*Ad Hoc & Sensor Wireless Networks*, Vol. 0, pp. 1–18
Reprints available directly from the publisher
Photocopying permitted by license only



# Energy Confirmable Overlapping Target Tracking Based on Compressive Sensing in Wireless Sensor Networks

JUAN LUO[1,*], ZANYI HE[1], YU LIU[1], JUNLI ZHA[1] AND KEQIN LI[2]

[1]*College of Computer Science and Electronic Engineering, Hunan University, CHINA*
[2]*Department of Computer Science, State University of New York, New Paltz, NY 12561,USA*

*Received: July 23, 2014. Accepted: May 20, 2015.*

Localization is highly critical for wireless sensor network applications. The present paper makes the following noticeable contributions. First, an energy confirmable overlapping tracking algorithm for mobile targets is proposed in wireless sensor networks. Different from most target localization algorithms based on compressive sensing, it improves localization accuracy through overlapping area and predicting regions in online tracking phase. Second, theoretical analyses suggest that grids number in an overlapping area is related to energy consumption. By exploiting a common communication schedule, we derive the compressive sensing tracking for the solution and formulate the threshold of grids number and the energy consumption. Third, our algorithm shows good scalability. Since only the network topology information around the unknown nodes is used, it can be applied to large-scale wireless sensor networks. Finally, analytical studies and simulations are provided to show that our proposed approach achieves significant tracking accuracy in four different trajectories.

*Keywords:* Compressive sensing, energy-efficient, tracking, wireless sensor networks

## 1 INTRODUCTION

### 1.1 Motivation

Advances in embedded computing, sensing, and communication technologies offer unprecedented opportunities to add intelligence into sensors and

* Contact author: E-mail: juanluo@hnu.edu.cn

enable the creation of environment-aware functions for large-scale monitoring and response in wireless sensor networks (WSNs). The problem of efficient tracking of targets in WSNs is encountered in many diverse applications such as electronic warfare, surveillance, mobile user location, and so on [5, 9, 14]. A wide range of well-known applications, such as traffic control, environmental surveillance, and smart home, require robust and accurate tracking in WSNs [18].

Compressive sensing (CS) based tracking has gained much attention in recent years [4, 6, 10], which usually consists of three phases, i.e., an offline training phase, an online positioning phase and a predicting phase [2]. In the offline training phase, some grid-based nodes are used to collect received signal strength (RSS) vectors from available anchors to generate a fingerprint database. In the online positioning phase, a sampled RSS vector of a mobile target will be compared with the fingerprints stored in the database. For example, the strongest node selection algorithm selects a node with the maximum signal strength to locate the target. In the online predicting phase, the Kalman filter is used to predict the location.

The theory of compressive sensing, also known as compressive sampling or CS, is a novel sensing/sampling paradigm that goes against the common wisdom in data acquisition [8]. Data acquisition typically works as follows: massive amounts of data are collected only to be, in large part, discarded at the compression stage to facilitate storage and transmission. However, compressive sensing theory asserts that one can recover certain signals from far fewer samples or measurements than traditional methods. To track the location of a target in a large-scale environment, the number of samples becomes very large, which increases the complexity in generating the offline database. Compressive sensing has been proposed as an efficient approach to reducing the number of samples in the offline training phase. The idea of compressive sensing localization is to reconstruct a sparse signal with a much lower sampling rate. In the offline training phase, much fewer samples are collected to generate the offline database. In the online positioning phase, a sampled RSS vector of a mobile target is first examined to find which grids it belongs to, and then a compressive sensing algorithm is used to locate the target.

### 1.2 Related Work

Compressive sensing is an emerging field. Various research efforts have been devoted to exploring different in-network localization solutions in WSNs based on wireless signal readings. Bahl and Padmanabhan proposed a classic indoor localization mechanism called Fingerprint localization in [3]. Signals are collected and processed according to Nyquist sampling principle (double signals maximum frequency) in the Fingerprint approach, based on empirical measurements with signal propagation modeling. And the locations are determined by the overlapping coverage in the region of interest. Because of offline signal dictionary, the size of signal samples in Fingerprint localization is good enough to reflect complex indoor environmental factors, and therefore, it can reduce the localization error efficiently. Fingerprint localization has gained much attention recently due to its ease of implementation [15]. However, under changing environments, localization using the Fingerprint method may produce too much communication overhead and high computation cost during signal collection and processing [17].

Candés *et al.* proposed a compressive sensing (CS) theory to recover sparse signals with far fewer noisy measurements than predicted by the Nyquist theorem [7, 11]. With the development of this theory in recent years, Feng *et al.* designed a localization model based on compressive sensing in wireless sensor networks [13]. Then, a clustering algorithm is used to group all anchors and achieve a redundant dictionary. In the online phase, a cluster header is chosen to determine the most likely cluster where an unknown node is located, and then one can run location algorithms and derive the position of the unknown node. Recently, many researchers [16,19] have focused on accurate tracking mechanisms in wireless sensor networks. However, there is little focus on tracking accuracy combined with efficient energy consumption.

