Received December 14, 2018, accepted December 26, 2018, date of publication January 4, 2019, date of current version January 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2890850

Indoor Location Learning Over Wireless Fingerprinting System With Particle Markov Chain Model

SOK-IAN SOU[1,2], (Member, IEEE), WEN-HSIANG LIN[1], KUN-CHAN LAN[3], AND CHUAN-SHENG LIN[1]

[1]Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan 701, Taiwan
[2]Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan
[3]Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan

Corresponding author: Sok-Ian Sou (sisou@mail.ncku.edu.tw)

The work of S.-I. Sou was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Contract MOST 105-2628-E-006-007-MY3.

ABSTRACT This paper describes research toward a tracking system for locating persons indoor based on low-cost Bluetooth Low Energy (BLE) beacons. Wireless fingerprinting based on BLE beacons has emerged as an increasingly popular solution for fine-grained indoor localization. Inspired by the idea of mobility tracking used in the cellular network, this paper proposes a BLE-based tracking system, designated as BTrack, to learn the location area (LA) of an indoor user based on the reported wireless fingerprinting combined with statistical analysis. We propose a new particle Markov chain model to evaluate the LA-level performance regarding the visibility area in an environment with large obstacles. In the presence of sight obstructions, BTrack is evaluated using a real-world test bed built in a library with tall bookshelves. The performance of the proposed system is evaluated in terms of the mean distance error and the LA prediction accuracy considering the direct line-of-sight. Compared with the existing methods, BTrack reduces the average localization error by 25% and improves the average prediction accuracy by more than 16% given a random mobility pattern.

INDEX TERMS People tracking, resampling, position accuracy, Bluetooth, beacons, machine learning, particle filter, Markov chain.

I. INTRODUCTION

Indoor Location-based Services (ILBS) have attracted much attention in recent years due to the growing commercial demand [1], [2]. With the absence of Global Positioning System (GPS) signal [3], many other solutions have been proposed for indoor positioning. The majority of previous indoor localization approaches utilizes Received Signal Strength (RSS) as a metric for location determination. This approach estimates the target location by matching online measurements of RSS with the closest offline features (i.e., the location fingerprints) composed of sample location coordinates and respective RSS vectors, based on algorithms such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM) [4], [5] and so on. The most commonly-used RSS-based localization is Wi-Fi fingerprinting approach because 802.11 access points and routers are widely available and deployed in most buildings [6], [7].

Due to the signal fluctuation, Wi-Fi-based localization methods have limitations and unpredictable performance in environments with the presence of large obstacles. A small localization error is important to some ILBS, e.g., a few meters of error in estimated location can place someone in a different room within a building. Therefore, high localization accuracy (within meter range) is still expected in order to offer satisfactory ILBS. In 2013, Apple Inc. introduced a new technology known as iBeacon to deliver proximity services to users. iBeacon is established upon Bluetooth Low Energy (BLE) technology [8], [9], which is both more energy-efficient and far cheaper than traditional Bluetooth and Wi-Fi technologies. BLE transmissions have a very short range (typically several meters, or less). Thus, in practice, the estimation errors obtained using BLE-based techniques are generally much lower than those obtained in Wi-Fi-based systems.

Although originally envisaged for proximity purposes [10], BLE technology also offers an intriguing opportunity for indoor localization and tracking applications [11]. Due to the increasing deployment of BLE beacons and easy accessibility to users, the feasibility of BLE beacon-based indoor localization systems is extensively investigated in [12]–[16]. Meanwhile, the Internet of Thing (IoT) concept is employed to realize localization and tracking services [17], [18]. For example, kid tracking applications based on BLE interface and wearable tracking devices (e.g., tags, straps or watches) becomes common nowadays [19], [20]. To assist the parent in locating the child, the app paired with a BLE beacon worn or carried by the child to monitor the signal received from the beacon and trigger an alarm if the beacon moves beyond a certain predefined range. Therefore, the use of BLE beacons is an alternative for ILBS [21]. However, such app reports the location of the child at the moment the signal was previously connected. It is of only limited use in actually finding the child; particularly in busy environments where the child is easily lost from sight among crowds of people, retail displays, promotional stands, and so on. Advanced tracking system is still needed.

In the literature, most existing WiFi fingerprint based indoor positioning system focuses on techniques that match the vector of RSS values reported by a mobile device to the fingerprints collected at predetermined reference points (RPs) by comparing the similarity between them. Existing Wi-Fi fingerprinting based solutions can achieve meter-level or room-level accuracy depending on the service requirement [22], [23]. Most recent studies mainly evaluate the localization performance in terms of the mean distance error, i.e., the average distance between the predicted position of the tracking user and the ground truth data. On the other hand, some works focus on region-based or room-based positioning that estimates the specific region/room a user resides in. To support the positioning requirement with different kinds of ILBS, both metrics are essential in building a location management system. This paper studies location tracking for ILBS with BLE beacons. The goal is to track the user location on both fine-grained level (i.e., typically less that 1 m^2) as well as coarse-grained level (i.e., typically less than 5 m^2).

Mobility management is the key functionality that has been utilized in the cellular networks for many years. Inspired by this idea, the present study proposes a *BLE-based tracking system*, designated as BTrack, to manage the user position with the scale of cell as well as location area (LA). It is the first work to consider both coarse-grained and fine-grained indoor positioning accuracy, in the form of a general and hierarchy-level (i.e., LA-level and cell-level inspired by the location management in cellular networks [24]–[27]) for location tracking and management. We note that the new metric LA-level accuracy can support ILBS such as the one mentioned in [28].

