Cooperative Target Tracking and Signal Propagation Learning Using Mobile Sensors

JIAJIE TAN*, The Hong Kong University of Science and Technology, China
WANGKIT WONG, The Hong Kong University of Science and Technology, China
XINYU ZHU, The Hong Kong University of Science and Technology, China
HANG WU, The Hong Kong University of Science and Technology, China
S.-H. GARY CHAN, The Hong Kong University of Science and Technology, China

Target tracking refers to positioning mobile objects over time. The targets may be hospital patients, park visitors, mall shoppers, warehouse assets, etc. We consider a novel cooperative system to track targets, where a target carries low-cost RF tag which not only beacons its ID, but also receives and rebroadcasts beacons of tags within a certain hop away. Mobile sensors, equipped with localization and communication modules, are used to capture and forward the beacons to a server to track the targets. Such multi-hop approach greatly extends the sensing range of the mobile sensors, or equivalently, the beaconing range of the tags, leading to cost-effective deployment.

We propose Mosent, a highly accurate multi-hop system using mobile sensors for target tracking. To account for complex signal propagation in different indoor and outdoor environment, we represent the received signal strength (RSS) matrix overcoming the assumption on propagation model. Given sensor locations, beacons detected by the sensors and RSS matrix, Mosent jointly considers temporal and spatial information to track targets using a modified particle filter. Mosent has an optional, independent and offline module to learn spatial signal propagation in terms of RSS matrix using cooperative mobile sensors equipped with beaconing transceivers. We have implemented Mosent and conducted extensive experiments. Our results show that Mosent achieves 4.37m and 9.46m tracking error in the campus and the shopping mall, respectively, which outperforms other state-of-the-art approaches with significantly lower tracking error (often by more than 30%).

CCS Concepts: • **Networks → Location based services**; • **Human-centered computing**;

Additional Key Words and Phrases: Target tracking, indoor localization, particle filter, mobile sensor, fingerprint

ACM Reference Format:
Jiajie Tan, Wangkit Wong, Xinyu Zhu, Hang Wu, and S.-H. Gary Chan. 2018. Cooperative Target Tracking and Signal Propagation Learning Using Mobile Sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 136 (September 2018), 21 pages. https://doi.org/10.1145/3264946

*This is the corresponding author

Authors' addresses: Jiajie Tan, The Hong Kong University of Science and Technology, Hong Kong, China, jtanad@cse.ust.hk; Wangkit Wong, The Hong Kong University of Science and Technology, Hong Kong, China, wwongaa@cse.ust.hk; Xinyu Zhu, The Hong Kong University of Science and Technology, Hong Kong, China, xzhuah@ust.hk; Hang Wu, The Hong Kong University of Science and Technology, Hong Kong, China, hwuav@cse.ust.hk; S.-H. Gary Chan, The Hong Kong University of Science and Technology, Hong Kong, China, gchan@cse.ust.hk.

2474-9567/2018/9-ART136 $15.00
https://doi.org/10.1145/3264946

1 INTRODUCTION

Target tracking refers to knowing the location of mobile objects over time. The technology has wide applications, e.g., following patients in a hospital, understanding people flow in a mall, finding people in an amusement park, locating assets in a warehouse, etc.

We consider many targets to be tracked in a venue. To track them, an approach is to require each target carrying a device which continuously computes his/her location and reports it to a server. As each device is equipped with a sensing module, a localization module and a communication module, this is costly for a large number of targets. Alternatively, one may install a number of fixed sensors in the venue with the targets carrying light-weight tags. However, to achieve satisfactory accuracy, this incurs substantial investment in sensor infrastructures, especially for large sites.

A much more cost-effective solution is to use mobile sensors carried by personnel such as hospital staff, security guards, mall patrols, etc. These sensors know their own locations according to some localization technique, e.g., GPS [33], Wi-Fi fingerprinting [5, 25, 55] and MIMO TOF [50]. A target carries a low-cost light-weight active tag.[1] A tag has a *transceiver* which can broadcast and receive beacons from other tags. The broadcasted beacons are opportunistically captured by the mobile sensors within their coverage. These sensors then transmit the captured beacons to a server. Based on the beacon information and the sensor locations, target locations are then computed. Note that a sensor, due to its sensing, localization and communication functions, is a computing and networked device and hence is more expensive than a tag. However, as sensors are mobile and may be dynamically deployed to different areas, this approach employs much fewer sensors as compared with the fixed infrastructure. In this paper, we consider the case that both sensors and targets are mobile, though our study is by no means limited to that.

Among all tag signals, we consider radio frequency (RF) such as Wi-Fi and iBeacon, due to cost-effectiveness and deployability. Their transceiving modules are readily available in the market and widely embedded into different mobile devices. These RFs can also be easily detected in terms of received signal strengths (RSS).

The signal of a tag has some certain *coverage*, which is also a range to be sensed. When sensors are sparse with respect to beacon coverage, some targets may not be sensed, let alone tracked. To overcome this, we consider that tags *cooperatively* rebroadcast the received beacons of the other tags within a certain hop limit away. In other words, the hop limit is also known as the lifetime of the beacons. This greatly extends the sensing scope of a sensor and achieves much better trackability in terms of continuity and accuracy.

We show in Figure 1 an example of the system. A sensor keeps listening to the beacons from its neighboring tags (which are the nodes within its sensing range). Upon receiving the beacons, it forwards them to a server along with its location. A tag periodically broadcasts beacons, consisting of its own unique identifier, the remaining lifetime (initialized to the hop limit), and the measured RSS of its neighboring tags. Meanwhile, it also rebroadcasts the beacons from the other tags within a certain hop limit away. It is clear that the hop limit (maximum beacon lifetime) balances the sensing coverage and rebroadcast overhead of the tags.

To model signal propagation, traditional approaches often assume line-of-sight (LOS) scenarios by using some simple fading formulas. While this may be reasonable in open or outdoor space, it does not work well for complex indoor environment due to fading, multipath and shadowing [39]. We hence introduce a more general concept of *RSS matrix* to capture signal propagation between any two locations, where the matrix entry at (i, j) is the RSS received at point j for a transmitter at point i. In such way, RSS matrix does not require any propagation model, and provides spatial information between the transmitter and receiver given a received RSS. We remark that our RSS matrix may be applied in any environment. For the simple environment where signal propagation may be approximated by a fading model, RSS matrix can be straightforwardly generated using that model.

[1] In this paper, we use "target" and "tag" interchangeably.

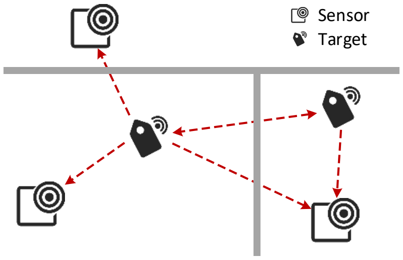


Fig. 1. An example of mobile sensor tracking system. The gray solid lines represent partitions, and the dotted lines with arrows represent beaconing signal transmission.

We propose **Mosent**, a cooperative, cost-effective and highly accurate system using **mo**bile **sen**sors to **t**rack multiple targets. Mosent solves the following *cooperative target tracking* problem: given RSS matrix, the RSS measurements for tags and sensors and sensor locations, how can we efficiently localize the targets over time? We propose a novel and efficient algorithm to address that. We employ a modified particle filter, which constrains the tag locations by means of temporal and spatial information. Instead of tracking a single target at a time, Mosent takes advantage of the spatial relationship among targets as obtained from the RSS matrix, and jointly considers temporal target movements.

Noting that RSS matrix is an important input for Mosent, we further investigate how to obtain it efficiently in a general environment without any model assumption and explicit site survey. Clearly, site survey is too costly (time-consuming and labor-intensive). To this end, Mosent optionally and independently includes an offline module which efficiently learns signal propagation by generating RSS matrix using only mobile sensors. The sensor has a transceiver which emits beacon signal and receives the beacons from its neighboring sensors, and forwards its location and its received beacon information to a server. As server location is uncertain, we consider the following offline *RSS matrix generation* problem: given noisy estimated locations and the RSS received over time of the sensors, how to construct a complete RSS matrix efficiently? We propose leveraging the movement and cooperation of sensors using Rao-blackwellized particle filter [16], a framework widely used for SLAM (Simultaneous Localization and Mapping) [34]. In each time period, we first refine the sensor location according to the signal measurement and current RSS matrix. Based on the better location, the RSS matrix is then updated.

To the best of our knowledge, this is the first work that applies fine-grained model-free approach to the mobile sensor tracking problem. We implement Mosent and conduct extensive experiments in university campus and shopping mall. Our experimental results show that Mosent achieves 4.37m and 9.46m tracking error in the campus and the shopping mall, respectively, which outperforms other state-of-the-art approaches with significantly lower tracking error (often by more than 30%).

The remainder of this paper is organized as follows. We briefly discuss related work in Section 2, followed by system overview and architecture in Section 3. We present the cooperative tracking algorithm in Section 4, and efficient RSS matrix generation in Section 5. Section 6 illustrates the experimental results. We conclude in Section 7.

2 RELATED WORK

We discuss the related work in this section. Traditional RF-based positioning problems can be treated as localizing stationary targets with static sensors. They usually leverage the pre-deployed network infrastructure, such as access points (AP) and iBeacons, to determine the target locations. A number of works, such as tri-lateration [29, 36, 54], employ range-based schemes, which estimate the distance between nodes using signal measurements, e.g., RSS and ToA (Time of Arrival). However, the performance is often hampered by non-line-of-sight (NLOS) measurement. Other researchers focus on the range-free solutions. One of the representative works is the fingerprinting approach [5, 13, 19, 23, 24, 26, 46, 52]. The signal patterns at predefined positions are recorded as unique fingerprints. Then, the location estimation problem becomes searching for the matched fingerprint. These works usually cast the target localization as a per-time-slot estimation problem, hence have not taken the mobility of targets into consideration.