### 1.3 Contributions

Localization using CS can greatly reduce sampling cost compared to the Fingerprint approach. However, the pre-processing of signal collection and clustering in [13] to run CS localization still results in extra computation and energy consumption. Other work [1, 12, 20] using similar ideas on static redundant dictionary cannot solve this problem either. Thus, a more efficient mechanism is needed to select the useful network grids and achieve a dynamic signal dictionary. In this paper, we propose a novel energy confirmable overlapping tracking algorithm based on compressive sensing, which is also composed of offline and online phases.

Overall, the present paper makes the following noticeable contributions.

First, an energy confirmable overlapping tracking algorithm for mobile targets is proposed in wireless sensor networks. Different from most target localization algorithms based on compressive sensing, it improves localization accuracy through overlapping area and predicting regions in an online tracking phase.

Second, theoretical analyses suggest that grids number in overlapping area is related to energy consumption. By exploiting a common communication schedule, we derive the compressive sensing tracking for the solution and formulate the threshold of grids number and the energy consumption.

| Symbol | Description |
|---|---|
| $N$ | Number of grids |
| $\Psi$ | A measurement matrix |
| $M$ | Number of measurement |
| $\Phi$ | Sampling matrix |
| $\tilde{N}$ | A subset of $N$ |
| $\theta$ | Recovery signal |
| $\varepsilon$ | Error of recovery |
| $r$ | Anchors communication radius |
| $L$ | Number of anchors |
| $\sigma$ | Location error |

TABLE 1
Notations and definitions

Third, our algorithm shows good scalability. Since only the network topology information around the unknown nodes is used, it can be applied to large scale wireless sensor networks.

The rest of the paper is organized as follows. The traditional compressive sensing tracking model is presented in Section 2. An improved compressive sensing tracking algorithm using the Kalman filter is then proposed in Section 3. Unlike most approaches in the literature [13], we allow a mobile target to have a redundant dictionary in time without a clustering algorithm. Furthermore, the proposed algorithm is proved to have confirmable energy consumption in Section 4. Section 5 presents the simulation results. The letter is concluded in Section 6.

## 2 COMPRESSIVE SENSING TRACKING MODEL

The traditional compressive sensing tracking model consists of three steps, i.e., offline RSS collection, online localization, and online location prediction.

For convenience, notations used in this paper are summarized in Table 1.

### 2.1 Offline Phase

*Offline RSS Collection*
Assume that there are $n \times n$ grids and $L$ anchors in a target tracking field, as showed in Figure 1. We define $N$=$n \times n$, as the total number of grids. We use the subscript $i = 1, 2, ..., L$ and $j = 1, 2, ..., N$ to denote the $i$th anchor and the $j$th grid, respectively. For a grid $j$, the collected RSS vectors in the area are denoted by $\varphi_{i,j} = \left(\varphi_{1,j}, \varphi_{2,j}, \ldots, \varphi_{L,j}\right)^T$. For each anchor, we collect $q$ such measurements, and each measurement is $\varphi_{i,j}(\tau)$, which represents the

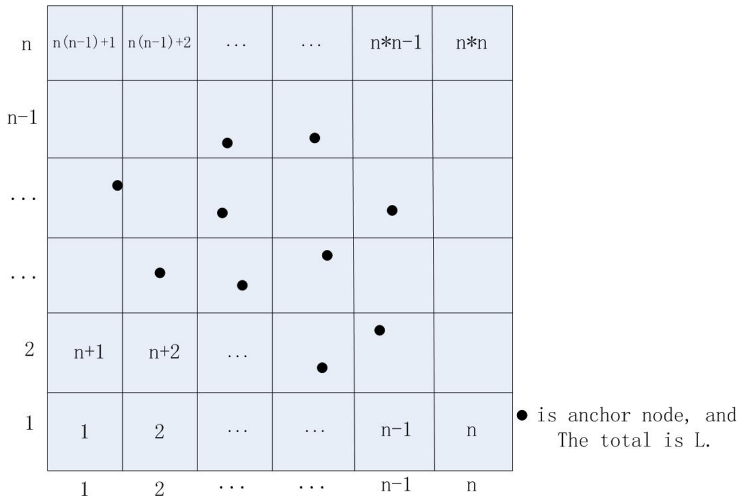


FIGURE 1
$n \times n$ grids and $L$ anchors

$\tau$th RSS that $j$th grid receives from anchor $i$. Hence the collected RSS vector of a specific grid $j$ for one anchor $i$ is denoted by $\varphi_{i,j} = \sum_{\tau=1}^{q} \varphi_{i,j}(\tau)/q$. First, we generate the original $L \times N$ database matrix as follows:

$$\Psi = \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} & \cdots & \varphi_{1,N} \\ \varphi_{2,1} & \varphi_{2,2} & \cdots & \varphi_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{L,1} & \varphi_{L,2} & \cdots & \varphi_{L,N} \end{bmatrix}.$$

If anchor $i$ is not perceived by grid $j$, then $\varphi_{i,j} = 0$.