In our proposed system, we introduce a hierarchical database architecture for location management to support IBLS, where both cell (i.e., RP) and LA (a set of neighboring cells with the presence of direct path) are recorded. The user continuously monitors the signals of the surrounding BLE beacons and periodically reports the wireless fingerprinting (RSS vectors) to the BTrack server. The BTrack system learns the user location (i.e., the cell and the LA) and acts as a mediator that provides user location to other advanced ILBS. According to [29] and [30], the widely used cell-level localization algorithm is mainly based on particle filtering by combining the well-known KNN majority voting strategy. However, the LA-level localization is more complicated since a cell can belong to multiple LAs. Also, the reasons why a user resides in a specific LA can be interpreted from both human behavior and the geographical environment. Existing indoor positioning methods only give the estimated cell for a user; however, when a cell is included in more than one LA, it is necessary to further determine which LA is the best considering the sight obstruction problem. Therefore, it is challenging to build a LA learning model based on the LA transition statistics. Recently, some studies [31], [32] employ the particle filter along with the Markov chain Monte Carlo method to build a dynamic system for data assimilation model in earth science. In this work, we develop a particle Markov chain model to build the LA-level transition model, and to solve it by finding the stationary probability for each LA.

Specifically, the study provides three main contributions. First, we are the first to bring the concept of hierarchy location management into fingerprinting-based indoor localization. Second, we propose a new particle Markov Chain model to evaluate the LA-level performance regarding the visibility area with large obstacles environment, which is commonly seen in nowadays beacon deployment scenario (such as hypermarkets, shopping mall, libraries) but seldom discussed. Third, we assess both mean distance error and LA learning accuracy in a real-world environment containing significant obstacles that block the line-of-sight path.

We note that BTrack is not limited for the usage of kid tracking, other tracking applications can also employ the techniques of BTrack. Also, BTrack is developed based on KNN, a modified particle filtering, and a particle Markov chain. Particle filter-based techniques have been applied with great success to a variety of state estimation problems including localization with sensing data [33]–[35], object tracking [36]–[38], mobile robot localization [39]–[41], people tracking [42], [43], etc. The main novelty of this work includes a new hierarchy indoor localization concept (inspired by the cellular location management system) and a new particle Markov Chain model to evaluate the LA-level localization accuracy.

The remainder of this paper is organized as follows. Section II describes the problem scenario in this study. Section III introduces the basic operation and implementation architecture of the proposed BTrack system. Section IV describes the localization algorithms used in the proposed approach. Sections V and VI present the experimental setup and results, respectively. Finally, Section VII provides some brief concluding remarks and indicates possible areas of future work.

II. PROBLEM SCENARIO

This section develops the problem scenario of a potential ILBS application for kid tracking, which has attracted growing interest in recent years. In [19] and [20], BLE-based tracking devices are generally paired with iOS/Android smartphone apps to allow the parent to monitor the child over short distances (e.g., 20-30 m). Typically, such apps monitor the signal received from a BLE beacon worn or carried by the child and trigger an alarm if the beacon moves beyond a certain range. While such apps provide some measure of reassurance to the parents, they are of only limited use for actually locating the child since it only provides the ''last location'' of the child before the signal was lost.

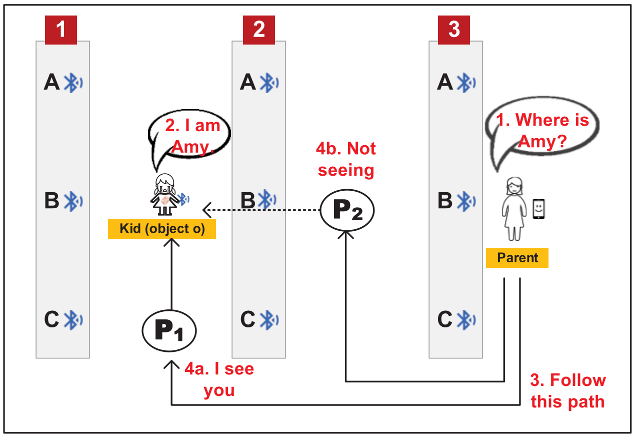


FIGURE 1. Illustration of a kid tracking application in an indoor environment with shelf layout ($f(o, P_1) = 1$ and $f(o, P_2) = 0$).

To support a better localization service in kid tracking, Fig. 1 demonstrates the concept of the BTrack system developed in this study. Assuming that the parent (mother) and kid (Amy) are in a typical store environment consisting of three rows of shelves (1, 2 and 3). In accordance with the store policy, each row of shelves is divided into three sections (A, B and C) and a BLE beacon is mounted on the top of each section. Assuming that both the parent (tracker) and the kid (tracked user) are in possession of mobile devices (usually, a parent carries with a smartphone and a child wears a lightweight device with WiFi/BLE modules) installed with the tracking app. Suppose further that while the mother is standing at Shelf 3B, Amy wanders off and is now standing adjacent to Shelf 1B. As soon as the mother does not see Amy (Step 1), she activates the tracker app (parent side) and the kid's side of the app installed on Amy's device immediately reports the wireless fingerprint to the BTrack server (Step 2). The wireless fingerprint is used to learn Amy's position using a localization algorithm based on our proposed BTrack system. The kid tracking application then sends a set of navigational instructions based on the estimated position. In Step 4, we illustrate two possible cases of the returned paths (locations) to the parent app to guide the mother toward Amy, and the detailed discussion is given below.

Practical tracking and localization environments often contain sight obstructions (e.g., shelving units in a retail store or bookshelves in a library). Thus, a lower average localization error does not always represent a better outcome in all ILBS. For example in the above kid tracking application, consider the estimated location outcomes P_1 and P_2 in Fig. 1. The distance error for the two outcomes is approximately the same. However, for P_2, Amy is hidden from the tracker (the parent) by Shelf 2B. In other words, if the parent is guided to point P_2, she still cannot see Amy. Thus, the tracking outcome is unsatisfactory from the parent's perspective. By contrast, for P_1, the parent is approximately the same distance from Amy, but, in this case, can actually see her. As a result, P_2 represents a far more satisfactory outcome. In other words, prediction accuracy excluding line-of-sight obstacles is an important issue when evaluating the performance of tracking and localization schemes. *To the best of our knowledge, localization performance based on direct line-of-sight location area is not discussed before. Accordingly, this study measures the location and tracking system in both mean distance error and prediction accuracy considering direct line-of-sight.* The location is tracked on the level of cell (with a size up to 1 m^2) and of LA (with a total size up to 5 m^2). In particular, a LA does not contain any obstacles such that the tracked user is being visible from the predicted position P. Specifically, the event of satisfactory tracking is based on having direct line-of-sight, which can be defined by the function $f(o, i)$ in [28] as

$$f(o,p) = \begin{cases} 1, & \text{if person } o \text{ is visible at location } p. \\ 0, & \text{else.} \end{cases} \tag{1}$$

For example, in Fig. 1, Amy (i.e., person o) is visible from position P_1, and hence $f(o, P_1) = 1$. By contrast, she is invisible from position P_2, and hence $f(o, P_2) = 0$.

