the estimated paths. It is because of smaller or more precise intersection areas are estimated when sensor density increases. So the distance between two neighboring intersection areas increases and more gaps are created in this way.

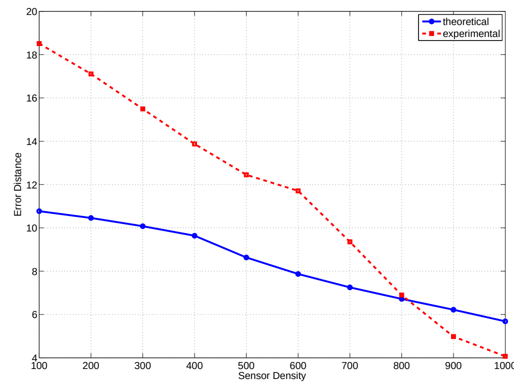


Figure 13. Comparison between Experimental Results and Theoretical Results

We compared theoretical results and experimental results through a series of experiments. The results are shown in Figure 13. In this set of experiment, targets are moving at the speed of 0.03 meter per sample. We can observe the experimental curve follows the theoretical curve of average tracking resolution. The experiments results are in the same order of theoretical results.

7.5 Number of Targets

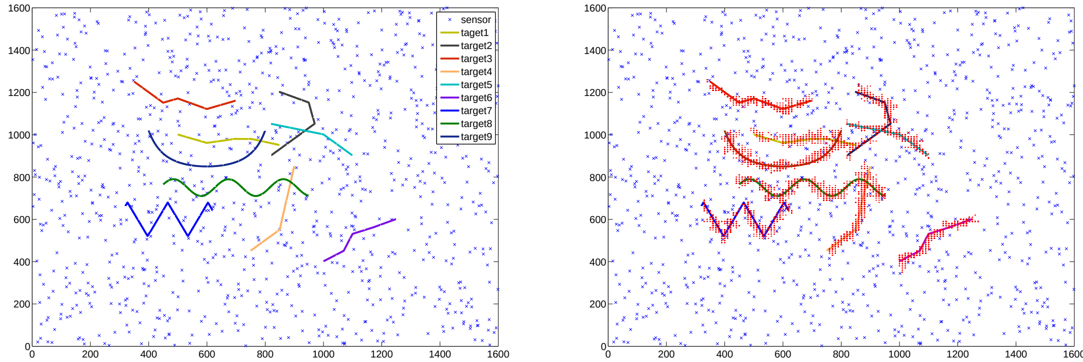
In this set of experiments, we vary the number of targets moving in the field. The results are shown in Figure 14. From Figure 14(a), we can observe: (a) When the field is crowded with targets, our algorithm can still track target with reasonable accuracy and precision. (b) The error distance increases when number of targets increases. It is because our separation step can not perfectly separate out all the signals when the number of moving target increases. As shown in Figure 14(b), the percentage of coverage decreases when the number of targets increases. The decrease is caused by the decrease in separation performance so that intersection areas from different slots are not consistently covering the actual paths.

7.6 Moving Speed

In this set of experiments, we investigate the effect of moving speed on tracking performance. As shown in Figure 15, the error distance increases when moving speed increases. The reasons are as analyzed in Section 6.3: speed increase can lead to decrease of separation performance and less number of sensor groups sensing enough signal for tracking.

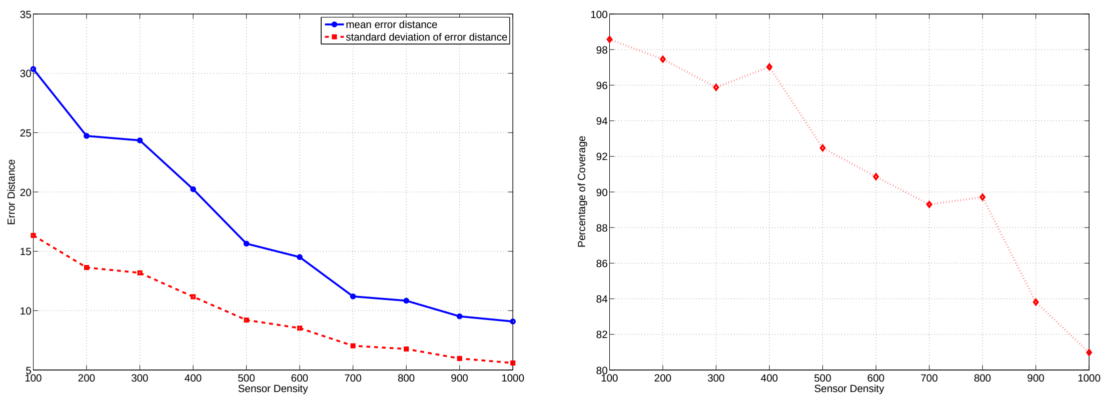
7.7 Segment Length(l_{seg})