There have been lots of studies considering localizing mobile targets with fixed sensors over time [1, 3, 6, 18, 21, 31, 35, 51, 53, 57]. The problem is usually called as tracking problem. Bayesian filtering is widely applied in these work, of which the basic idea is to combine the signal measurements with the motion of targets. Work in [35] further proposes a trajectory estimation method based on Viterbi map-matching [45] to improve the tracking accuracy with a limited number of sensors. These works often assume the independence of each target to simplify the system. In Mosent, we consider both the temporal and cooperative information among targets to extend the tracking coverage as well as improve the accuracy.

There is another line of research studying on tracking mobile targets with mobile sensors. Based on the location estimation methods, they can be further classified into range-based [12, 14, 32, 40] and range-free [28] methods. A pioneer work [27] employs particle filter to combine target motion model and sensor measurement for target tracking. Other works plan the moving trajectories of mobile sensors for better localization performances [4, 22]. Though inspiring, nevertheless, their performance depends highly on the sensor density and sensor trajectory [20]. To combat such problem, Mosent considers leveraging both cooperation among devices and their temporal information to constraint the targets even out of the sensing coverage.

There have been studies on cooperative localization [11, 37, 43, 44, 47–49], where each node calculates its own position by analyzing the signal distance from its neighbors. Some works compute the relative positions of nodes using MDS (Multi-dimensional Scaling) [8, 15, 30, 38, 41, 42] due to its computational efficiency. However, the requirement of the LOS between nodes hinders their applicability in the complex area. Our work introduces the concept of RSS matrix to fit complex environments. There is another work [56] proposing a range-free localization scheme to address the issues of RSS uncertainty and heterogeneous sensor coverage ranges. Work in [10] utilizes ZigBee as the neighbor-detection sensor to identify the moving crowd cluster, and hence mitigates the localization error of each individual in the cluster. Though promising, these works estimate target locations based on only current network states and do not leverage temporal information of target movements.

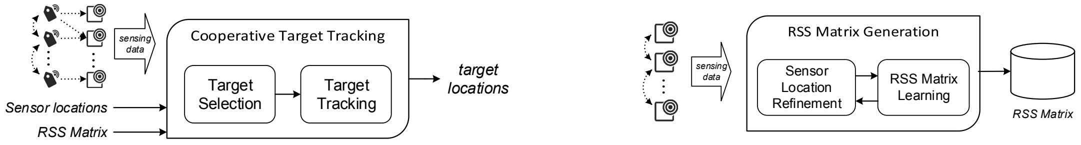
3 SYSTEM OVERVIEW

3.1 RSS Matrix

An RSS matrix captures signal propagation information in the site. In order to form an RSS matrix, we need to discretize the whole site first. Specifically, we scatter a set of *seed* points in the accessible region, and then generate a Voronoi diagram accordingly. We call each polygon partition a *cell*. Thanks to the characteristics of Voronoi diagram that any point inside the cell is closer to the corresponding seed point than others, we can find the cell where a point locates by searching its nearest seed point in the plain.

Given a site partitioned into c cells, we can generate a $c \times c$ RSS matrix $\boldsymbol{M}$. We label every cell with an index. RSS measurement is often assumed having Gaussian noise [2]. We hence store $\boldsymbol{M}_{i,j} = \langle \mu_{\boldsymbol{M}_{i,j}}, \sigma_{\boldsymbol{M}_{i,j}} \rangle$ to represent the RSS distribution, where $\mu_{\boldsymbol{M}_{i,j}}$ and $\sigma_{\boldsymbol{M}_{i,j}}$ are the mean and standard deviation of signal strength emitted from the i-th cell and received at the j-th cell.

One issue in generating the RSS matrix is the so-called device heterogeneity problem. Because different sensors and targets may have different transmission power and antenna gain, the offset among them can distort the tracking accuracy seriously. To address the problem, we follow the approach mentioned in [26]. The idea is



(a) Cooperative Target Tracking.

(b) RSS Matrix Generation.

Fig. 2. The major components of Mosent.

that we select one specific device as the reference device, and adjust the measurement from other devices to the reference level. More specifically, for every device, we measure the RSS at the same distance (say, 1 m) and calculate the offset with the measurement of reference device. Given the offset list and an RSS from the known devices, we perform a linear transformation to adjust the RSS value to the reference device.

3.2 System Framework

Figure 2(a) shows the workflow of cooperative target tracking in Mosent. Each sensor reports the beacons received and their own locations. Besides, an RSS matrix representing the spatial signal propagation of the site is also imported. Given the required data, the tracking module employs a modified particle filter to estimate target locations. For each iteration, the targets are localized sequentially. Specifically, to localize the next target to be estimated, Mosent performs target selection based on the number of neighbors. We localize the target with the highest confidence every time. Later, for tracking targets, the particles are updated based on the motion model and spatial relation to its neighbors. At the end of each iteration, Mosent generates the estimated locations for all the targets.

RSS matrix generation module is designed under the Rao-Blackwellized particle filter (RBPF) framework. We show its workflow in Figure 2(b). The generation algorithm works in an iterative way. Within each iteration, the particles are predicted and updated based on the sensor mobility as well as their estimated locations. Using the mutual RSS measurement, we can further constrain the possible locations of each sensor. According to the refined sensor locations, the RSS matrix is updated over time. We summarize the major notations used throughout the paper in Table 1.

4 COOPERATIVE TARGET TRACKING

We present in this section our novel algorithm to efficiently track multiple targets using mobile sensors and cooperative targets. We first overview the multi-target tracking problems in Section 4.1. We then discuss the details of two major steps, i.e. target selection (Section 4.2) and target tracking (Section 4.3). Finally, we give a toy example of the algorithm in Section 4.4.

4.1 Overview

The cooperative multi-target tracking problem is modeled as estimating the true states of a dynamic system from noisy observations. At discrete time t, the locations of targets are represented by system state $\boldsymbol{x}_t = \{x_t^1, x_t^2, ..., x_t^n\}$, where x_t^i = {x-coordinate, y-coordinate} is the 2-D coordinate of target i. Similarly, the location of sensors are denoted by $\boldsymbol{y}_t = \{y_t^1, y_t^2, ..., y_t^m\}$. During the time from $t-1$ to t, targets are assumed to move according to the given motion model $p(x_t^i|x_{t-1}^i)$, which is the probability density function of target i's current location given previous location x_{t-1}^i.

Table 1. Major symbols used in Mosent.

Notation	Definition
$\boldsymbol{C}$	The set of sensors
$\boldsymbol{D}$	The set of targets
$\boldsymbol{x}_t$	The set of target locations at time t
x_t^i	The location of target i at time t
$\boldsymbol{y}_t$	The set of sensor locations at time t
y_t^j	The location of sensor j at time t
$\boldsymbol{o}_t$	The set of RSS measurements from targets to sensors at time t
o_t^j	The RSS measurements from target to sensor j at time t
$\boldsymbol{s}_t$	The set of RSS measurements among sensors at time t
s_t^j	The RSS measurements from other sensor to sensor j at time t
$\boldsymbol{z}_t$	The set of RSS measurements among targets at time t
z_t^i	The RSS measurements from other target to target i at time t
$\boldsymbol{\psi}_t$	The set of sensor locations reported by external source at time t
ψ_t^j	The location of sensor j reported by external source at time t
$\boldsymbol{M}$	The RSS matrix
$\boldsymbol{M}_{i,j}$	The element with coordinate (i, j) in the RSS matrix

To leverage temporal information, we use sensing data from time 1 to time t (denoted as $\{\boldsymbol{o}_{1:t}, z_{1:t}\}$) and the motion model to estimate the target locations at time t. Therefore, Sequential Importance Resampling (SIR) particle filter is employed to incorporate temporal sensing data for location estimation. The posterior probability distribution of target locations at time $t-1$ is given by $p(\boldsymbol{x}_{t-1}|\boldsymbol{o}_{1:t-1}, z_{1:t-1})$. The observation model $p(\boldsymbol{o}_t, z_t|\boldsymbol{x}_t)$ describes the likelihood of observing sensing result $\{\boldsymbol{o}_t, z_t\}$ given the targets locations $\boldsymbol{x}_t$. We are interested in estimating the joint posterior probability distribution of target locations, i.e. $p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, z_{1:t})$. A particle filter computes the posterior distribution recursively according to

$$p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, z_{1:t}) \propto \underbrace{p(\boldsymbol{o}_t, z_t|\boldsymbol{x}_t)}_{\text{likelihood}} \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \underbrace{p(\boldsymbol{x}_{t-1}|\boldsymbol{o}_{1:t-1}, z_{1:t-1})}_{\text{posterior of previous location}} \,\mathrm{d}\boldsymbol{x}_{t-1}. \tag{1}$$

However, since targets are able to sense with each other, leveraging the mutual sensing data to estimate the joint posterior $p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, z_{1:t})$ incurs unaffordable time complexity. To address this, Mosent devises a modified particle filter to approximate the individual conditional posterior of target location sequentially instead of considering them together at the same time.

We say a sensor is covered by a target if it receives the beacon from the target. Mosent first sorts targets according to their localization confidence, which is the number of sensors covered by the target. Then for each target, we estimate the conditional posterior distribution, and hence the location for each target. By doing so, targets with high confidence are located first and the resultant locations can be used to locate other targets.