After getting the offline RSS database, clustering algorithms such as K-means, K-modes, fuzzy c-means, *et al.*, are used to cluster grids. Thus, a target area is divided into several clusters.

*Weighting Anchors*

In conventional offline approaches, the average value is often taken as the unique metric of localization. However, it is not practical for our problem. As discussed above, a complex environment could cause some outliers, and even remove a cup on the desk. Another example is that some researchers bring many kinds of new equipments into a laboratory which work in the same frequency band with our sensors. Under either condition, it is necessary to set a new rule to evaluate anchors.

We formalize this rule as the weighted value of an anchor, which reflects the contribution to the whole offline sampling phase. In the complete offline database, $w_i$ denotes the weighted value of $i$th anchor.

In this work, we employ the anchor utilization rate to estimate the weighted value. Intuitively, a testing anchor, which has more receiving RSS information sent by adjacent anchors, is suitable to a higher weighted value. There- fore, the weighted value should be directly proportional to the sensing adjacent anchors. We define the weighted value as follows:

$$W = [w_1, w_2, \cdots, w_{\mathrm{L}}]^{\mathrm{T}},$$

and

$$w_i = \sum\nolimits_{k=1}^{N} Num_k/N,$$

where $w_i$ describes the weighted value of $i$th anchor, $Num_k$ is the total number of the grids that the anchor is perceived, and $N$ is the total number of all grids.

*Constructing Complete Databases*

As we conduct the experiment in two dimensions, the important parameters in localization should be x-coordinate and y-coordinate of anchors. Thus, each anchors x-coordinate, y-coordinate, RSS value, and weighted number are combined together to constitute the complete offline database, as shown in Eq. (1),

$$(x_i, y_i; \psi_i; w_i), i = 1, \cdots, N, \tag{1}$$

where $(x_i, y_i)$ is the position of anchor $i$.

## 2.2 Online Location

Compressive sensing (CS) is designed here to support fast localization efficiently. First, through a cluster matching algorithm we construct a measurement matrix $\Psi$. $y$ is the M-dimension projective under measurement matrix. Next, sampling matrix $\Phi$ is selected according to online RSS information. Meanwhile, CS collects online RSS readings $y$. Finally, L1-norm minimization algorithm is used by CS to recover signal $\theta$, which includes location information in our scenario. In Eq. (2):

$$\theta = \arg\min_{\theta \in R^{\tilde{N}}} \|\theta\|_1, \text{ such that } y = \Psi\Phi\theta + \varepsilon, \tag{2}$$

$\theta$ is a column vector denoting discrete-time signal in the $N$-dimensional space, and $\varepsilon$ is the error of the recovery.

### 2.3 Online Location Prediction

Many prediction algorithms could be used to track a moving target, such as particle filter, Kalman filter, Monte Carlo *et al*. In this letter, the Kalman filter is used to predict the target.

## 3 OVERLAPPING TRACKING ALGORITHM BASED ON COMPRESSIVE SENSING

Our proposed algorithm is based on the traditional compressive sensing tracking algorithm. We use an overlapping mechanism to construct a dynamic measurement matrix, and thus eliminate the offline clustering algorithm and the online cluster matching algorithm. An overlapping tracking approach is proposed. Moreover, a threshold value for grids in an overlapping area is formulated accurately.

In the online phase, a box-overlapping mechanism is utilized to get useful grids without a clustering algorithm. Assume that there are four anchor nodes around the target. Let $Y_i$ represent the RSS vector which is received by a target from anchor $i$. We divide anchors into two parts by $Y_i \times \omega_i$. The first part includes the value of all zeros, and the remaining values compose the second part. The top four anchors are prepared for overlapping. The communication radius is defined as $r$. Conventionally, if a node can hear from another one, the other node should be in this node's communication range. So, if node $A$ can perceive node $B$, then the distance between $A$ and $B$ should be smaller than $r$. As shown in Figure 2, only when four anchors can hear from a moving target, these four anchors' communication regions are overlapped. Node1 to Node4 are the candidate anchors and the moving target should be in the overlapping area which is marked as shadow. We define this shadowing area as

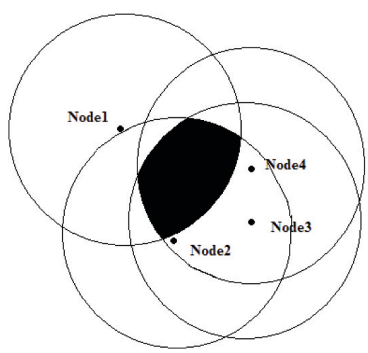


FIGURE 2
Overlapping mechanism

*Region*1 and overlapping operation as $\cap$. The first overlapping can be denoted by $Region1 = R_1 \cap R_2 \cap R_3 \cap R_4$, where $R_i$ is the communication region of *anchor i*.