This set of experiments focus on the length of signal segments used in tracking algorithm. In this set of experiments, we fix step size and vary segment length. Since the tracking algorithm process signal in the unit of segments, segment length is a critical parameter for the algorithm. The experiment results are shown in Figure 16. As shown in Figure



(a) Experiment Setup (b) Tracking Result

Figure 11. An Example



(a) Error Distance (b) Percentage of Coverage

Figure 12. Tracking Performance for Different Sensor Density

16(a), the error distance increases when the segment length increases. It is because of less number of sensor groups which can sense thew whole segment in their sensing range. The decrease in the number of sensor group also causes the decrease in percentage of coverage as shown in Figure 16(b).

7.8 Step Size(l_{step})

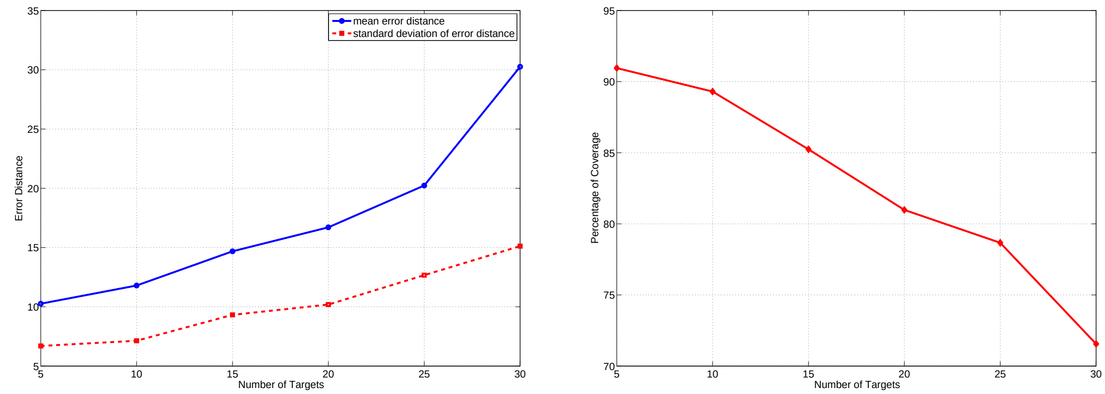
In this set of experiments, we fix segment length and vary the step length. As shown in Figure 17(a), error distance increase with step size. This is because for a certain segment length, larger step size reduces the length of common part of two successive time slots. So the link correlation is less reliable. Obviously when step size is small, the percentage of coverage is around 100 percent. When step size is comparable with segment length, the percentage of coverage is also high. It is because of larger intersection areas caused by less reliable link correlation.

7.9 Effect of Parameter n_{slot} in Center Selection Step

As described in Section 5.5, the parameter n_{slot} is used in center select step to select center segments. The parameter determines the number of successive time slots in consideration for picking center segments. We investigate the parameter with a set of experiments with different n_{slot}. The results are shown in Figure 18. From Figure 18(a), we can observe the sudden decrease in error distance when n_{slot} is larger than one. It shows that increasing n_{slot} can significantly increase the tracking performance by considering more successive time slots for picking center segments. The performance does not change significantly when n_{slot} is larger than four. We can also observe less percentage of coverage when n_{slot} is four in Figure 18. It is because the intersection areas is smaller.