In Mosent, the joint posterior distribution of target locations $p(x_t^1, ..., x_t^n|\boldsymbol{o}_{1:t}, z_{1:t})$ at time instance t is represented by a set of particles, which is denoted as $\boldsymbol{P}_t = \cup_{i\in D}\boldsymbol{P}_t^i$, where $\boldsymbol{P}_t^i = \{P_t^{i,[1]}, ..., P_t^{i,[K]}\}$ is the set of particles representing target i's location and $P_t^{i,[k]}$ is the k-th particle of target i. The location distribution of each target is represented by K particles. K is a system parameter which balances the computational complexity and the tracking accuracy, and detailed studies are in Section 6. Each particle is formed by a tuple $\langle x_t^{i,[k]}, w_t^{i,[k]}\rangle$, where

$x_t^{i,[k]}$ is a possible location of target i and weight $w_t^{i,[k]}$ represents how likely the true target location is $x_t^{i,[k]}$. In each iteration, the tracking can be separated into two successive steps: target selection (Section 4.2) and target tracking (Section 4.3), which are detailed in the following.

4.2 Target Selection for Localization

Usually, the number of sensors within the coverage of a target is different. Intuitively, the large the number of covered sensors in the coverage of a target, the more accurate we can estimate the location of the target. Therefore, we sort targets according to the number of sensors within their coverage (ties are broken arbitrarily). With the sorted targets, the joint posterior is approximated as

$$p(x_t^1,\ldots,x_t^n|\boldsymbol{o}_{1:t},\boldsymbol{z}_{1:t}) \approx p(x_t^1|\boldsymbol{o}_{1:t},\boldsymbol{z}_{1:t-1})p(x_t^2|\boldsymbol{o}_{1:t},\boldsymbol{z}_{1:t-1},x_t^1,z_t^1)\ldots\underbrace{p(x_t^n|\boldsymbol{o}_{1:t},\boldsymbol{z}_{1:t-1},x_t^1,\ldots,x_t^{n-1},z_t^1,\ldots,z_t^{n-1})}_{\text{conditional posterior of target } i}, \qquad (2)$$

where $p(x_t^i|\boldsymbol{o}_{1:t},\boldsymbol{z}_{1:t-1},x_t^1,\ldots,x_t^{i-1},z_t^1,\ldots,z_t^{i-1})$ is the approximated conditional posterior distribution of target i. Initially, we see that the computation of posterior distribution of the location of target 1 is approximated by $p(x_t^1|\boldsymbol{o}_{1:t},\boldsymbol{z}_{1:t-1})$ which does not depend on any sensing data at targets at the current time instance. While the computation of posterior of target 2 is approximated by $p(x_t^2|\boldsymbol{o}_{1:t},\boldsymbol{z}_{1:t-1},x_t^1,z_t^1)$, which depends on the the sensing data at target 1 and the location of target 1. Since we do not know the exact location of target 1, we use the estimated location instead. Target locations are estimated according to the sorted order of targets one by one, in the sense that the estimation of latter targets are able to make use of the estimated location and sensing data of previously estimated targets.

4.3 Target Tracking

We first discuss the prediction step in the particle filter. The location distribution of the i-th target at time instance $t-1$ is represented by particles $\{P_t^{i,[1]},\ldots,P_t^{i,[K]}\}$. At time 0, particles are uniformly distributed in the whole accessible area as we do not have any prior knowledge of target locations and a target may be at any location. We consider that we are unaware of the moving speed and direction of the target. However, we do know that its moving distance within time interval must less than some value d_{max}. If target i is at location x_{t-1} in last time instance, at time t, it could be at any location within a circular region with origin x_{t-1} and radius d_{max}. More precisely, the motion model is given by the following probability density function

$$p\big(x_t|x_{t-1}\big) = \begin{cases} \frac{1}{\pi d_{max}^2}, & \text{if the Euclidean distance between } x_{t-1} \text{ and } x_t \text{ is not greater than } d_{max}, \\ 0, & \text{otherwise.} \end{cases} \qquad (3)$$

To predict the next location of target i, we hence move each of the particles of target i to a random position within in the circular region with radius d_{max} centered at the particle. That is, for each particle $P_{t-1}^{i,[k]} = \langle x_{t-1}^{i,[k]}, w_{t-1}^{i,[k]}\rangle$

$$P_{t-1}^{i,[k]} \xrightarrow{\text{motion model}} P_t^{i,[k]} = \langle x_t^{i,[k]}, w_{t-1}^{i,[k]}\rangle, \qquad (4)$$

where $x_t^{i,[k]}$ is the new particle's location.

The weight of each particle is updated according to the observation. Note that before estimating the posterior distribution of target i, we have already estimated the locations of targets $\{0,1,\ldots,i-1\}$. Given the sensing data $\boldsymbol{o}_{1:t}$ at sensors, the sensing data at target $\{0,1,\ldots,i-1\}$ and the the estimated location of target $\{0,1,\ldots,i-1\}$, we are able to measure the likelihood of each particle $P_t^{i,[k]}$ being at the true location. Recall that the signal propagation loss between any two points in the map is given by the RSS matrix $\boldsymbol{M}$. Denote x and y as the locations of a target and a sensor, respectively, and o is the detected RSS at the sensor. Then the likelihood of

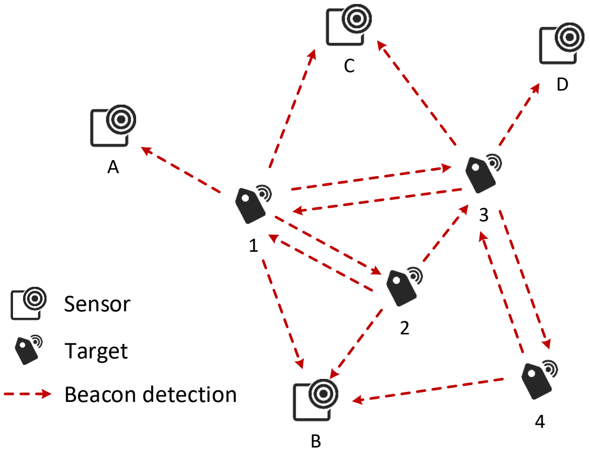


Fig. 3. An illustrative example of cooperative target tracking in Mosent.

RSS measurement $p(o|x, y)$ follows $\mathcal{N}(\mu_{\boldsymbol{M}_{\zeta,\eta}}, \sigma^2_{\boldsymbol{M}_{\zeta,\eta}})$, where ζ and η are the cell indexes of x and y, respectively. Considering that the RSS observed at each sensor/target is independent, the likelihood of observing sensing result $\{\boldsymbol{o}_t, z_t^1, ..., z_t^{i-1}\}$ can be calculated as

$$p(\boldsymbol{o}_t, z_t^1, ..., z_t^{i-1}|x_t^i, x_t^1, ..., x_t^{i-1}, \boldsymbol{y}_t) = \prod_{j\in C} p(o_t^j|x_t^i, y_t^j) \prod_{j=0}^{i-1} p(z_t^j|x_t^i, x_t^j), \tag{5}$$

where $p(o_t^j|x_t^i, y_t^j)$ is the likelihood that the RSS of target i's beacon received at sensor j is o_t^j, and, similarly, $p(z_t^j|x_t^i, x_t^j)$ is the likelihood that the RSS of target i's beacon received at another target j is z_t^j. Note that the likelihood of target i being at location x_t^i only depends on sensing data at all sensors and data at targets whose locations are already estimated.

We update the weight of each particle of target i according to $p(\mathbf{o}_t, z_t^1, ..., z_t^{i-1}|x_t^{i,[k]}, x_t^1, ..., x_t^{i-1}, \boldsymbol{y}_t)$. That is, for each $P_t^{i,[k]} = \langle x_t^{i,[k]}, w_{t-1}^{i,[k]}\rangle$,

$$P_t^{i,[k]} \xrightarrow{\text{update}} \langle x_t^{i,[k]}, w_t^{i,[k]}\rangle, \text{ where } w_t^{i,[k]} \propto p(\mathbf{o}_t, z_t^1, ..., z_t^{i-1}|x_t^{i,[k]}, x_t^1, ..., x_t^{i-1}, \boldsymbol{y}_t). \tag{6}$$

Then, the weight of particles are normalized to satisfy $\sum_k w_t^{i,[k]} = 1$.

The location of target i at time t is estimated using the expectation

$$\hat{x}_t^i = \sum_k x_t^{i,[k]} w_t^{i,[k]}. \tag{7}$$

It is applied to address the degeneracy issue of SIR particle filter. The basic idea of resampling is to eliminate trajectories that have small normalized importance weights and to concentrate upon trajectories with large weights. Resampling is done with the following: 1) Draw K particle samples from all particles, each of which is drawn with the probability as its weight. Replace $\boldsymbol{P}_t^i$ with the drawn particles. 2) Set the weight of each particle to be $1/K$.

ALGORITHM 1: Efficient RSS Matrix Generation

```
Initialize particles.
while there is new sensing data collection do
    foreach particle do
        Predict sensor locations through motion model.
    end
    foreach particle do
        Estimate local RSS matrix in each particle accoridng to Equations 15 or 16.
    end
    Update weights for particles according to Equation 11.
    Yield the sensor locations according to Equation 14.
    Yield the RSS matrix according to Equation 17.
    Resample particles.
end
```

4.4 An Illustrative Example

We illustrate in Figure 3 a toy example to show how the tracking algorithm works. There is a typical sensor network consisting of 4 sensors and 4 targets. During time from $t-1$ to t, targets broadcast beacons asynchronously. Dashed arrows indicate the reception of beacons at nodes. To estimate target locations at time t jointly, we first sort targets according to the number of covered sensors. Clearly, the sorted target sort would be $\{1, 3, 2, 4\}$. Then, we locate target according to the sorted order. We first estimate posterior $p(x_t^1|\boldsymbol{o}_{1:t})$ and estimate the location of Target 1. Usually, we could locate the position of Target 1 without the information of other targets as there are 3 sensors sensing the beacon of Target 1. Subsequently, we estimate the location of Target 3, 2 and 4, respectively. When estimating the location of Target 2, we have already estimated the location of Target 1 and 3. Therefore, we are able to make use of Target 1 and 3 to estimate the location of Target 2. However, if we estimate the location of Target 2 first, we can only use sensing data at sensor B to make an estimation. We see that depending on the location of previously estimated locations, Mosent has the potential to improve estimation accuracy.