After first online overlapping, we get the rough area *Region*1. According to the prediction mechanism, there would be a prediction area from time $T - 1$, we define it to be $Region1'$. In order to get accurate area, we conduct the operation $Region2' = Region1 \cap Region1'$.

$Region1'$ consists of two part, i.e., time $T - 1$ prediction location region and time $T - 1$ overlapping prediction region. From the prediction operation, we can get time $T - 1$ predict target's coordinate. Note that a tracked target has its own maximum movement speed, and moreover it could predict the next possible area by itself. A target's maximum movement speed and unit time are defined as $\upsilon$ and $t$ respectively. When at time $T$, the target could be in a circle whose center is the last time's coordinate and the radius is $\upsilon \times t$. We define this area as *Region*3. *Region*2 is defined as time $T - 1$ prediction location. After overlapping in time $T - 1$, we get overlapping region *Region*2. According to the same mechanism, we get the overlapping prediction region *Region*2. Therefore, we conduct the operation $Region1' = Region3 \cap Region2$ to get a smaller overlapping area.

There will be useful grids in $Region2'$. We further collect these grids from in $Region2'$, and define them as set $G = \{g_1, g_2, ..., g_{\tilde{N}}\}$. Choosing these grids data from offline database, and then we generate the measurement matrix $\Psi'$.

In the online positioning phase, a mobile target sends its RSS vectors to the localization server. That is the set $Y = \{Y_1, Y_2, ..., Y_L\}$. With the aid of overlapping, the server first gets the measurement matrix $\psi'$, and then collects the $M$ largest RSS vectors. We further find the anchors that these values belong to. So, the online observation matrix $\Phi$ is generated by these anchors.

In order to get a target's location at time $T$, the compressive sensing algorithm is used as shown in Eq. (2), where $\varepsilon$ is the error, and $\theta$ is the location set.

After localization, we could get the location set $\theta = \{\theta_1, \theta_2, ..., \theta_{\tilde{N}}\}$. We choose all these grids' location from the offline database and use a weighted method to get the target's position:

$$P_x = \sum\nolimits_{k=1}^{\tilde{N}} \theta_k x_k,$$

and

$$P_y = \sum\nolimits_{k=1}^{\tilde{N}} \theta_k y_k.$$

For next time tracking, the prediction area should be used. Finally, the Kalman filter is used to predict the next position.The proposed algorithm is summarized as Algorithm 1.

**Algorithm 1**. Overlapping Tracking Algorithm

*Input*: All grids' coordinate $P(x, y)$, communication region $R$, the maximum speed $v$, measurements $Y$, and offline database $\Psi'$.
*Output*: A target's prediction position $P_x$, $P_y$.

/* Initialization */
$\theta = \Gamma$, the set of nonzero elements in $\theta$ is null.
/* Steps */
(1) Collect offline training phase RSS vectors for each grid and generate the offline database.
(2) Overlap anchors' communication region and achieve $Region1 = R_1 \cap R_2 \cap R_3 \cap \cdots$.
(3) Overlap time $T-1$ prediction location region $Region3$ and overlapping prediction region $Region2$ to get $Region1' = Region3 \cap Region2$.
(4) Overlap $Region1$ in time $T$ and $Region1'$ in time $T-1$ to get the final target region $Region2' = Region1 \cap Region1'$.
(5) Select useful grids in $Region2'$ to conduct the measurement matrix $\psi'$, and use the compressive sensing algorithm to locate the target and the Kalman filter to predict the position for the next time.
(6) Return the prediction position $P_x$, $P_y$.

## 4 FORMULATION FOR CONFIRMABLE ENERGY CONSUMPTION

Suppose $r$ is communication radius, $d_1$, $d_2$, $d_3$ are the length of the grids. $A$ is the grid RSS collecting node, and $B$ is one of anchors that are located nearest to $A$. In the offline phase, we should collect RSS information from anchors that $A$ can hear from. So, $A$'s nearest neighbor $B$ should hear from $A$; otherwise $A$ would have no RSS information in the offline phase. In order to get each grid's RSS information from all anchors, we should ensure that a node in the grid can hear from at least the nearest anchor. Therefore, grid's edge $d_{grid}$ should meet $d_{grid} \leq r$. As Figure 3(a) shows, if $B$ is in $d_1$, it can hear from $A$, whereas they cannot hear from each other if $B$ is in $d_2$.