III. DESIGN OF BTrack SYSTEM

The BTrack system proposed in this study employs a BLE beacon paradigm to accomplish the fingerprint-based localization and tracking of a person. Specifically, we consider crowded indoor environments such as retail stores, shopping malls, public libraries, museums, and so on. Briefly, BLE beacons are deployed at known locations throughout the indoor environment (e.g., in the aisles or on the shelves in a retail store), and a mobile device containing both Wi-Fi and BLE modules is attached to the tracking person in the form of a wearable device. The Wi-Fi module is used to communicate with the BTrack server, while the BLE module is used to detect the signals of the nearby BLE beacons. The received RSS vector is sent periodically to the BTrack server to learn the updated current location of the tracking person. Importantly, the frequency at which the RSS fingerprint is collected and sent to the BTrack server can be configured in accordance with the required localization accuracy and the frequency of the location information is used; thereby enabling a tradeoff to be made between the localization accuracy and the battery consumption of the tracking device.

A. IMPLEMENTATION ARCHITECTURE

This subsection describes the implementation architecture of the proposed BTrack system. The discussions commence by describing the protocol stack among the BLE beacons, the mobile handheld devices (e.g., a smartphone and a wearable device with a Wi-Fi/BLE interface), and the server. The communication between the ILBS app and the BTrack server is also handled by Wi-Fi or a cellular network. The detailed design of the location management in BTrack Server is then introduced and discussed.

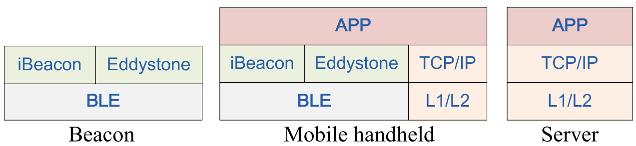


FIGURE 2. The protocol stack of BTrack system.

Fig. 2 shows the protocol stack employed in the BTrack system. As shown, the communications between the deployed beacons and the mobile handheld (e.g., wearable user device) are handled by iBeacon/Eddystone protocol over BLE interface, while those between the mobile handheld and the BTrack server are handled by Wi-Fi or cellular interface. The user device periodically detects the `Advertisement` messages (UUID, Major, Minor and TX_Power) of nearby beacons deployed over the BLE beacon platform. The ILBS app is implemented on top of the TCP/IP protocol between the handheld/wearable device and the server. In particular, the app is implemented using Representational State Transfer (REST); an architectural style whose Application Programming Interfaces (APIs) facilitate connection and interaction with both Web services and cloud services.

Fig. 3 shows the detailed implementation framework of BTrack server. The server contains both a `RESTful API module` and a `Web module` to support interaction with the BTrack app and other web/cloud services. All of the messages produced in BTrack are encapsulated using the `RESTful API module` and are transmitted over the HTTPS protocol. The `BTrack User Profile` is a database to store the user' profiles (i.e., the unique IDs of the tracked users, ILBS service information, and so on). The `Fingerprint` database stores the beacon profile (i.e., the IDs of the deployed beacons together with their position information (e.g., the cell identity and the LA identity)). The `Fingerprint` database additionally stores the mapping between the RSS vectors and the corresponding location coordinates. Finally, the `Map Data` database stores a map of the floor plan showing the positions of the BLE beacons and the corresponding cell/LA information, respectively.

B. INDOOR LOCALIZATION

This subsection presents the indoor localization method and the user tracking procedure performed by the BTrack server, as indicated in the lower part of Fig. 3. Implementing the location fingerprinting procedure in the BTrack system involves in two processes, namely training and operating. In the present study, the dataset required for training purposes was compiled by measuring the RSS values at each sample location in the experimental environment (see also Section IV-A) using an android app (see Fig. 5(c))) and then integrating these values with the shelf and aisle information in the `Map Data` database. The operating process is triggered when the `Location Area Update (LAU)` procedure is executed: The `LAU` function on the BTrack app collects the RSS data periodically and transmitted to the BTrack server, where they were used to learn the localization on the sever side. When requested by an ILBS, the BTrack server reports the location (cell, LA) to it.

As the operating process shown in Fig. 3, the RSS values reported by the user device are normalized and input to a KNN algorithm (see Algorithm 1), which provides an initial guess of the user's location based on the difference between the normalized RSS values of the user device and those obtained during the training process. An improved estimate of the user's position is then obtained using a particle filter algorithm (see Algorithm 2) based on the user's trajectory, map information, and pedestrian behavior, respectively.