5 EFFICIENT RSS MATRIX GENERATION

We present in this section how to generate the RSS matrix under noisy sensor location efficiently. This module acts as a plug-in for Mosent. The generated RSS matrix can be used as the input for more accurate cooperative tracking. We give the brief overview and preliminary about Rao-Blackwellized particle filter (RBPF) in Section 5.1, and then discuss the sensor location refinement (Section 5.2) and RSS matrix learning (Section 5.3).

5.1 Overview

In order to generate the RSS matrix, we require the accurate sensor locations. However, sensor locations are usually noisy in reality, and RSS matrix can help us reduce such error to some extent. To solve this *chicken-or-egg* problem, we consider estimating the joint posterior

$$p(\boldsymbol{y}_{0:t}, \boldsymbol{M}|\boldsymbol{s}_{1:t}, \boldsymbol{\psi}_{1:t}), \tag{8}$$

where, following the notations in Section 3, $\boldsymbol{y}_{0:t}$ is the sequence of sensor locations from time 0 to t, $\boldsymbol{M}$ is the RSS Matrix, $\boldsymbol{s}_{1:t}$ are the sequence of mutual measurements among sensors, and $\boldsymbol{\psi}_{1:t}$ is the sequence of noisy location estimation of sensors provided by external sources.

To take the advantage of temporal and spatial information, Monte Carlo filtering approaches can be employed again for the problem. However, they cannot estimate Equation 8 efficiently as the existence of the RSS matrix in

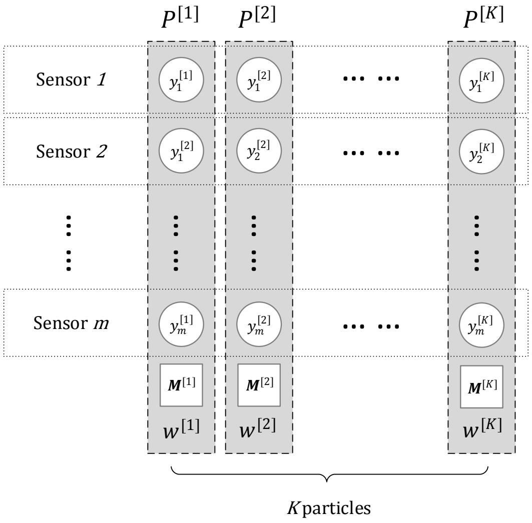


Fig. 4. An illustration of particles structure in the module. A gray column represents a particle. Each particle contains a set of sensors' locations, an estimation of RSS matrix and an associated weight.

the posterior increases the dimensionality of the estimation space. Note that Equation 8 can be factorized as

$$p(\boldsymbol{y}_{0:t}, \boldsymbol{M}|s_{1:t}, \boldsymbol{\psi}_{1:t}) = p(\boldsymbol{y}_{0:t}|s_{1:t}, \boldsymbol{\psi}_{1:t})\, p(\boldsymbol{M}|\mathrm{y}_{0:t}, s_{1:t}, \boldsymbol{\psi}_{1:t}) = \underbrace{p(\boldsymbol{y}_{0:t}|s_{1:t}, \boldsymbol{\psi}_{1:t})}_{\text{sensor trajectory posterior}}\ \underbrace{p(\boldsymbol{M}|\boldsymbol{y}_{0:t}, \mathrm{s}_{1:t})}_{\text{RSS matrix posterior}}. \tag{9}$$

The factorization decomposes the joint posterior estimation into two separated problems, i.e. sensor trajectory estimation and RSS matrix estimation based on the sensor locations. Therefore, we realize it by employing a modified RBPF [16], which has been successfully applied in SLAM problems [34]. We show the key steps in Algorithm 1, which are detailed in the following subsections.

5.2 Refining Sensor Locations

We first estimate the sensor trajectory posterior $p(\boldsymbol{y}_{0:t}|s_{1:t}, \boldsymbol{\psi}_{1:t})$ in Equation 9. Further notice that we can factorize the posterior as

$$\begin{aligned} p(\boldsymbol{y}_{0:t}|s_{1:t}, \boldsymbol{\psi}_{1:t}) &\propto p(s_t, \boldsymbol{\psi}_t|\mathrm{y}_t)\, p(\boldsymbol{y}_{0:t}|s_{1:t-1}, \boldsymbol{\psi}_{1:t-1}) \\ &= p(\boldsymbol{y}_{0:t}|s_{1:t-1}, \boldsymbol{\psi}_{1:t-1})\, p(\boldsymbol{\psi}_t|\boldsymbol{y}_t)\, p(s_t|\boldsymbol{y}_t), \end{aligned} \tag{10}$$

where $p(\boldsymbol{y}_{0:t}|s_{0:t-1}, \boldsymbol{\psi}_{0:t-1})$ is the prediction of current sensor trajectory based on history information, $p(\boldsymbol{\psi}_t|\boldsymbol{y}_t)$ is the likelihood of estimated sensor locations, and $p(s_t|\boldsymbol{y}_t)$ is the likelihood of mutual sensing. Hence we can draw samples from the motion model first, and then update the weight according to the two types of likelihood above.

As the sensor estimation based on external approach is independent with each other, we have $p(\boldsymbol{\psi}_t|\boldsymbol{y}_t) = \prod_{j=1}^m p(\psi_t^j|y_t^j)$. By the central limit theorem [17], we further assume the estimated sensor location follows the zero-mean multivariate Gaussian distribution, i.e. $p(\psi_t^j|y_t^j) \sim \mathcal{N}(y_t^j, \Sigma_{Sj})$, where j is the sensor index, and the covariance matrix Σ_{Sj} of sensor location is a diagonal matrix whose elements on the main diagonal equal to σ_S^2. The value of σ_S relates to the choice of external sensor localization techniques and its accuracy.

We show in Figure 4 the structure of particles applied in RBPF. Each particle represents a combination of the potential locations of all the sensors. Thus every particle contains m locations for m sensors. As the RSS matrix is unknown, each particle also includes a local RSS matrix following the RBPF. We will discuss the estimation of RSS matrix in Section 5.3. We denote $P_t^{[k]}$ as the k-th particle at time t. Then we have $P_t^{[k]} = \langle \boldsymbol{y}_t^{[k]}, \boldsymbol{M}_t^{[k]}, w_t^{[k]} \rangle$, where $\boldsymbol{y}_t^{[k]}$ is the "guess" of sensor locations in the particle, $\boldsymbol{M}_t^{[k]}$ is the local RSS matrix estimated in the particle, and $w_t^{[k]}$ is the weight of the particle.

In the prediction stage, we predict the particles state based on previous estimation. Considering that the mobility of each sensor is independent, we can predict particle states by predicting each sensor location individually. Here we use the same motion model mentioned in Equation 3.

In the updating stage, we update the weight $w_t^{[k]}$ of each particle. The weight represents the belief on each particle. Higher weight indicates that the paricle is more likely to be chosen. To compute $w_t^{[k]}$, we derive it as follows:

$$\begin{aligned} w_t^{[k]} &\propto \frac{p(\boldsymbol{y}_{0:t}^{[k]}|\boldsymbol{s}_{1:t}, \boldsymbol{\psi}_{1:t})}{p(\boldsymbol{y}_{0:t}^{[k]}|\boldsymbol{s}_{1:t-1}, \boldsymbol{\psi}_{1:t-1})} \\ &\propto \frac{p(\boldsymbol{s}_t, \boldsymbol{\psi}_t|\boldsymbol{y}_{0:t}^{[k]}, \boldsymbol{s}_{1:t-1}, \boldsymbol{\psi}_{1:t-1})\, p(\boldsymbol{y}_{0:t}^{[k]}|\boldsymbol{s}_{1:t-1}, \boldsymbol{\psi}_{1:t-1})}{p(\boldsymbol{y}_{0:t}^{[k]}|\boldsymbol{s}_{1:t-1}, \boldsymbol{\psi}_{1:t-1})} \\ &= p(\boldsymbol{s}_t, \boldsymbol{\psi}_t|\boldsymbol{y}_{0:t}^{[k]}, \boldsymbol{s}_{1:t-1}, \boldsymbol{\psi}_{1:t-1}) \\ &= p(\boldsymbol{\psi}_t|\boldsymbol{y}_t^{[k]})\, p(\boldsymbol{s}_t|\boldsymbol{y}_{0:t}^{[k]}, \boldsymbol{s}_{1:t-1}). \end{aligned} \tag{11}$$

Note that the first term in Equation 11 follows the sensor location observation model. To further compute the second term $p(\boldsymbol{s}_t|\boldsymbol{y}_{0:t}^{[k]}, \boldsymbol{s}_{1:t-1})$, we have

$$\begin{aligned} p(\boldsymbol{s}_t|\boldsymbol{y}_{0:t}^{[k]}, \boldsymbol{s}_{1:t-1}) &= \int p(\boldsymbol{s}_t, \boldsymbol{M}|\boldsymbol{y}_{0:t}^{[k]}, \boldsymbol{s}_{1:t-1})\, \mathrm{d}\boldsymbol{M} \\ &= \int p(\boldsymbol{s}_t|\boldsymbol{M}, \boldsymbol{y}_{0:t}^{[k]})\, p(\boldsymbol{M}|\boldsymbol{y}_{0:t-1}^{[k]}, \boldsymbol{s}_{1:t-1})\, \mathrm{d}\boldsymbol{M}. \end{aligned} \tag{12}$$

Equation 12 can be approximated as Gaussian density [7, 34]:

$$p(\boldsymbol{s}_t|\boldsymbol{y}_{0:t}^{[k]}, \boldsymbol{s}_{1:t-1}) \propto \prod_{s_t^{i,j} \in \boldsymbol{s}_t} \exp\big(-\frac{(s_t^{i,j} - \hat{s}_t^{i,j})^2}{2\phi_{i,j}^2}\big), \tag{13}$$

where $s_t^{i,j}$ is the RSS measurement by the device in i-th cell and received in j-th cell, and $\hat{s}_t^{i,j}$ is the expected RSS measurement, i.e. $\hat{s}_t^{i,j} = \mu_{\boldsymbol{M}_{i,j}}$. The variance $\phi_{i,j}^2$ in Equation 13 represents the uncertainty on estimation, which can be computed as the summation of the variance of $p(s_t^{i,j}|\boldsymbol{M}_{i,j}, \boldsymbol{y}_{0:t}^{[k]})$ and the variance $p(\boldsymbol{M}_{i,j}|\boldsymbol{y}_{0:t-1}^{[k]}, \boldsymbol{s}_{1:t-1})$ [7]. We can see that $p(\boldsymbol{s}_t|\boldsymbol{y}_t^{[k]}, \boldsymbol{s}_{1:t-1})$ actually indicates how likely the particle matches to the current RSS matrix estimation. Note that the weight of particles should be normalized to satisfy $\sum_k w_t^{i,[k]} = 1$.