However, if $d_{grid}$ is much smaller than $r$, then there would be a lot of anchors within $A$'s communication area. Such situation could affect each anchor's communication with $A$, and lead to much inaccurate RSS information. Finally, it may cause location error. As shown in Fig. 3(b), when $d_{grid} \leq d_3$, in $A$'s lower right area there might be 13 anchors which could hear from $A$. And to the whole area, there might be up to 60 anchors within

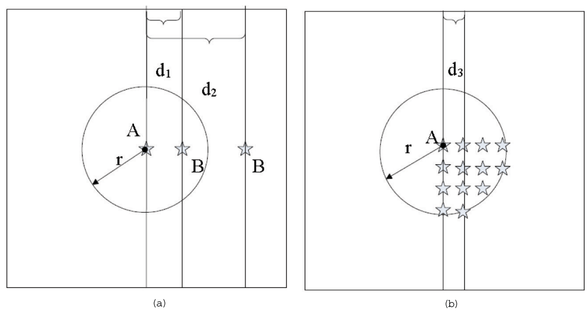


FIGURE 3
Communication with each other

the communication area. With much RSS information losing, interferences can exist in this situation. So, $d_{grid}$ must have a lower limit value to avoid this situation.

**Theorem 1.** *$R$ is the communication radius of anchor. $R_{inf}$ is the interference radius. $d$ is the distance between sending node and receiving node. $d_{inf}$ is the distance between the interference node and the collecting node. $SNR\_THRESHOLD$ is maximum signal-to-noise ratio (SNR) which the network can tolerate. There is the following relation:*

$$\begin{cases} R < R_{inf} & d > R/(SNR\_THRESHOLD)^{1/4} \\ R \geq R_{inf} & d \leq R/(SNR\_THRESHOLD)^{1/4}. \end{cases}$$

*Proof.* Based on the TWO-WAY GRIOUND model, the received power of receiving node is

$$P_r = P_t G_t G_r (h_t h_r)^2 / d^4.$$

$P_t$ represents the transmitted power. $P_r$ is the received power. $G_t$ and $G_r$ represent antenna gain of sending and receiving nodes respectively. $h_t$ and $h_r$ are the height of antenna for sending and receiving nodes respectively. $I$ is the interference loss.

According to the 802.11 protocol, the signal is valid only when its $SNR$ is greater than a certain set of threshold. $P_g$ is the power received from sender by the receiver. And $P_{inf}$ is the power received from the interference node by the receiver.

$$SNR = P_g/P_{inf} = (d_{inf}/d)^4 \geq SNR\_THRESHOLD.$$

We can get

$$d_{inf} \geq SNR\_THRESHOLD^{1/4} \times d.$$

That means it is a successful reception only when the distance between interference node and the receiver is greater than $SNR\_THRESHOLD^{1/4} \times d$. We define the threshold value as interference radius $R_{inf}$.

$$d_{inf} \geq SNR\_THRESHOLD^{1/4} \times d.$$

If $d$ satisfies

$$d > R/(SNR\_THRESHOLD)^{1/4},$$

then we have

$$R < R_{inf}.$$

Otherwise, if $d$ satisfies

$$d \leq R/(SNR\_THRESHOLD)^{1/4},$$

then we have

$$R \geq R_{inf}.$$

This completes the proof of Theorem 1.

**Theorem 2.** *The grid number* $\tilde{N}$ *should satisfy the condition* $\frac{S_{region}}{r^2} \leq \tilde{N} \leq \frac{S_{region}}{(10^{\frac{\lg(I_{\min}-\Lambda)}{\alpha}})^2}$, *where* $\Lambda = \lg(\frac{P_r}{P_t G_t G_r (h_t h_r)^2})$, *and make energy consumption varying in certain range.*

*Proof.* Suppose the limits of overlapping area are *top*, *bottom*, *left*, and *right*. The overlapping area is approximately $S_{region} = (right - left) \times (top - bottom)$. $d_{grid}$ is the length of grid and $d_{region}$ is the longest edge of the area. $P_t$ is the transmission power. $G_t$ and $G_r$ represent antenna gain of sending and receiving nodes respectively. $h_t$ and $h_B$ are the height of antenna for sending and receiving nodes respectively. $I$ is interference loss, $r$ is communication ration, $\alpha$ is interference loss coefficient.

According to Theorem 1, the energy consumption model can be described as Eq. (3):

$$E_\sigma \propto \lambda \tilde{N},\ \lambda = O(ML). \tag{3}$$

According to Eq. (3), when $M$ and $L$ are constants, energy consumption depends on $\tilde{N}$.