1) RSS CALIBRATION BASED ON RESCALING

To solve the device diversity problem [44], several RSS calibration methods have been proposed, including the DIFFerence of signal strength (DIFF) [45], the Signal Strength Difference [46], the cosine similarity [47], and linear calibration [48]. However, in practical tracking environments, such methods have limited performance due to line-of-sight obstructions. As described above, BTrack computes the real-time location of the user. As a result, the computation time must be minimized such that the server can respond with the estimated user position in real-time. Accordingly, the present study adopts a simple min-max normalization method (also called rescaling) to match the RSS values in a range of [0, 100]. How to select the feature scaling method and the target range depends on the nature of the data. Specifically, the system first finds the minimum and maximum RSS values in the collected RSS dataset in the `Fingerprint` database, and stores them as $R_{\min}$ and $R_{\max}$, respectively. Each RSS value, R_i, in the dataset is then mapped to an integer value in the range of 0 to 100 in accordance with Eq. (2). Note that in the case of missing data, i.e., the user device does not receive any signal from a BLE beacon since the RSS vector was last collected, BTrack maps the missing RSS value as 0. The general formula is given as:

$$R'_i = \frac{R_i - R_{\min}}{R_{\max} - R_{\min}} \times 100 \tag{2}$$

2) MACHINE LEARNING

As shown in Algorithm 1, the KNN algorithm takes the normalized RSS data from the user device as the input and

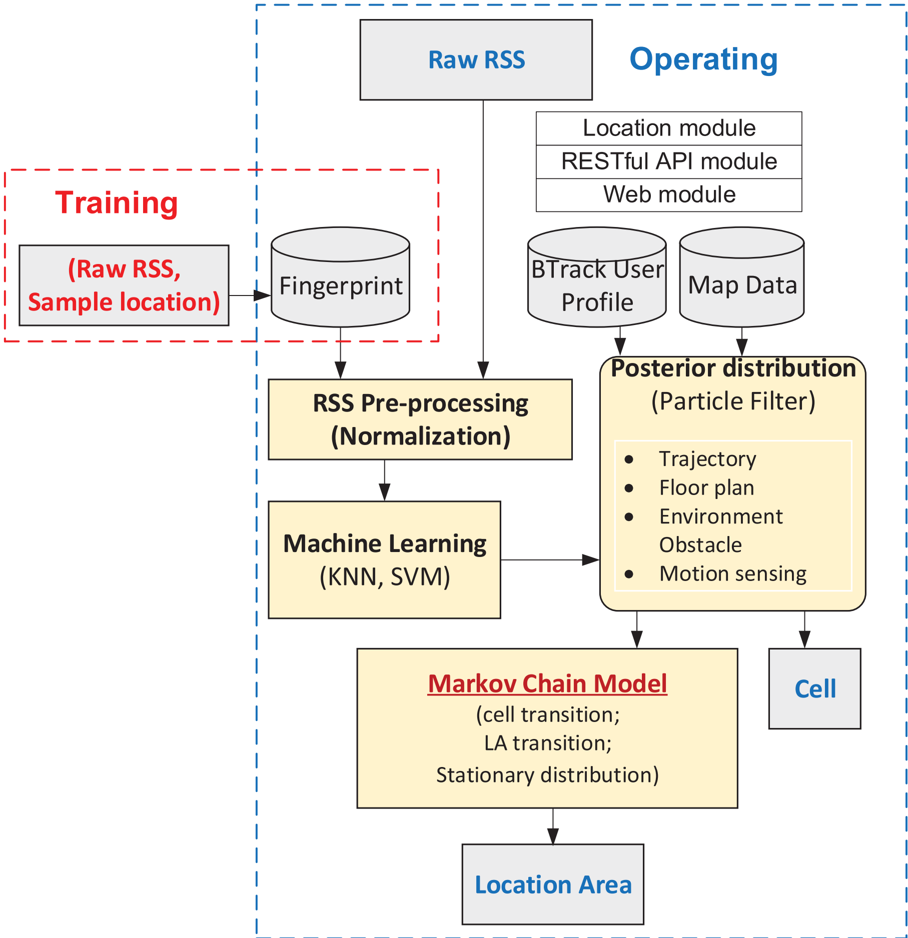


FIGURE 3. The implementation framework of BTrack server.

Algorithm 1 K-Nearest Neighbors (KNN) Algorithm

Require: $R_i = (r_i^1, r_i^2, \ldots, r_i^m)$, $T = (t^1, t^2, \ldots, t^m)$, K
Ensure: position P

$R'_i = \frac{R_i - R_{min}}{R_{max} - R_{min}} \times 100$
for $i = 1$ to n **do**
  $D_i = \sqrt{\sum_{j=1}^{m} (T(j) - R'_i(j))^2}$
end for
$P_K \Leftarrow$ Select K positions according to smallest D_i
$P \Leftarrow$ majority position of P_K
return P

then estimates the position of the user based on the k nearest data instances in the `Fingerprint` database (i.e., the training dataset). Note that R_i represents the i-th set of training RSS vector, T is the input RSS vector, and m is the total number of nearby beacons in the considered region. The KNN algorithm first computes the Euclidean distance D_i between the i-th training data instance and the input data instance. Then it selects K closest sample locations, i.e., those having the smallest values of D_i and chooses the majority of these locations as the output. That is, the outcome P obtained by the KNN algorithm is one particular instance of the set of sample locations, and is also treated as an initial estimated location of the tracking person.

3) POSTERIOR DISTRIBUTION OF LOCATION SAMPLES

In the context of particle filter, the posterior distribution of a stochastic process given noisy and/or partial observations is approximated by a set of random samples called particles, each associated with an important weight [49]. Here, a particle represents a possible cell where a user might reside in. The output of the user location is modeled as a predicted particle with noise, where limitation of gait speed, the user's mobility trace, and the floor plan are relevant environmental information.

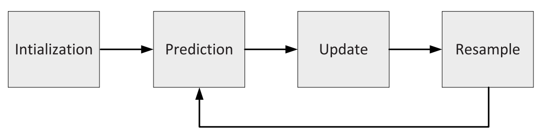


FIGURE 4. The procedure of a typical particle filter.

The following section presents how our modified particle filter method continuously maintain the posterior distribution of the set of location samples based on a new guess location P in each time period t. Fig. 4 first shows the basic operation of the particle filter. In the initialization stage, according to a prescribed sampling distribution (a uniform distribution, in the present study), the algorithm generates a number of particles (i.e., location samples refer to cells or RPs in the region of interest) for the tracking user on the map of the study area considered. The prediction stage considers (1) the last estimated location, (2) the particle distribution obtained in the previous round and (3) a new guess of the current position by a fingerprinting mapping system, such as KNN used in this work. We note that both the particle distribution (i.e., location sample distribution) and the new guess location are used to update the particle distribution in next round. Also, environmental factors, such as the constraints imposed on the particle movement by the pedestrian gait speed and obstacles are considered for updating the weight associated with each particle in this round. Here, we propose two rules for adjusting the particles' weight:

1) the weights associated with particles around the current guess position P are added;
2) the weight of any particle (referring to any location sample) associated with infeasible movement, such as the user passing through an obstacle or traveling too far from the previous predicted position, should be reduced.