Based on the updated weight of each particle, we can estimate the i-th sensor location $\hat{y}_t^i$ by computing the expectation of sensor location. Mathematically, we have

$$\hat{y}_t^i = \sum_k w_t^{[k]}\, y_t^{i,[k]}. \tag{14}$$

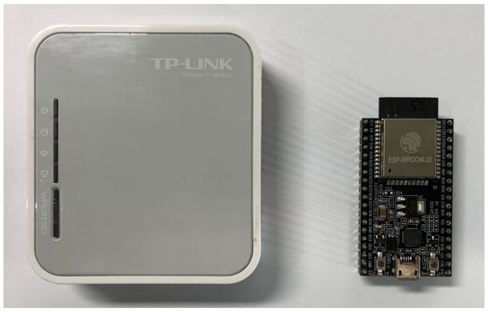


Fig. 5. The prototypes of sensors and tags in the experiment. The left one is the sensor running on TP-Link MR3020, and the right one is the tag implemented on the platform of Espressif ESP32.

5.3 RSS Matrix Learning

Under the framework of RBPF, we estimate the RSS matrix at the same time of refining the sensor locations. The estimation is conducted by recursively updating the posterior of RSS matrix $p(\boldsymbol{M}|\boldsymbol{y}_{0:t}, \boldsymbol{s}_{1:t})$.

For any element $\boldsymbol{M}_{i,j}$ in the RSS Matrix $\boldsymbol{M}$, if there is a measurement at time t where the transmitting sensor is located in l_i and the receiver sensor is located at l_j, then element $\boldsymbol{M}_{i,j}$ need to be updated. Here we use $\boldsymbol{M}_t^{i,j}$ to represent the element with coordinate (i, j) in the RSS matrix $\boldsymbol{M}$ of time t. For the simplicity of the symbol, let $\boldsymbol{M}_t^{i,j} \sim \mathcal{N}(\mu_t, \sigma_t^2)$, and s_t be the RSS measurement for $\boldsymbol{M}_{i,j}$ at time t. As $p(\mu_t|s_{1:t}) \propto p(s_t|\mu_t)\, p(\mu_t|s_{1:t-1})$ where both the prior $p(\mu_t|s_{1:t-1})$ and the likelihood function $p(s_t|\mu_t)$ are Gaussian variables, the posterior $p(\mu_t|s_{1:t})$ can be updated as Gaussian with

$$\mu_t = \frac{1}{B}\left(\frac{\mu_{t-1}}{\sigma_{t-1}^2} + \frac{s_t}{\sigma^2}\right), \text{ and } \sigma_t^2 = \frac{1}{B}, \tag{15}$$

where $B = 1/\sigma_{t-1}^2 + 1/\sigma^2$ and σ^2 is the measurement variance [7]. On the other hand, if the matrix element does not need to be updated, we simply leave it unchanged:

$$p(\mu_t|s_{1:t}) = p(\mu_{t-1}|s_{1:t-1}). \tag{16}$$

In order to yield the RSS matrix, we take the weighted average as the approximation (we hide the subscript t for simplicity):

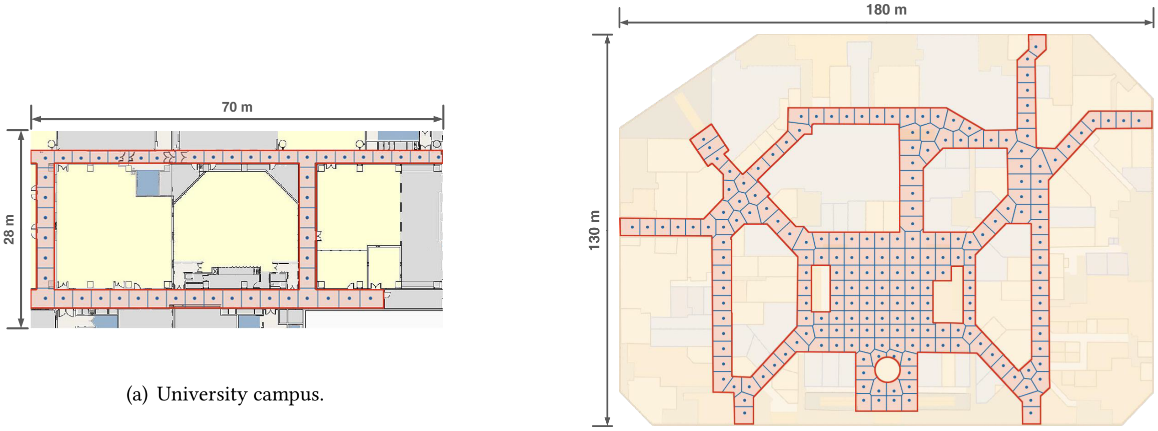
$$\boldsymbol{M}_{i,j} = \sum_k w_i^{[k]}\, \boldsymbol{M}_{i,j}^{[k]}. \tag{17}$$

6 EXPERIMENTAL EVALUATION AND ILLUSTRATIVE RESULTS

This section presents the experimental evaluation of Mosent. We first discuss the experiment settings in Section 6.1, and then demonstrate the experimental results in Section 6.2.

6.1 Experiment Settings and Methodologies

We implement the sensors on TP-Link MR3020 portable router running an open source embedded operating system OpenWrt 15.05. We choose the Espressif ESP32 as the platform of the active tags because it provides convenient APIs to manipulate IEEE 802.11 beacon frame. Figure 5 shows the prototype of Mosent. The cooperation used in Mosent is based on the beacon frame broadcasting so that no network connection is required. The beacon frame carries our customized data, e.g. hop limits, current hop, RSS measurement, etc., in the fields of *Element Information*. The beaconing frequency is around 1 s. Both sensors and targets operate on the 2.4 GHz wireless



(a) University campus.

(b) Shopping mall.

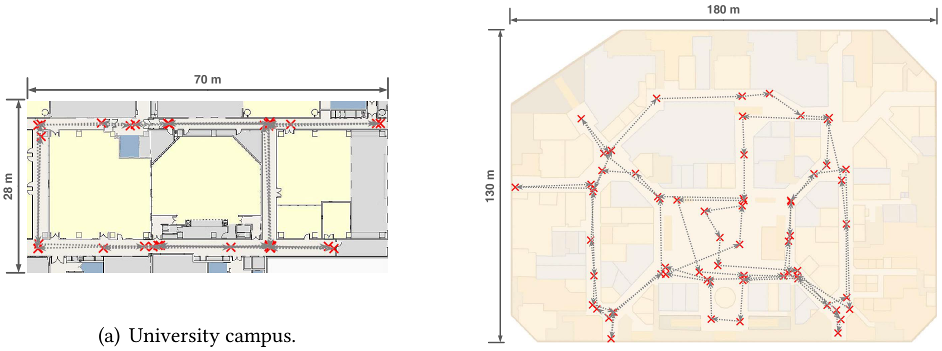
Fig. 6. The floor plans and cell partitions of the two experimental sites. The area in red represents the accessible region. The blue line segments partition the cells, in which the blue dots are the seed points.

network. Channel 1 is used in the experiments. The transmission power of sensors is set to 10 dBm, and the tag is set to Level 0 (as no precise power setting is provided). Sensors connect to the server through a wireless local network (WLAN). We also implement our server on a PC with 3.6 GHz processor and 16 GB RAM running Ubuntu 16.04.

We evaluated the system performance in two different sites. The first one is in the office building of our university campus (Figure 6(a)). The area is around 70 m × 28 m with corridors, wall partitions, and rooms. Such environment is similar to the application environments like hospital, nursing home, company office, etc. The other site is chosen in a large shopping mall (Figure 6(b)) which consists of multiple corridors with complicated connections and large open area. The total area is about 180 m × 130 m. We choose these two sites because both of them are the representative application scenarios which are suitable for mobile sensor based tracking system. We conduct both experiments in the normal working hours. There are other wireless infrastructures in the area and passengers walking through the site during the experiment, which represent the real world situation.

Following the discretization process described in Section 3.1, we partition the accessible region in the both sites. The partitions have been shown in Figure 6. The area of each cell in the campus is about 3 m × 3 m, and there are 51 cells in total. In the site of shopping mall, the distance between the neighboring seed points is around 5 m, and the total number of cells is 214.