We can get the number of grids in overlapping area by principle of square area calculation formula, as shown in Eq. (4).

$$\tilde{N} = \frac{(right-left)\times(top-bottom)}{{d_{grid}}^2} = \frac{S_{region}}{{d_{grid}}^2}, 0 \le d_{grid} \le d_{region}. \tag{4}$$

According to Eq. (4), $\tilde{N}$ is inversely proportional to the square of $d_{grid}$.

$S_{region} = (right - left) \times (top - bottom)$ is the area of overlapping region, $d_{grid}$ is the length of the grid, and $d_{region}$ is the longest edge of the area. So $\frac{S_{region}}{{d_{grid}}^2}$ is the number of grid that in the overlapping region. If $d_{grid}$ increases indefinitely to the longest edge of the region, the time of positioning will be shorter. But the accuracy of positioning will be lower, and must make the grid in this region. So $d_{grid}$ should be adjusted under certain conditions.

Premise for grid division is that, each node in the grid should at least hear from anchors in the neighboring grid point, that is, $0 \le d_{grid} \le r$.

However, if edges of a grid are much closer to each other, a lot of RSS information will be lost, and interference of each node may cause a lower localization error. So, we should find lower limits for the length of grid edge.

The classical transmission model is shown in Eq. (5):

$$P_r = \frac{P_t G_t G_r I}{d^\alpha}(h_t h_r)^2. \tag{5}$$

According to Eq. (5), we can get

$$\lg(I) = \alpha \lg(d) + \Lambda, \tag{6}$$

where $\Lambda = \lg(\frac{P_r}{P_t G_t G_r (h_t h_r)^2})$ is a constant. $d$ is the distance between communication nodes.

The interference loss coefficient $\alpha$ is a fixed number in certain environment. According to Eq. (6), we can get the smallest $d$ when interference loss is the smallest. So, we can get Eq. (7):

$$10^{(\lg(I_{\min})-\Lambda)/\alpha} \le \quad d_{grid} \le r. \tag{7}$$

According to Eq. (7) and Eq. (4), we can get

$$\frac{S_{region}}{r^2} \leq \tilde{N} \leq \frac{S_{region}}{\left(10^{\frac{\lg(I_{\min})-\Lambda}{\alpha}}\right)^2}.$$

This completes the proof of Theorem 2.

Therefore, according to Eq. (3), given certain measurements $M$ and $L$ anchors, with certain $\tilde{N}$, the energy consumption $E_\sigma$ is confirmable.

## 5 SIMULATION RESULTS

We simulate the algorithm in a $100 \times 100m^2$ field. We create $10^4$ grid points with the grid side length of $1m$. There are 25 anchors placed at the grid points. In the simulation, $N = 10000$, $L = 25$. As shown in Table 2, $v_x$, $A_x$, $v_y$ and $A_y$ are moving speed and accelerated speed in x-coordinates and y-coordinates of nodes, respectively.

The localization error for a mobile is defined as

$$\sigma = \sqrt{(x - P_x)^2 + \left(y - P_y\right)^2},$$

where $(x, y)$ and $(P_x, P_y)$ are its true location and estimated location, respectively.

Table 2 shows the moving preferences for four trajectories of the target. We varied the target's speed and accelerated speed to change its moving direction and speed.

Figure 4 shows the true and estimated four trajectories with the moving preferences in Table 2. As shown in Figure 4, we can see that our algorithm shows higher positional accuracy. The positional accuracy is increasing because through many times of region overlapping, the determine area of localization is shrunk. The individual $x$ and $y$ coordinates of the true tracks and the estimated positions in four cases are shown in Figure 5(a) and Figure

| **Trajectory** | $v_x(m/s)$ | $v_y(m/s)$ | $A_x(m/s^2)$ | $A_y(m/s^2)$ |
|---|---|---|---|---|
| 1st TRACKING | 7 | 0 | 0 | -2 |
| 2nd TRACKING | 3 | 4 | -1 | 0.6 |
| 3rd TRACKING | 6 | 3 | 0.19 | 1 |
| 4th TRACKING | 5 | 3 | 0 | 0.16 |

TABLE 2
Moving preferences for four trajectories

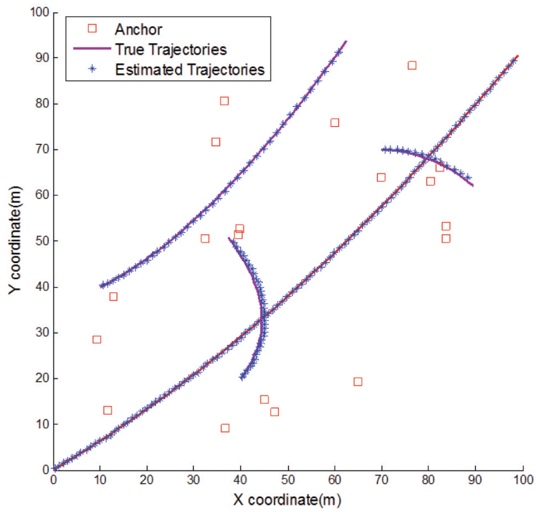


FIGURE 4
True and estimated trajectories

5(b). In these two figures, the moving speed and the accelerated speed of the target on the individual $x$ and $y$ coordinates are $v_x$, $A_x$, $v_y$, and $A_y$, respectively. And the value of $v_x$, $A_x$, $v_y$, and $A_y$ are set to be the same as these set in Figure 4.