7.10 Effect of Number of Sensors in Sensor Groups

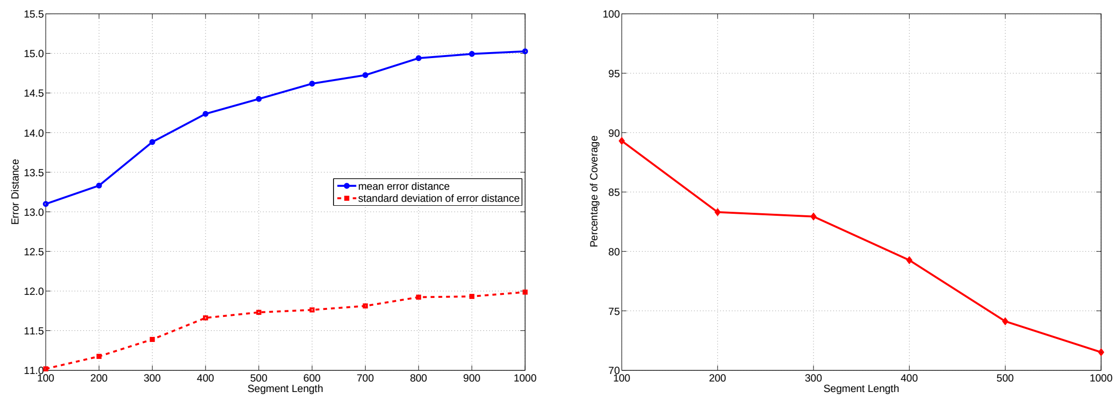
In this subsection, we describe our experiments on the parameter n_{group}, i.e., the number of sensors in each sensor group. The results are shown in Figure 19. As shown in the figure, error distance is larger when n_{group} is too small or too large. When n_{group} is small, the number of targets can be larger than the number of sensors. Generally BSS algorithms perform better when the number of observations is larger than the number of individual signals. So more sensors in a sensor group can lead to better separation performance. But when the number of sensors in sensor group increase, the



(a) Error Distance

(b) Percentage of Coverage

Figure 14. Tracking Performance for Different Number of Targets



(a) Error Distance

(b) Percentage of Coverage

Figure 16. Effect of Signal Segment Length(l_{seg}) on Tracking Performance

sensing range also increases. This lead to larger intersection areas when intersecting larger sensing areas.

7.11 Paths with High-Frequency Variation

In this set of experiments, we experiment on the performance of tracking targets following paths with high-frequency variations. In the experiments, we focus on the path between two points with distance of 300m from each other. The path between these two points is zigzag path. We vary zigzag period in our experiments. From the results shown Figure 20, we can observe the tracking algorithm can track targets following zigzag paths successfully. We believe the slight increase of error distance with the number of zigzag periods is largely because of higher speed to finish longer paths. This experiments demonstrate the benefit of applying BSS algorithm in tracking targets. It enable tracking algorithm to have richer information for target-tracking. So the proposed algorithm can successfully track targets following paths with high-frequency variations.

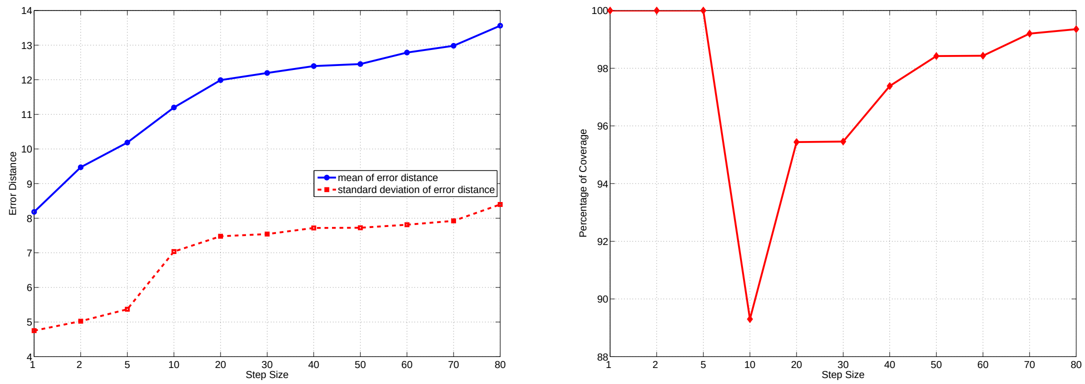
8 Discussion

In this paper, we assume the sensors are placed randomly in the field. From the analysis in Section 6, we know that better separation performance can be achieved when sensor groups can be both distant from targets and sense targets. So we can possibly reduce the number of sensors needed for tracking by placed sensors in a better way such as in clusters. This is one of the topics in our future work.