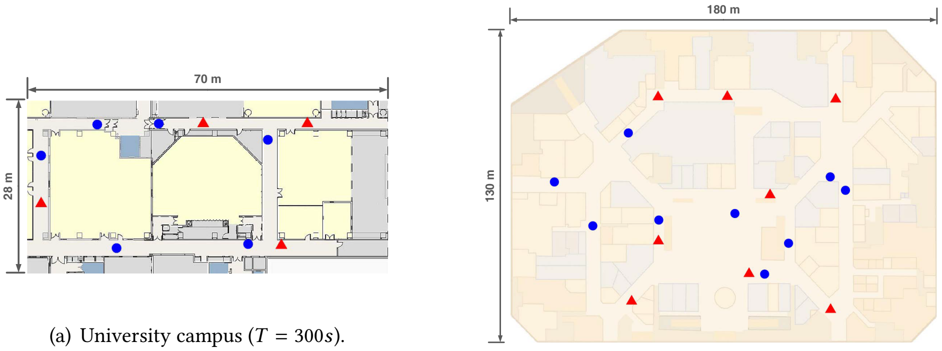
We recruit multiple volunteers to conduct the experiments. Each participant holding a device (either a sensor or a tag) walks in the accessible region of the sites (labeled with red color in Figure 6). There is no pre-planned path for them. They are allowed to walk at their own pace, and even stop at any position. Figures 7(a) and 7(b) show examples of the trajectories of one participant in the two sites, respectively. 10 people in total participate in the experiment of the campus. 4 users carry the sensors and the rest carry the tags. In the shopping mall scenario, 17 participants involve. We assign 8 of them carrying sensors and 9 of them carrying tags. In order to have an intuitive understanding on the distribution and density of devices, we visualize in Figures 8(a) and 8(b) the locations of sensors and targets at a time instance.



(a) University campus.

(b) Shopping mall.

Fig. 7. An example of walking trajectories and labeled locations. The red cross markers are the labeled positions during the experiment. The gray dot lines represent the trajectories of a participant, and the corresponding arrow indicates the walking direction.



(a) University campus ($T = 300s$).

(b) Shopping mall ($T = 300s$).

Fig. 8. An example of sensors/targets positions at a certain time instance. The red triangles and blue dots represent the positions of sensors and targets, respectively.

We also collect the true walking trajectories (ground-truth locations) for evaluating the tracking accuracy. Participants are required to label their positions when he/she encounters some critical locations such as turning corners and landmarks. We design an App on mobile phones for the ease of labeling. The location label is denoted by a tuple $< t, l >$, where t is the timestamp, and l is the coordinate of current location. In Figures 7(a) and 7(b), the red cross markers are the positions labeled by users during the whole experiment. In addition, we record the steps of participants through a software-implemented pedometer [9], which indicates their walking status (walk/stop) and speed. Those labeled positions, as well as the step counts, are used together to recover their true walking trajectories. To be specific, for any two consecutive location labels $< t_0, l_0 >$ and $< t_1, l_1 >$, the position l

of the participant at time t can be computed by

$$l = l_0 + (t - t_0 - \Delta t_s)v, \tag{18}$$

where $v = (l_1 - l_0)/(t_1 - t_0 - t_s)$, t_s is the non-walking time between t_0 and t_1, and Δt_s is the non-walking period from t_0 to t. Notice that the collected locations are only used for the purpose of performance validation.

At each site, we conduct the RSS matrix generating stage and tracking stage separately. We conduct the experiment in the campus for 20 minutes. We generate the RSS matrix for 10 minutes and then test the tracking accuracy for another 10 minutes. Considering the larger scale of the shopping mall, we spend more time there for fully testing. Specifically, 25 minutes are scheduled for generating the RSS matrix, and 20 minutes for tracking evaluation.

We implement the Wi-Fi based fingerprint indoor localization system [5] for providing the location estimation of sensors. The fingerprint database is collected on a different day to avoid correlation. The mean localization error is 3.74 m in the campus and 6.17 m in the mall.

We also collect the ground truth of signal propagation for validating the performance of RSS matrix learning. Two surveyors, as a group, holding sensors, stand at two different positions to record the RSS at each side. For each pair of positions, we collect RSS for 15 seconds to mitigate the influence of signal fluctuation. The total number of pairs in the campus is 1275. Due to the time and resource limit, we only collect the ground truth RSS matrix for the experiment in the campus.

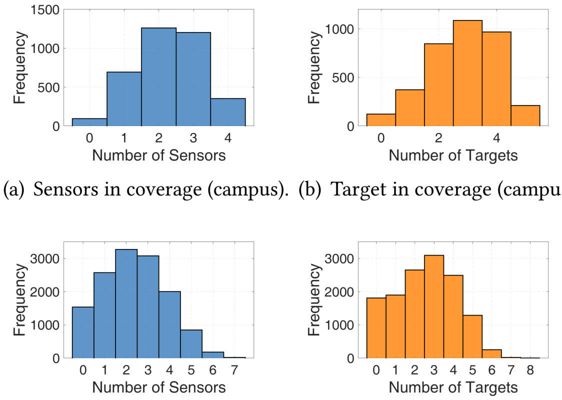
We compare the performance of Mosent with the following state-of-the-art target tracking schemes:

- *Proximity* reports the location of sensors with the maximum RSS as the target locations. This approach can be treated as the baseline for system performance.
- *Monte Carlo Localization using mobile sensors (MCL)* [27] considers the mobile sensor based tracking without distance estimation. It only uses a binary indicator to identify whether the target is within the sensing range or not. It also applies a particle filter to leverage the temporal information.
- *MDS-based collaborative localization (MDS)* [30, 41, 42] takes the advantage of cooperation among nodes. It first maps nodes to relative positions based on the eigenvalue decomposition of the distance matrix. Then, linear operations such as rotation and translation are performed based on the known sensor locations. To implement it, we use the path loss model to estimate the distance among the directed linked sensors/targets.

We evaluate the performance of Mosent using the following metrics:

- *Tracking error*: Let $\hat{l}_t$ be the target's estimated location of target at time instance t, and l_t be the true location of the same time. Note that l_t is calculated by Equation 18 according to the ground-truth location labeled by the user. The tracking error is defined as the Euclidean distance between the estimated location $\hat{l}_t$ and its true location l_t, i.e. $e_t = ||l_t - \hat{l}_t||_2$. Particularly, we define the mean tracking error of a trajectory as $\bar{e} = \sum_t ||l_t - \hat{l}_t||_2/\eta$, where η is the number of location estimation in the trajectory. We evaluate the tracking system performance in terms of tracking error.
- *RSS error*: Let $\hat{\boldsymbol{M}}$ be the estimated RSS matrix, and $\boldsymbol{M}$ be the ground truth. For the entry $\boldsymbol{M}_{i,j}$, the RSS error is defined as $e_{\boldsymbol{M}_{i,j}} = |\mu_{\boldsymbol{M}_{i,j}} - \mu_{\hat{\boldsymbol{M}}_{i,j}}|$, where $\mu_{\boldsymbol{M}_{i,j}}$ is the expectation value of the random variable $\boldsymbol{M}_{i,j}$. For the entry without any update, we treat its value as −90 dBm which is the minimum detectable RSS. The RSS error represents the correctness of RSS matrix generation.

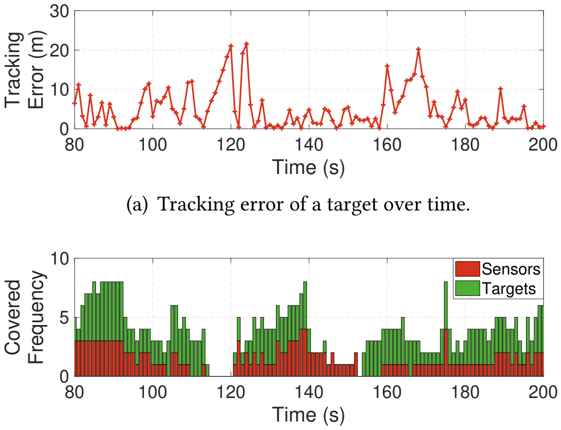
In the experimental study, unless otherwise specified, we use the following default parameters as the baseline settings: the number of particles for tracking is 125, the hop limit is 2, and the number of particles used in RSS matrix generation is 200.



(a) Sensors in coverage (campus). (b) Target in coverage (campus).

(c) Sensor in coverage (mall). (d) Target in coverage (mall).

Fig. 9. The histograms of the number of sensors/targets in the coverage of targets in different sites.



(a) Tracking error of a target over time.

(b) The number of sensor/traget in the target coverage over time.

Fig. 10. The tracking error and the number of devices covered of tag No.3 from time 80 s to 200 s (campus).

6.2 Illustrative Results

Figure 9 shows the histograms of sensor/target coverage in both sites. Each distribution indicates the number of cases in which sensors/targets are able to sense the targets. Due to the limited area in the campus site, we can see that, in the most cases, 1 - 3 sensors (Figure 9(a)) and 2 - 4 targets (Figure 9(b)) can capture the beacon from targets. However, the scenario in the shopping mall is more challenging. There are 11.4% (Figure 9(c)) and 14.4% (Figure 9(d)) of all the cases in which no sensor or other target can ever detect the target, respectively.

To illustrate the relation of tracking error and the number of devices in the coverage, we show a period of data from tag No.3 as an example. Figure 10 visualizes the tracking error and the quantities of devices in the coverage of the particular tag. We can see from the figure that if many sensors or targets are within the coverage, the tracking error is usually low. There are two interesting observations can be also found from the figure. Notice that there is no device in the coverage at all within the time range from 114 s to 120 s. As a result, the tracking error increases as we lose all the information about the target. In another time range from 145 s to 160 s, only limited amount of sensors/tags sense the particular tag, which is not sufficient to guarantee a good tracking accuracy in traditional approaches. However, since Mosent further considers mobilities, it can still achieve satisfying performance.

Figures 11 and 12 show the impact of the number of particles against the tracking accuracy. We can see that a large number of particles (usually more than 75) bring significant improvement on tracking accuracy (over 10% and 40% in the campus and mall, respectively). The reason is that more particles can have a more accurate representation of the posterior of target positions. However, when the amount of particles further increases, the improvement is limited. This is because Mosent uses a discrete representation on the signal propagation (i.e. the RSS matrix), and hence different particles may be located at the same cell with the same signal propagation information. On the other hand, a large number of particles requires more computational cost. To balance the robustness and accuracy, we recommend using 125 particles for Mosent.