Figure 6 shows that the location error between the true and the estimated trajectories of the tracked targets is under 0.7m. Noting that the localization error is also under 0.7m when the direction of curve abrupt changes, which illustrates that our algorithm performs great tolerance to track a curve whose direction abrupt changes. And the average localization error of all curves is under 0.5m, which implies it contributes higher target-state estimate accuracy for the overlapping tracking algorithm. And more smooth trajectories has higher accuracy, which are under 0.4m.

Table 3 reveals the average location error for four kinds of trajectories. As shown, the average location error is below 0.5m in four trajectories. Moreover, the lowest average location error is 0.2188m. Note that trajectories with higher accelerated speed have higher average location error, because the algorithm cannot change moving target's direction effectively. Improving direction recognizable will be our next research.

Figure 7 shows the average energy consumption of our algorithm compared to the traditional CS and the fingerprint algorithms with respect to target numbers. Our algorithm has lower average energy consumption compared to the traditional CS and the fingerprint algorithms. According to

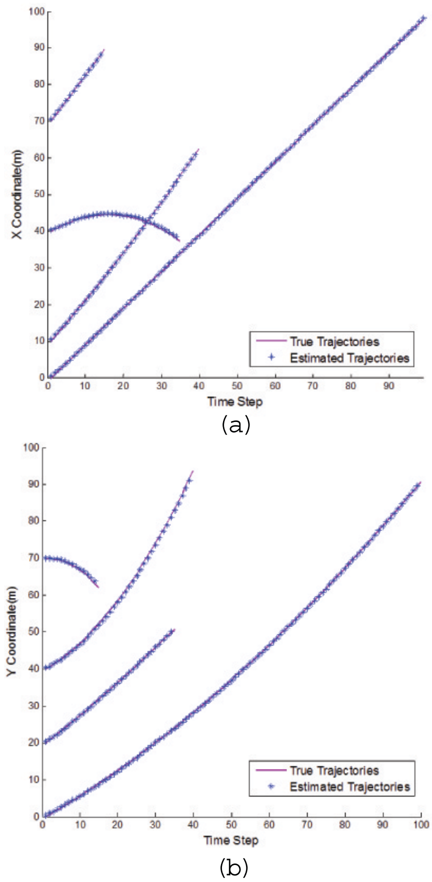


FIGURE 5
x-coordinates(a) and y-coordinates(b) of the true and estimated trajectories

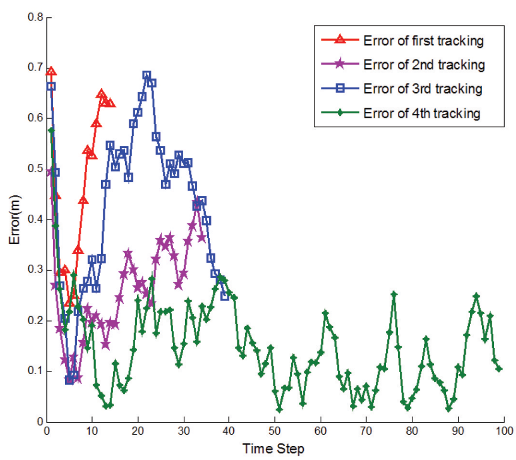


FIGURE 6
The average target-state estimation accuracy performance

| **Trajectory** | $\sigma$ **(m)** |
|---|---|
| 1st TRACKING | 0.4776 |
| 2nd TRACKING | 0.4524 |
| 3rd TRACKING | 0.3690 |
| 4th TRACKING | 0.2188 |

TABLE 3
The average location error for four trajectories

Theorem 1 and Theorem 2, energy consumption depends on the number of anchors and the number of targets. The traditional fingerprint algorithm has the most chosen grids, because it does not choose any grid and all grids in the position region are regarded to be the chosen grids. The average energy consumption of these two algorithms ranges from 10J to 13J. However, our algorithm's average energy consumption is bellow 10J, which exhibits significant energy efficiency.