In this paper, we use BSS algorithms for tracking purpose. The algorithms can also be used to process data collected by sensor networks for other applications. Since data collected by sensors are essentially aggregate data and BSS algorithms can recover data generated by different sources from aggregate data, analysis based on BSS algorithms can be more accurate.

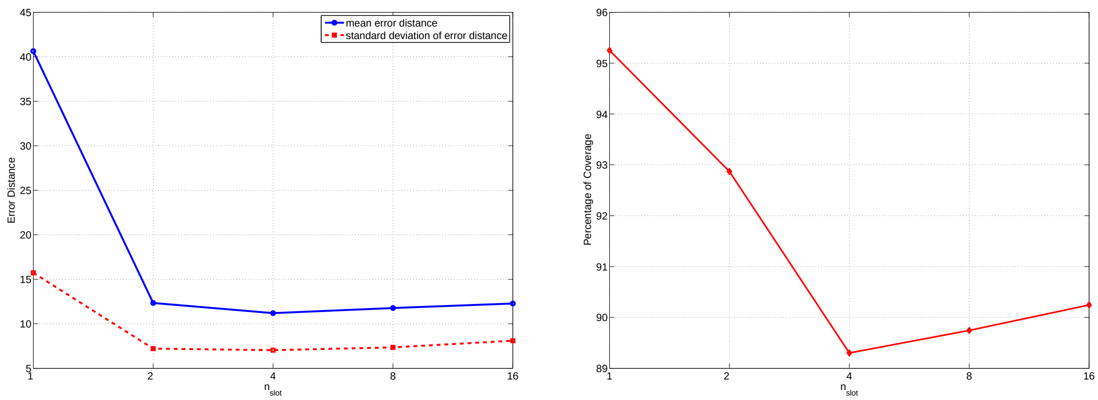
9 Conclusion

We propose a general approach to track multiple targets using wireless sensor networks. The approach is based on blind source separation (BSS) algorithms. By applying BSS algorithms on aggregate signals collected from sensors, we can recover individual signals from targets for tracking. The proposed tracking algorithm fully utilize both spatial and temporal information available for tracking. We evaluate the proposed tracking algorithm both experimentally and theo-



(a) Error Distance (b) Percentage of Coverage

Figure 17. Effect of Step Size(l_{step}) on Tracking Performance



(a) Error Distance (b) Percentage of Coverage

Figure 18. Effect of Parameter n_{slot} on Tracking Performance

retically. The tracking algorithm can track targets both accurately and precisely. Because of richer information made available by BSS algorithms, the proposed algorithm can also track paths with high-frequency variations.

10 References

[1] K. Pister, J. Kahn, and B. Boser, "Smartdust: wireless networks of millimeter-scale sensor nodes," 1999.

[2] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, "Tracking multiple targets using binary proximity sensors," in *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*. New York, NY, USA: ACM, 2007, pp. 529–538.

[3] L. Tong, V. Soon, and Y. F. H. R. Liu, "Indeterminacy and identifiability of blind identification," *IEEE Transactions*, vol. 38, pp. 499–509, March 1991.

[4] L. Tong, Y. Inouye, and R. Liu, "Waveform-preserving blind estimation of multiple independent sources," *IEEE Transactions on Signal Processing*, vol. 41, no. 7, pp. 2461–2470, July 1993.

[5] A. C. R. Szupiluk, "Blind signal separation using second order statistics," *Proc. of SPETO 2001*, pp. 485–488.

[6] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 55.

[7] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. A. Stankovic, T. Abdelzaher, and B. H. Krogh, "Lightweight detection and classification for wireless sensor networks in realistic environments," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2005, pp. 205–217.