Finally, in the resampling stage, the particle filter normalizes and updates all of the weights with their sum and obtains a probability distribution of the tracking user being located around each sample location. Finally, the particle filter chooses the particle with the largest weight as the output cell (i.e., the estimated position of the tracking user).

Algorithm 2 shows the pseudocode of the particle filter algorithm. In lines 1-6, the algorithm generates N particles (sample locations) uniformly distributed over the considered region, and assigns each particle an initial weight. Here, notation $< x_t^i, w_t^i >$ indicates that at time t, the i-th particle is associated with weight w_t^i at position x_t^i. For each location update period t, as shown in Line 10, the algorithm takes the initial estimate of the tracking user P as the input, which is obtained by the KNN algorithm. In lines 11-17, the algorithm resamples each particle in accordance with the weight w_{t-1} in the previous round, and determines the new particle position x_t^i based on P_t and the environmental factors, e.g., the gait speed limitation S. After adjusting the weight (here, by reducing the weights of any particles associated with abnormal movement, such as blocked by an obstacle O_t), the algorithm resamples all the particles, and BTrack generates the user location, L, associated with the largest weight. Specifically, the map information acquired from the `Map Data` database is also used in Line 14. We adjust the weights in Lines 14-15 as follows:

1) Adding associated weights to particles near the current guess position P, where distance$(P, x_{t-1}^i) = 0$ represents the distance between P and x_{t-1}^i. When $P = x_{t-1}^i$. In this part, we adjust the weight by adding $\eta_1[\frac{1}{1+\text{distance}(P,x_{t-1}^i)}]$, where η_1 $(0 \leq \eta_1 \leq 1)$ is the learning factor for the new guess of location.
2) Reducing associated weights to particles that would cause infeasible movement from the previous location to the current location. We adjust the weight by $(1-\eta_2)w_{t-1}^i + \eta_2 w_{t-1}^i I(x_{t-1}^i, x_t^i)$, where η_2 is the learning factor for the infeasible movement caused by the obstacles and
$$I(x_{t-1}^i, x_t^i) = \begin{cases} 1, & \text{if moving from } x_{t-1}^i \text{ to } x_t^i \text{ is feasible} \\ 0, & \text{else.} \end{cases} \tag{3}$$
In this part, the associated weight is decreased by $\eta_2\ w_{t-1}^i$ $(0 \leq \eta_2 \leq 1)$ when there is infeasible movement; while other associated weight is unchanged when the movement is feasible.

Algorithm 2 Modified Particle Filter

Require: Map information
Ensure: L

1: **if** $t = 0$ **then**
2:   **for** $i = 1$ to N **do**
3:     Draw sample x_0^i from $q(.)$
4:     Construct weights w_0^i
5:   **end for**
6: **end if**
7:
8: **for** $t = 1$ to T **do**
9:   $S_t = \phi, \eta = 0$
10:   Using Algorithm 1 to find an initial P_t
11:   **for** $i = 1$ to N **do**
12:     Resample a particle x_{t-1}^i according to distribution given by x_{t-1} and w_{t-1}
13:     Sample x_t^i from $p(x_t|x_{t-1}, P_t, S_t)$
14:     Adjust weight w_t^i from $p(O_t|x_t^i)$
15:     $\eta = \eta + w_t^i$
16:     $S_t = S_t \cup < x_t^i, w_t^i >$
17:   **end for**
18:   $L \Leftarrow$ Find location with largest weight from $< x_t, w_t >$
19:   Output L
20:   **for** $i = 1$ to N **do**
21:     $w_t^i = w_t^i/\eta$
22:   **end for**
23: **end for**

Based on the above discussion, the next set of location samples is given by the weights as follows:

$$w_t^i = \eta_1 \left[\frac{1}{1 + \text{distance}(P, x_{t-1}^i)} \right] + (1-\eta_2)w_{t-1}^i + \eta_2 w_{t-1}^i I(x_{t-1}^i, x_t^i) \tag{4}$$

In Lines 20-22, the weights associated with the particles are further normalized and stored. In each location update period, the algorithm proceeds to the next round and computed the new position based on the last stored distribution and weights.

We note that the proposed Algorithm 2 is basically follows the practical filter method, where the core part (Lines 9-22) executed for tracking the user movement at each time period has computational complexity only linear $O(N)$, in the number of particles N. In the present study, a particle represents a RP or cell in the considered region. It is worth to note that the weight assignment to each particle is important and should be adjusted based on the characteristics of the mobility traces collected in the environment.

C. INDOOR LOCATION LEARNING WITH PARTICLE MARKOV CHAIN

Based on the above particle filtering, we obtain the posterior distribution for each cell. This subsection focuses on indoor location learning. Here, we first construct the cell-level transitions $B(z, z')$ according to the particles' located cells at times $t-1$ and t. Then, we further construct the LA-level transition probabilities from a set of transitions $B(.)$. In practice, we define an indicator function $I_t^i(z)$ as

$$I_t^i(z, z') = \begin{cases} 1, & \text{if } (x_{t-1}^i = z) \wedge (x_t^i = z') \\ 0, & \text{else.} \end{cases} \tag{5}$$