## 6 CONCLUSION

In this letter, we have proposed a new overlapping tracking algorithm based on compressive sensing. The proposed method can completely eliminate the clustering and cluster matching costs. Moreover, we have proved that the new approach has a threshold for grids in overlapping area, which make the

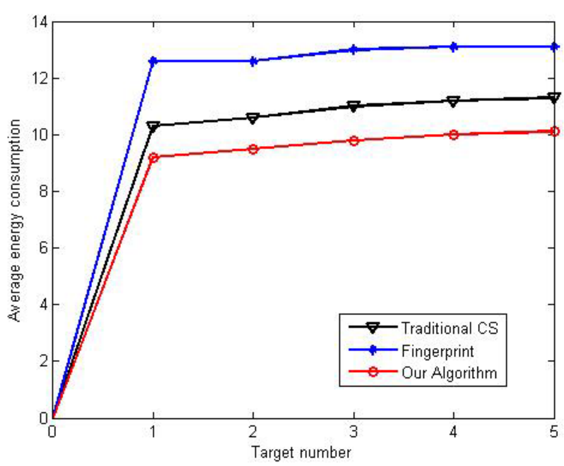


FIGURE 7
Average energy consumption of traditional CS, fingerprint, and our algorithm with respect to target numbers

energy consumption to be a confirmable value. Simulation results show that the proposed method has better performance in tracking the moving target.

The tracking algorithm is based on the weighted compression perception, using overlapping tracking technology to locate unknown nodes. In the process, only the network topology information around the unknown nodes is used, which shows that the localization algorithm has good scalability, and it can be applied to large-scale wireless sensor networks. Our next work is to explore prediction direction recognizable for the tracking and to examine its performance in the real environment.

## ACKNOWLEDGMENTS

This work was partially supported by National Key Technology R&D Program (2012BAD35B06); Program for New Century Excellent Talents in University (NCET-12-0164); National Natural Science Foundation of China (61370094); Natural Science Foundation of Hunan (13JJ1014).

## REFERENCES

[1] Ian F Akyildiz and Ismail H Kasimoglu. (2004). Wireless sensor and actor networks: research challenges. *Ad hoc networks*, 2(4):351–367.

[2] Anthea Wain Sy Au, Chen Feng, Shahrokh Valaee, Sophia Reyes, Sameh Sorour, Samuel N Markowitz, Deborah Gold, Keith Gordon, and Moshe Eizenman. (2013). Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device. *Mobile Computing, IEEE Transactions on*, 12(10):2050–2062.

[3] Paramvir Bahl and Venkata N Padmanabhan. (2000). 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*, volume 2, pages 775–784. IEEE.

[4] Richard G Baraniuk. (2007). Compressive sensing. *IEEE signal processing magazine*, 24(4).

[5] Jeff Boleng and Tracy Camp. (2004). Adaptive location aided mobile ad hoc network routing. In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pages 423–432. IEEE.

[6] Emmanuel J Candès, Justin Romberg, and Terence Tao. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509.

[7] Emmanuel J Candès, Justin Romberg, and Terence Tao. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509.

[8] Emmanuel J Candès and Michael B Wakin. (2008). An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30.

[9] Lina Chen, Binghao Li, Kai Zhao, Chris Rizos, and Zhengqi Zheng. (2013). An improved algorithm to generate a wi-fi fingerprint database for indoor positioning. *Sensors*, 13(8):11085–11096.

[10] David L Donoho. (2006). Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306.

[11] David L Donoho. (2006). Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306.

[12] Chen Feng, Wain Sy Anthea Au, Shahrokh Valaee, and Zhenhui Tan. (2010). Compressive sensing based positioning using rss of wlan access points. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.

[13] Chen Feng, Shahrokh Valaee, and Zhenhui Tan. (2009). Multiple target localization using compressive sensing. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE.

[14] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. (2002). The anatomy of a context-aware application. *Wireless Networks*, 8(2/3):187–197.

[15] Dimitris Milioris, George Tzagkarakis, Artemis Papakonstantinou, Maria Papadopouli, and Panagiotis Tsakalides. (2014). Low-dimensional signal-strength fingerprint-based positioning in wireless lans. *Ad hoc networks*, 12:100–114.

[16] Diba Mirza and Curt Schurgers. (2012). On the performance of range based bayesian tracking. *Communications Letters, IEEE*, 16(7):1129–1132.

[17] Yun Mo, Zhongzhao Zhang, Weixiao Meng, Lin Ma, and Yao Wang. (2014). A spatial division clustering method and low dimensional feature extraction technique based indoor positioning system. *Sensors*, 14(1):1850–1876.

[18] Jan M Rabaey, M Josie Ammer, Julio L da Silva Jr, Danny Patel, and Shad Roundy. (2000). Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7):42–48.

[19] Jiawei Ren, Weimin Jia, Huihua Chen, and Minli Yao. (2012). Unambiguous tracking method for alternative binary offset carrier modulated signals based on dual estimate loop. *Communications Letters, IEEE*, 16(11):1737–1740.

[20] Arash Tabibiazar and Otman Basir. (2011). Compressive sensing indoor localization. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1986–1991. IEEE.