[8] Q. Pan, J. Wei, H. Cao, N. Li, and H. Liu, "Improved ds acoustic-seismic modality fusion for ground-moving target classification in wireless sensor networks," *Pat-*

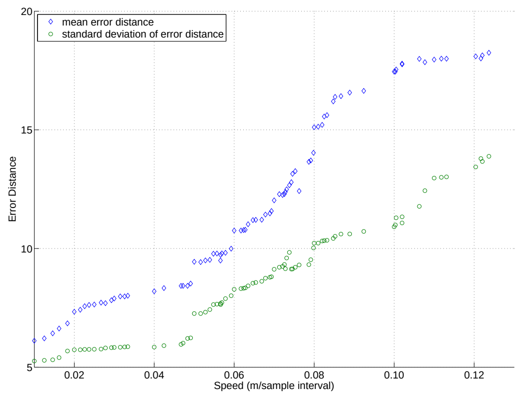


Figure 15. Scatter Plot of Tracking Performance vs. Moving Speed

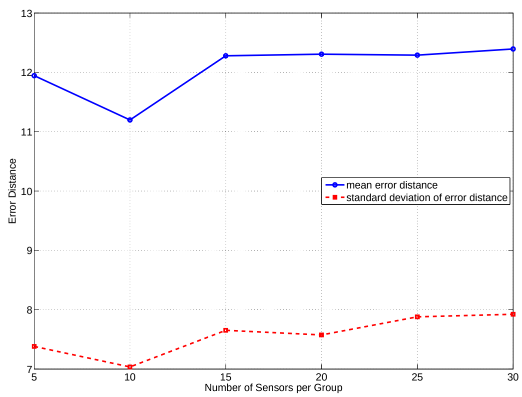


Figure 19. Effect of Number of Sensors in Sensor Groups

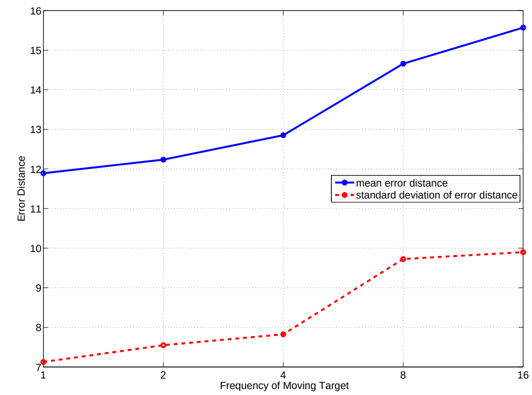


Figure 20. Path with High Frequency Variation

tern Recogn. Lett., vol. 28, no. 16, pp. 2419–2426, 2007.

[9] W. Hu, V. N. Tran, N. Bulusu, C. T. Chou, S. Jha, and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 71.

[10] A. Pereira, A. Monteiro, L. Nunes, and N. Costa, "Wireless sensor network for mobile entities localization people monitor," in *ICSNC '07: Proceedings of the Second International Conference on Systems and Networks Communications*. Washington, DC, USA: IEEE Computer Society, 2007, p. 51.

[11] S. Ray, W. Lai, and I. C. Paschalidis, "Statistical location detection with sensor networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2670–2683, 2006.

[12] S. Ray, D. Starobinski, A. Trachtenberg, and R. Ungrangsi, "Robust location detection with sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1016–1025, 2004.

[13] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," vol. 2, 2000, pp. 775–784 vol.2. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=832252

[14] M. Drinic, D. Kirovski, and M. Potkonjak, "Model-based compression in wireless ad hoc networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 231–242.

[15] B. Zaidi, Z.R. Mark, "A mobility tracking model for wireless ad hoc networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, March 2003, pp. 1790– 1795.

[16] M. Moghavvemi and L. C. Seng, "Pyroelectric infrared sensor for intruder detection," in *TENCON 2004. 2004 IEEE Region 10 Conference*, Nov. 2004, pp. 656–659.

[17] R. F. Lutomirski and R. G. Buser, "Phase difference and angle-of-arrival fluctuations in tracking a moving point source," *Appl. Opt.*, vol. 13, no. 12, p. 2869, 1974. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-13-12-2869

[18] Edwards and R. G. Jayne, "Location tracking using differential range measurements," vol. 27. ACTA PRESS, 2005, pp. 199–205.

[19] D. Macagnano, G. Destino, F. Esposito, and G. Abreu, "Mac performances for localization and tracking in wireless sensor networks," in *Positioning, Navigation and Communication, 2007. WPNC apos;07.*, vol. 22, March 2007, pp. 297 – 302.

[20] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 105–111.

[21] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 32–43.

[Online]. Available: http://portal.acm.org/citation.cfm?id=345917

[22] K. Doğançay and A. Hashemi-Sakhtsari, "Target tracking by time difference of arrival using recursive smoothing," *Signal Process.*, vol. 85, no. 4, pp. 667–679, 2005.