According to the N particle locations, we model the cell-level transitions probability from cell z at time $t-1$ to cell z' at time t as

$$B_t(z, z') = \sum_{i=1}^{N} I_t^i(z, z') w_t^i \tag{6}$$

From (6), we model the LA transition as a first-order non-homogeneous Markov chain with transition matrix $\mathcal{P}_t$, whose state space $\mathcal{S}$ is the set of all LAs in the considered indoor environment. Suppose that both $Z', Z \in \mathcal{S}$; LA Z contains n_z cells, denoted by a set $C(Z') = \{z'_1, z'_2, \ldots, z'_{n_z}\}$; while a neighboring LA Z contains $n_{z'}$ cells, denoted by a set $C(Z) = \{z_1, z_2, \ldots, z_{n_z}\}$.The transition probability $T(.)$ from LA Z to LA Z' can be computed by combining all the possibilities that a user located in any cell $z' \in S(Z')$ moving to any cell $z \in S(Z)$, which is shown below

$$\mathcal{P}_t(Z', Z) = \sum_{k=1}^{n'_z} \sum_{l=1}^{n_z} B_t(z'_k, z_l) \tag{7}$$

Since a cell can be included in multiple overlapping LA, we further normalize the transition probability as

$$\mathcal{P}_t^*(Z', Z) = \frac{\mathcal{P}_t(Z', Z)}{\sum_{\forall Y \in \mathcal{S}} \mathcal{P}_t(Z', Y)} \tag{8}$$

After we obtain the transition matrix $\mathcal{P}^*$, we computed the stationary probability $\pi(Z)$ by solving

$$\pi(Z) = \sum_{\forall Z' \in \mathcal{S}} \mathcal{P}_t^*(Z', Z)\pi(Z') \tag{9}$$

Finally, the estimated LA is the one which has the maximum likelihood that a user located at, which is given by

$$Z^* = \arg_{Z, Z \in \mathcal{S}} \max \pi(Z) \tag{10}$$

IV. EXPERIMENTAL TESTBED

This section describes the experimental testbed setup, the method used to construct the location training dataset, and the procedure employed to test the BTrack system under various pedestrian mobility patterns.

A. TESTBED SETUP

The testbed was constructed in the library of National Cheng Kung University (Tainan, Taiwan). The testbed is deployed on a region having dimensions of 7.1 m ×4.2 m, with three rows of tall bookshelves as sight obstructions (see Fig. 5(a)). Each row of shelves contain five separate bookcases, each with a width of 0.6 m, a depth of 0.9 m and a height of 2.15 m[1] The aisles between the bookshelves have a width of 1.1 m. As shown in Fig. 5(b), the experimental area was partitioned into 34 grid cells,[2] with a size ranging from 0.54 m^2 to 0.66 m^2. The sample locations (see the • symbols) are the center of the cells and are referred to as possible positions for user tracking. To provide direct line-of-sight path, each LA contains several neighboring cells and forms a visible location area with no obstacle blocking. In our testbed, a LA ranges from 4.2 m^2 to 4.6 m^2 (see Fig. 5(c)) . Also, a cell can belong to more than one LA.

TABLE 1. Average RSS of the smartphones measured in the experiments.

Beacon ID	ASUS (ZE551ML) (unit: dB)	HTC (Desire 728) (unit: dB)	Diff. (unit: dB)
1	−85.69	−87.72	2.03
2	−80.72	−83.02	2.30
3	−87.47	−91.64	4.17
4	−85.12	−88.49	3.37
5	−88.03	−91.69	3.66
6	−82.13	−84.21	2.08
7	−92.41	−97.70	5.52
8	−91.90	−97.42	5.52
9	−84.70	−86.14	1.44

The experiments were performed using two different smartphone models, namely an HTC Desire 728 and an ASUS ZE551M, respectively (see Fig. 6(a)). Table 1 shows the RSS variance between these two devices. Also, Table 2 shows the comparison under different experimental settings. Note that we intend to employ two different smartphone models throughout the evaluation; one is used to build the

[1]We note that the large size of the obstacle does prevent the signal propagation and cause multipath fading, which highly affects the performance of location accuracy. The testing environment we choose is with high and big shelves, which is a very challenging environment.

[2]Although we use a grid-based partition to identify a cell, other cell shapes are also possible. Therefore, one can partition the region of interest into several cells with different size and/or different shape even in a complex environment. We suggest that a cell should not contain obstacle in the environment. It is worth to note that the key point is we use the center of a cell to identify a position (i.e., a reference point (RP)). Therefore, BTrack can be easily used any indoor environment.