In order to study the impact of hop limit on the tracking accuracy, we store all the sensing data within at most 5 hops and label the hop counts for each set of sensed data during the experiment. We can thus filter the data within desired hop limit to investigate its impact. Figures 13 and 14 show the tracking error versus hop limit in the campus and shopping mall, respectively. We notice that, in the campus, the increment of hop limit does not

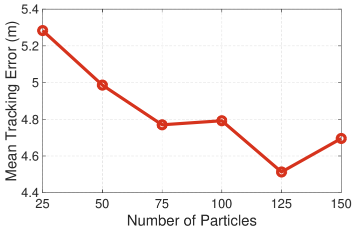


Fig. 11. Tracking error versus the number of particles in the campus.

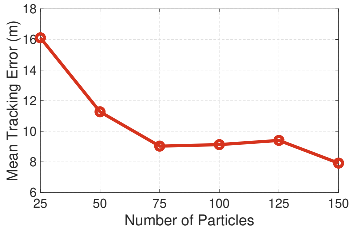


Fig. 12. Tracking error versus the number of particles in the mall.

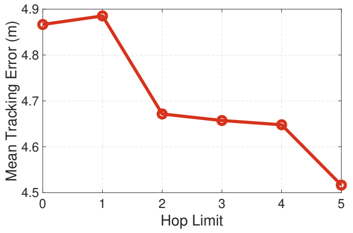


Fig. 13. Tracking error versus the hop limit in the campus.

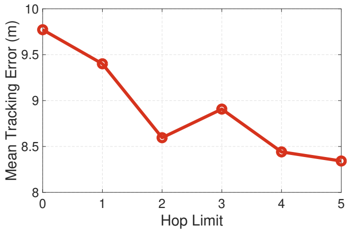


Fig. 14. Tracking error versus the hop limit in the mall.

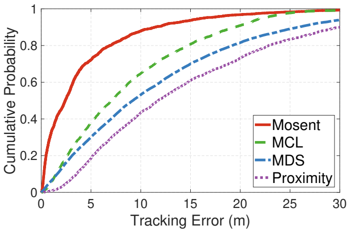


Fig. 15. The CDF of tracking error with different schemes in the campus.

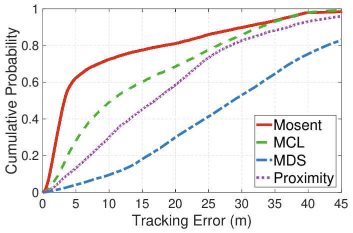


Fig. 16. The CDF of tracking error with different schemes in the mall.

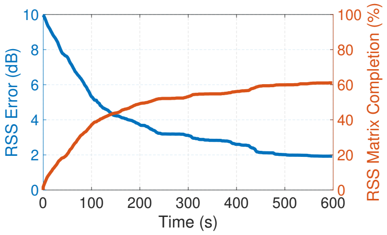


Fig. 17. The RSS error and RSS matrix completion over time in the campus.

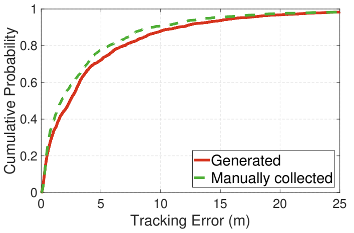


Fig. 18. The CDF of tracking error with different RSS matrix in the campus.

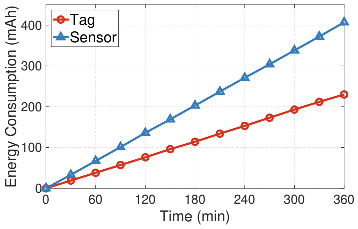


Fig. 19. The energy consumption versus time.

improve the system performance too much, because the number of covered sensors has already been adequate as shown in Figure 9(a). In the mall site, with the increasing hop limit, we have a more comprehensive view on the collaborated relationship of targets. Hence, the error will be reduced by more than 12% when hop limit is larger than 2 compared with no target cooperation.

Figure 15 shows the CDF of tracking error of different algorithms in the scenario of the campus. In such complex indoor environment, Mosent can outperform the state-of-the-art schemes by more than 40%. The mean error of Mosent is 4.37 m. Both MCL and MDS-MAP suffer from the unpredictable signal propagation characteristics. Hence, they cannot achieve satisfying results.

Figure 16 shows the overall comparison results in the shopping mall, where tracking is rather challenging because of the huge area and low density of sensors/targets. The performance of Mosent is still better than other systems. The mean error of Mosent is 9.46 m, and the 50th percentile is 3.27 m. We notice that the long error tail exists in the CDF. Such error mainly comes from the case when no device can sense the target (see Figures 9(c) and 9(d)). On the other hand, MDS utilizes the shortest path distance to approximate the actual distance among devices. However, due to the extremely low sensor/target densities, such scheme fails to compute the accurate distance matrix. Hence it produces the highest error. Comparing with other schemes, Mosent can still cutting the error by 30%.

We investigate the RSS matrix generation performance in terms of the RSS error and the RSS matrix completion percentage. Figure 17 demonstrates the two metrics over time in the campus. We can see from the figure that the RSS matrix completion percentage keeps increasing and eventually converges at 60%. The rest entries cannot be learned because no detectable signal can transmit from one side and received at the other. Meanwhile, the mean RSS error also reduces with the matrix completion. After learning for 10 minutes, the mean RSS error reduces to 1.93 dB. This indicates that the matrix can gradually reflect the signal propagation in reality.

Figure 18 shows the CDF of tracking error with different RSS matrix. We compare the effectiveness of generated RSS matrix with the manually collected one. In the figure, the generated CDF curve is very close to the one by manual collection. The mean tracking error with generated RSS matrix is 4.37 m, while the error with manually collected matrix is 3.78 m.

We conduct another experiment to monitor the power consumption of both tag and sensor prototypes. Both devices operate under 5 V DC. No energy saving mode is configured. We use a USB-based power meter (UT-658B) to keep measuring the power consumption for 6 hours. Figure 19 shows the result over time. The power usage is quite linear during the whole test. The energy consumption of sensors is 1.77x to that of the tags. This is mainly due to the overload of network communication and self-localization on the sensors. Our proposed scheme leverages the tags with low energy consumption to achieve the better accuracy than systems using sensors only.

During our experiments, the running time for tracking all targets in each iteration is less than 1 second, which ensures fluid user experience during the deployment in real time and is competitive with other systems.

7 CONCLUSION

We have considered a novel multi-hop system to track targets using mobile sensors. The targets cooperatively relay each other beacons according to some hop limit. The sensors, with localization and communication modules, detect the beacons and forward them to a server to compute the target locations over time. Such multi-hop cooperative approach greatly extends sensing scopes of sensors and reduces its deployment cost.

We propose and study Mosent, a highly accurate system using mobile sensors to track targets. To capture spatial signal propagation in the most general environment, we propose to use an RSS matrix, where the (i, j) entry is the received signal strength (RSS) at location j for a transmitter at location i. Given sensor locations, beacons received by the sensors and RSS matrix, Mosent localizes the targets over time using a modified particle filter which jointly considers the temporal and spatial information of the targets. To learn spatial signal propagation, Mosent has another optional and independent module which generates RSS matrix efficiently using mobile sensors only. Based on the beacons received by sensors and their rough locations, the RSS matrix is refined over time as sensors move in the venue.

This is the first work leveraging both temporal information and target cooperation for tracking with mobile sensors. We have conducted extensive experiments at campus and mall environments. Our results show that Mosent achieves 4.37m and 9.46m tracking error in the campus and the shopping mall, respectively. Compared with state-of-the-art approaches, Mosent achieves significantly lower localization error (often by more than 30 percent) with much fewer sensors.

ACKNOWLEDGMENTS

This work was supported, in part, by Guangdong Provincial Department of Science and Technology (GDST16EG04/2016A050503024), and Guangzhou Science Technology and Innovation Commission (GZSTI16EG14/201704030079).

REFERENCES

[1] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C Miller. 2014. 3D Tracking via Body Radio Reflections. In *NSDI*, Vol. 14. Usenix, Seattle, WA, USA, 317–329.

[2] Craig S Agate and Ronald A Iltis. 1999. Statistics of the RSS estimation algorithm for Gaussian measurement noise. *IEEE transactions on signal processing* 47, 1 (1999), 22–32.

[3] Naoki Akai and Koichi Ozaki. 2015. Gaussian processes for magnetic map-based localization in large-scale indoor environments. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, IEEE, Hamburg, Germany, 4459–4464.

[4] Abdullah Alomari, Frank Comeau, William Phillips, and Nauman Aslam. 2017. New path planning model for mobile anchor-assisted localization in wireless sensor networks. *Wireless Networks* 0, 0 (2017), 1–19.

[5] 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*, Vol. 2. Ieee, Ieee, Tel Aviv, Israel, 775–784.

[6] Md Zakirul Alam Bhuiyan, Guojun Wang, and Athanasios V Vasilakos. 2015. Local area prediction-based mobile target tracking in wireless sensor networks. *IEEE Trans. Comput.* 64, 7 (2015), 1968–1982.

[7] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[8] Ingwer Borg and Patrick JF Groenen. 2005. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, NY, USA.

[9] Agata Brajdic and Robert Harle. 2013. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, ACM, Zurich, Switzerland, 225–234.

[10] Li-wei Chan, Ji-rung Chiang, Yi-chao Chen, Chia-nan Ke, Jane Hsu, and Hao-hua Chu. 2006. Collaborative Localization: Enhancing WiFi-Based Position Estimation with Neighborhood Links in Clusters. In *Pervasive Computing*, Kenneth P. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 50–66.