[23] J. Hightower, C. Vakili, G. Borriello, and R. Want, "Design and calibration of the spoton ad-hoc location sensing system," 2001. [Online]. Available: citeseer.ist.psu.edu/hightower01design.html

[24] G. V. Zàruba, M. Huber, F. A. Kamangar, and I. Chlamtac, "Indoor location tracking using rssi readings from a single wi-fi access point," *Wirel. Netw.*, vol. 13, no. 2, pp. 221–235, 2007.

[25] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, Tel Aviv, Israel, 2000, pp. 775–784.

[26] S.-P. K. Y.-C. T. F.-J. W. C.-Y. Lin, "A probabilistic signal-strength-based evaluation methodology for sensor network deployment," *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference*, vol. 1, pp. 319 – 324, March 2005.

[27] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus., "Tracking a moving object with a binary sensor network," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 150–161.

[28] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "target tracking with binary proximity sensors." *In Proc.IPSN*, 2005.

[29] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri., "Target tracking with binary proximity sensors: Fundamental limits, minimal descriptions, and algorithms." *In Proc. of ACM SenSys*, 2006.

[30] B. Malhotra and A. Aravind, "Path-adaptive on-site tracking in wireless sensor networks," *IEICE - Trans. Inf. Syst.*, vol. E89-D, no. 2, pp. 536–545, 2006.

[31] J. F. Scholl, J. R. Agre, , and L. P. Clare, "Wavelet packet target classification schemes," *Proc. 1999 Meeting of the MSS Specialty Group on Battlefield Acoustic and Seismic Sensing*, 1999.

[32] S. Yang, Y. Kim, and H. Choi, "Vehicle identification using wireless sensor networks," in *Proceedings. IEEE*, March 2007, pp. 41 – 46.

[33] K. Sreenath, F. L. Lewis, and D. O. Popa, "Simultaneous adaptive localization of a wireless sensor network," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 2, pp. 14–28, 2007.

[34] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *Oceanic Engineering, IEEE Journal of*, vol. 31, no. 4, pp. 949–958, Oct. 2006.

[35] D. S. Friedlander and S. Phoha, "Semantic information fusion for coordinated signal processing in mobile sensor networks," *The International Journal of High Performance Computing Applications*, 2002.

[36] C. Savarese, J. Rabaey, K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *ATEC '02: Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2002, pp. 317–327.

[37] C. Tang and C. S. Raghavendra, "Compression techniques for wireless sensor networks," pp. 207–231, 2004.

[38] P. von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing*. New York, NY, USA: ACM, 2004, pp. 60–66.

[39] J. Cardoso, "Blind signal separation: statistical principles," vol. 9, no. 10, pp. 2009–2025, 1998. [Online]. Available: http://citeseer.ist.psu.edu/221558.html

[40] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[41] Z. He, L. Yang, J. Liu, Z. Lu, C. He, and Y. Shi, "Blind source separation using clustering-based multivariate density estimation algorithm," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 48, no. 2, pp. 575–579, Feb. 2000.

[42] A. Hyvrinen, "Fast and robust xed-point algorithms for independent component analysis," *IEEE Trans. on Neural Networks*, vol. 10, 1999.

[43] A. Hyvrinen and E. Oja, "A fast xed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, pp. 3 917 392–0, 1997.

[44] M. Gaeta and J.-L. Lacoume, "Source separation without a priori knowledge: the maximum likelihood solution," in *In Proc. EUSIPCO*, 1990, pp. 621–624.

[45] D. Pham and P. Garrat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," 1997. [Online]. Available: citeseer.ist.psu.edu/pham97blind.html

[46] J. W. Hardy, "Sounds of Florida's Birds," Florida Museum of Natural History, 1998. [Online]. Available: www.flmnh.ufl.edu/birds/sounds.htm

[47] Kinsler, et al, in *Fundamentals of Acoustics*. New York: John Wiley Sons, 2000.

[48] H. E. Bass, L. C. Sutherland, A. J. Zuckerwar,

D. T. Blackstock, and D. M. Hester, “Atmospheric absorption of sound: Further developments,” *The Journal of the Acoustical Society of America*, vol. 97, no. 1, pp. 680–683, 1995. [Online]. Available: http://link.aip.org/link/?JAS/97/680/1