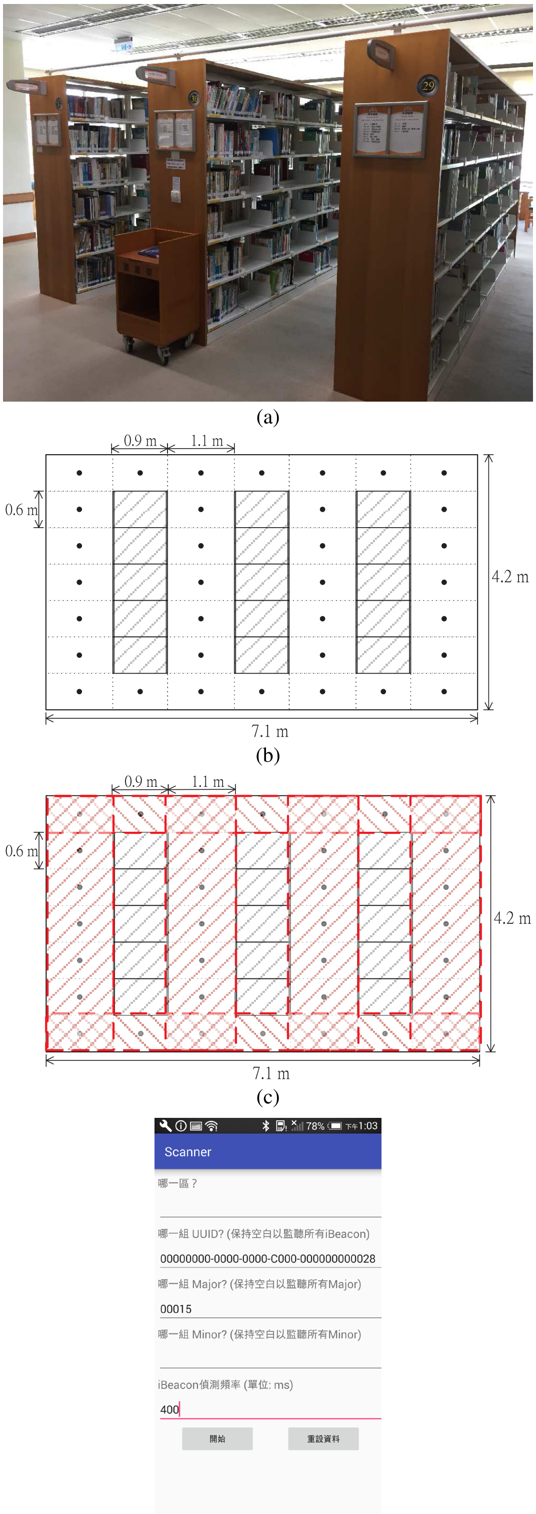


FIGURE 5. Experimental platform. (a) NCKU library. (b) Floor plan containing 34 cells. (c) Floor plan containing six LAs. (d) Android app.

`Fingerprint` database and the other is used to test the BTrack algorithm. Since the wireless signals are trained and tested from two different handheld chipsets, the experiment result is more convincing and useful. Based on the observations, we also select to use KNN algorithm instead of SVM algorithm. Here, the `LAU` function is implemented on both smartphones. However, it can be implemented on a small-sized development board with Wi-Fi/BLE modules, such as Raspberry Pi, and can be used as the form of a wearable device.

FIGURE 6. Experimental devices. (a) The training and the operating mobile devices: HTC Desire 728 and ASUS ZE551ML. (b) The beacon device.

Fig. 6(b) shows one of the BLE beacons used to periodically broadcast wireless signals throughout the experimental environment. The BLE beacons were mounted on top of the bookshelves. To build the location training dataset for the `Fingerprint` database at each sample location, we use a smartphone (HTC Desire 728) to measure RSS vectors at the center of each cell in the testbed. Then we utilize the machine learning technique mentioned in the previous section to train the system.

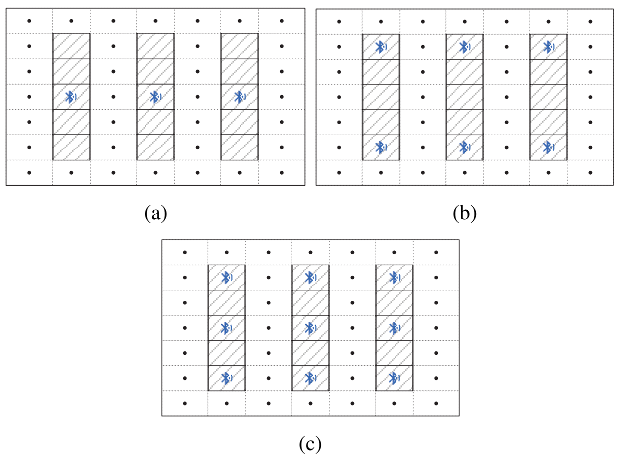


FIGURE 7. Beacon Deployments. (a) 3-beacon-deployment. (b) 6-beacon-deployment. (c) 9-beacon-deployment.

As shown in Fig. 7, three different beacon deployments were considered. In every case, the broadcasting period of the beacons was set as 200 ms and the BLE interface on the smartphone scanned the beacon signals every 400 ms. In performing the experiments, BTrack reported the estimated position of the tracking user in terms of the sample location associated with the center of the corresponding cell. The localization performance was evaluated in terms of both the mean distance error (i.e., the distance between the predicted

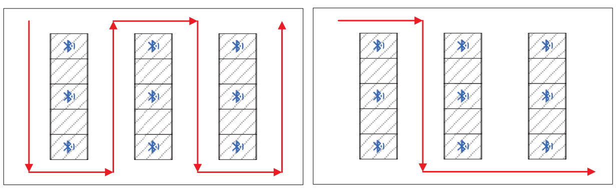
TABLE 2. **Comparison of performance under different experimental settings, and the same training device (ASUS ZE551ML) is used in all experiments.**

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**
Testing Device	heterogeneous HTC Desire 728	homogeneous ASUS ZE551ML	heterogeneous HTC Desire 728	homogeneous ASUS ZE551ML
ML method	KNN	KNN	SVM	SVM
Distance Error	1.32 m	0.32 m	1.32 m	0.48 m
Tracking Accuracy	78.23%	94.53%	77.12%	92.06%

location and the ground truth one) and the prediction accuracy (i.e., the probability that the predicted cell is included in the corrected LA.)

B. DATA COLLECTION AND TRAJECTORY CREATION

To test and evaluate the performance of BTrack, a person carries with a smartphone (ASUS ZE551M) walking through all 34 cells in the testbed. The `LAU` function installed on the smartphone periodically receives the RSS nearby and reports the current received wireless fingerprint to BTrack server such that the user current location can be learnt and updated. By comparing the location learned from BTrack system and the cell the user actually resided, the localization accuracy is evaluated in terms of the mean distance error (i.e., the cell-level accuracy) and the LA prediction accuracy (i.e., the LA-level).



(a) (b)

FIGURE 8. **The trajectory paths considered in the experiment. (a) W-shaped path. (b) Z-shaped path.**

The signals broadcast by the BLE beacons were detected via our developed Android app (see Fig. 5(d)) and are stored in a JSON (JavaScript Object Notation) format. The training dataset was collected from 3:00 PM to 5:00 PM on March 24, 2017. We use the training smartphone (HTC Desire 728) to receive 100 wireless fingerprints (RSS vectors) at each sample location, i.e., the center of each cell. The received data along with the corresponding location are stored as the location training dataset in the `Fingerprint` database. For the system evaluation, we also consider to receive 100 wireless fingerprints according to three typical pedestrian mobility patterns. The user carrying the operating smartphone (ASUS ZE551M) walks around the testbed with three different trajectory paths, namely random, W-shaped, and Z-shaped. In the random path, the user wanders around the testbed at a random movement; i.e., with an equal chance to move with a random neighboring cell. Meanwhile, the W-shaped and Z-shaped paths were defined as shown in Figs. 8(a) and 8(b), respectively. The walking speed is in the range from 0.88 m/s to 1.5 m/s [50]. The RSS vectors collected based on these mobility traces are used to assess the localization accuracy in the testbed.