[11] Ruifeng Chen, Zhangdui Zhong, and Minming Ni. 2011. Cluster based iterative GPS-free localization for wireless sensor networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. IEEE, IEEE, Yokohama, Japan, 1–5.

[12] Xi Chen, Andrea Edelstein, Yunpeng Li, Mark Coates, Michael Rabbat, and Aidong Men. 2011. Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*. IEEE, IEEE, Chicago, IL, USA, 342–353.

[13] Jie Cheng, Zeqi Song, Qiang Ye, and Hongwei Du. 2016. MIL: a mobile indoor localization scheme based on matrix completion. In *Communications (ICC), 2016 IEEE International Conference on*. IEEE, IEEE, Kuala Lumpur, Malaysia, 1–5.

[14] Sajal K Das, Jing Wang, Ratan K Ghosh, and Rupert Reiger. 2011. Algorithmic aspects of sensor localization. In *Theoretical aspects of distributed computing in sensor networks*. Springer, Berlin, Heidelberg, German, 257–291.

[15] Giuseppe Thadeu Freitas de Abreu and Giuseppe Destino. 2007. Super MDS: source location from distance and angle information. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*. IEEE, IEEE, Kowloon, China, 4430–4434.

[16] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. 2000. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., Morgan Kaufmann, San Francisco, CA, USA, 176–183.

[17] Rick Durrett. 2010. *Probability: Theory and Examples* (4th ed.). Cambridge University Press, New York, NY, USA.

[18] Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. 2015. CityMomentum: an online approach for crowd behavior prediction at a citywide level. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, ACM, Osaka, Japan, 559–569.

[19] Ningjia Fu, Jianzhong Zhang, Wenping Yu, and Changhai Wang. 2017. Crowdsourcing-based wifi fingerprint update for indoor localization. In *Proceedings of the ACM Turing 50th Celebration Conference-China*. ACM, ACM, Shanghai, China, 34.

[20] Subir Halder and Amrita Ghosal. 2016. A survey on mobile anchor assisted localization techniques in wireless sensor networks. *Wireless Networks* 22, 7 (2016), 2317–2336.

[21] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review* 41, 1 (2011), 53–53.

[22] Guangjie Han, Xuan Yang, Li Liu, Mohsen Guizani, and Wenbo Zhang. 2017. A Disaster Management-Oriented Path Planning for Mobile Anchor Node-Based Localization in Wireless Sensor Networks. *IEEE Transactions on Emerging Topics in Computing* 0, 0 (2017), 0.

[23] S. He and S. H. G. Chan. 2014. Sectjunction: Wi-Fi Indoor Localization Based on Junction of Signal Sectors. In *2014 IEEE International Conference on Communications (ICC)*. 2605–2610. https://doi.org/10.1109/ICC.2014.6883716

[24] S. He and S. H. G. Chan. 2016. Tilejunction: Mitigating Signal Noise for Fingerprint-Based Indoor Localization. *IEEE Transactions on Mobile Computing* 15, 6 (June 2016), 1554–1568. https://doi.org/10.1109/TMC.2015.2463287

[25] Suining He and S-H Gary Chan. 2016. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials* 18, 1 (2016), 466–490.

[26] Suining He, S-H Gary Chan, Lei Yu, and Ning Liu. 2015. Calibration-free fusion of step counter and wireless fingerprints for indoor localization. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, ACM, Osaka, Japan, 897–908.

[27] Lingxuan Hu and David Evans. 2004. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, ACM, Philadelphia, PA, USA, 45–57.

[28] Lutful Karim, Nidal Nasser, Qusay H Mahmoud, Alagan Anpalagan, and TE Salti. 2015. Range-free localization approach for M2M communication system using mobile anchor nodes. *Journal of Network and Computer Applications* 47 (2015), 137–146.

[29] Jung-Kyu Lee, Youngjoon Kim, Jong-Ho Lee, and Seong-Cheol Kim. 2014. An efficient three-dimensional localization scheme using trilateration in wireless sensor networks. *IEEE Communications Letters* 18, 9 (2014), 1591–1594.

[30] Xinrong Li. 2007. Collaborative localization with received-signal strength in wireless sensor networks. *IEEE Transactions on Vehicular Technology* 56, 6 (2007), 3807–3817.

[31] Zan Li, Danilo Burbano Acuna, Zhongliang Zhao, Jose Luis Carrera, and Torsten Braun. 2016. Fine-grained indoor tracking by fusing inertial sensor and physical layer information in wlans. In *Communications (ICC), 2016 IEEE International Conference on*. IEEE, IEEE, Kuala Lumpur, Malaysia, 1–7.

[32] Jun Liu, Zhaohui Wang, Jun-Hong Cui, Shengli Zhou, and Bo Yang. 2016. A joint time synchronization and localization design for mobile underwater sensor networks. *IEEE Transactions on Mobile Computing* 15, 3 (2016), 530–543.

[33] Pratap Misra and Per Enge. 2006. Global Positioning System: signals, measurements and performance second edition. *Massachusetts: Ganga-Jamuna Press* 0, 0 (2006), 466–490.

[34] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*. American Association for Artificial Intelligence, Edmonton, Alberta, Canada, 593–598.

[35] ABM Musa and Jakob Eriksson. 2012. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*. ACM, ACM, Toronto, Ontario, Canada, 281–294.

[36] Gabriele Oliva, Stefano Panzieri, Federica Pascucci, and Roberto Setola. 2015. Sensor networks localization: Extending trilateration via shadow edges. *IEEE Trans. Automat. Control* 60, 10 (2015), 2752–2755.

[37] Luis Oliveira, Hongbin Li, Luis Almeida, and Traian E Abrudan. 2014. RSSI-based relative localisation for mobile robots. *Ad Hoc Networks* 13 (2014), 321–335.

[38] Josep M Porta, Aleix Rull, and Federico Thomas. 2016. Sensor Localization from Distance and Orientation Constraints. *Sensors* 16, 7 (2016), 1096.

[39] Theodore Rappaport. 2001. *Wireless Communications: Principles and Practice* (2nd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[40] Yannic Schröder, Georg von Zengen, and Lars Wolf. 2015. Poster: NLOS-aware localization based on phase shift measurements. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, ACM, Paris, France, 224–226.

[41] Yi Shang, Wheeler Rumi, Ying Zhang, and Markus Fromherz. 2004. Localization from connectivity in sensor networks. *IEEE Transactions on parallel and distributed systems* 15, 11 (2004), 961–974.

[42] Yi Shang and Wheeler Ruml. 2004. Improved MDS-based localization. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, Vol. 4. IEEE, IEEE, Hong Kong, China, 2640–2651.

[43] Xiaotong Shen, Scott Pendleton, and Marcelo H Ang Jr. 2016. Scalable cooperative localization with minimal sensor configuration. In *Distributed Autonomous Robotic Systems*. Springer, Tokyo, Japan, 89–104.

[44] Junghun Suh, Seungil You, Sungjoon Choi, and Songhwai Oh. 2016. Vision-based coordinated localization for mobile sensor networks. *IEEE transactions on automation science and engineering* 13, 2 (2016), 611–620.

[45] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. 2009. VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM conference on embedded networked sensor systems*. ACM, ACM, Berkeley, CA, USA, 85–98.

[46] Xiaohua Tian, Mei Wang, Wenxin Li, Binyao Jiang, Dong Xu, Xinbing Wang, and Jun Xu. 2017. Improve Accuracy of Fingerprinting Localization with Temporal Correlation of the RSS. *IEEE Transactions on Mobile Computing* 17, 1 (2017), 113–126.

[47] Slavisa Tomic, Marko Beko, and Rui Dinis. 2015. RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes. *IEEE Transactions on Vehicular Technology* 64, 5 (2015), 2037–2050.

[48] Reza Monir Vaghefi and R Michael Buehrer. 2015. Cooperative Joint Synchronization and Localization in Wireless Sensor Networks. *IEEE Trans. Signal Processing* 63, 14 (2015), 3615–3627.

[49] Joshua Vander Hook, Pratap Tokekar, and Volkan Isler. 2015. Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints. *IEEE Transactions on Robotics* 31, 4 (2015), 864–876.

[50] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-Level Localization with a Single WiFi Access Point. In *NSDI*. Usenix, Santa Clara, CA, USA, 165–178.

[51] Hongwei Xie, Tao Gu, Xianping Tao, Haibo Ye, and Jian Lu. 2016. A reliability-augmented particle filter for magnetic fingerprinting based indoor localization on smartphone. *IEEE Transactions on Mobile Computing* 15, 8 (2016), 1877–1892.

[52] Hongwei Xie, Tao Gu, Xianping Tao, Haibo Ye, and Jian Lv. 2014. MaLoc: A practical magnetic fingerprinting approach to indoor localization using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, ACM, Seattle, WA, USA, 243–253.

[53] Qiang Xu, Rong Zheng, and Steve Hranilovic. 2015. IDyLL: Indoor localization using inertial and light sensors on smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, ACM, Osaka, Japan, 307–318.

[54] Zheng Yang and Yunhao Liu. 2010. Quality of trilateration: Confidence-based iterative localization. *IEEE Transactions on Parallel and Distributed Systems* 21, 5 (2010), 631–640.

[55] Moustafa Youssef and Ashok Agrawala. 2005. The Horus WLAN location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, ACM, Seattle, WA, USA, 205–218.

[56] Shigeng Zhang, Xuan Liu, Jianxin Wang, Jiannong Cao, and Geyong Min. 2015. Accurate range-free localization for anisotropic wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 11, 3 (2015), 51.

[57] Yang Zhao, Neal Patwari, Jeff M Phillips, and Suresh Venkatasubramanian. 2013. Radio tomographic imaging and tracking of stationary and moving people via kernel distance. In *Proceedings of the 12th international conference on Information processing in sensor networks*. ACM/IEEE, IEEE, Philadelphia, PA, USA, 229–240.

Received November 2017; revised May 2018; accepted September 2018