V. EXPERIMENTAL RESULTS

This section shows the experimental results obtained in our testbed. The mean distance error, which is the average Euclidean distance between the estimated location $\hat{p} = (\hat{x}, \hat{y})$ and the ground true location $p = (x, y)$ is computed as

$$\text{Mean distance error} = E[\sqrt{(x-\hat{x})^2 + (y-\hat{y})^2}] \quad (11)$$

Based on Eq. (1), the success LA tracking is defined on the event that a BTrack user can be tracked and is visible at the predicted location, where

$$\text{Success location tracking} = \begin{cases} 1, & \text{if } \sum_{p \in Z*} f(o,p) \geq 1 \\ 0, & \text{else.} \end{cases} \quad (12)$$

The LA prediction accuracy is the expected outcome of a success location tracking. For instance, in the kid tracking example, when the kid and the parent are in the same LA (it does not matter which cell he/she resides in), both users can see each other without sight obstruction. Fig. 9 shows the cumulative distribution function (CDF) of the mean distance error for the three beacon deployments described above. As expected, the distance error reduces with an increasing number of beacons. However, in practice, a tradeoff should be made between the deployment and operational cost as well as the positioning accuracy. It is worth noting here that when using only pure fingerprinting (i.e., KNN and the RSS dataset without any calibration), the mean distance error is greater than 1 m and is hence unsatisfactory in most indoor positioning applications. The following discussions evaluate the performance of BTrack (both the mean distance error and the LA prediction accuracy) when calibration and/or position tracking with the particle filter are considered.

In the remaining experiments, the evaluation is performed on the testbed employing the 9-beacon-deployment plan (see Fig. 7(c)). Fig. 10 shows the CDF of the mean distance error against different methods. In terms of the mean distance error, the BTrack system has the best performance, while the linear method is the second-best. Table 3 compares the localization performance metrics (mean distance error and LA prediction accuracy) obtained by BTrack for each trajectory path with those obtained by five existing methods.

TABLE 3. Comparison of the mean distance error and LA tracking accuracy with prior work.

Method	Pure fingerprinting [4]	DIFF [45]	SSD [46]	Cosine [47]	Linear [48]	**BTrack**
Path	Random Path					
Distance Error (m)	1.33	1.4	1.52	1.43	1.26	**0.94**
Prediction accuracy	78.77%	71.77%	67.48%	71%	78.57%	**91.38%**
Path	W-shaped Path					
Distance Error (m)	1.28	1.37	1.54	1.38	1.25	**1.01**
Prediction accuracy	76.83%	68.39%	63.82%	68.49%	76.95%	**85.27%**
Path	Z-shaped Path					
Distance Error (m)	1.27	1.44	1.48	1.47	1.2	**1.12**
Prediction accuracy	57.76%	67.4%	68.78%	65.91%	81.12%	**88.33%**

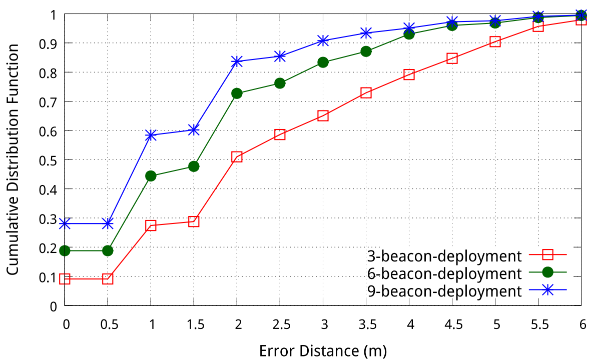


FIGURE 9. Positioning error of different deployments.

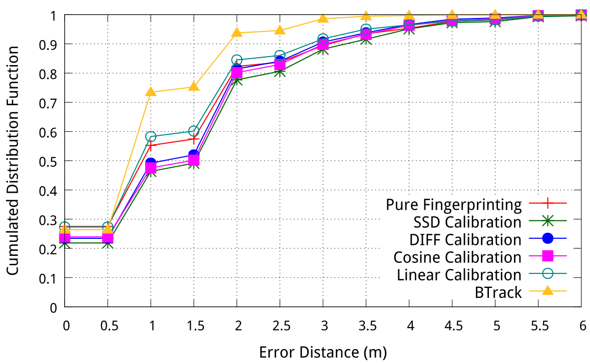


FIGURE 10. Accuracy comparison between our system and existing methods when user walks randomly.

In terms of the LA prediction accuracy, the pure fingerprinting method has the poorest performance of all the methods in the Z-shaped path; however, the LA prediction accuracy is actually higher than those of the other methods (DIFF, SSD, Cosine, Linear) for the random and W-shaped paths. In other words, our experimental results confirm that the performance depends on the user mobility. Since BTrack outperforms the other methods, we conclude that both RSS calibration as well as motion sensing are useful to implement the indoor fine-grained tracking system.

As described in Section III-B, BTrack uses both RSS calibration and a marginal distribution approach to estimate the possible sample location of the tracking person. Observing the results presented in Table 3, it is found that BTrack yields a lower mean distance error than all the existing methods for each of the considered trajectory paths. For example, BTrack yields an average distance error of just 0.94 m in the case of the random trajectory path. In other words, BTrack reduces the mean distance error by around 25% compared to that of the existing methods. Furthermore, BTrack achieves a prediction accuracy of 91.38% for the random path; corresponding to an average improvement of 16% over the existing methods. Similar results are obtained for both the W-shaped path and the Z-shaped path.

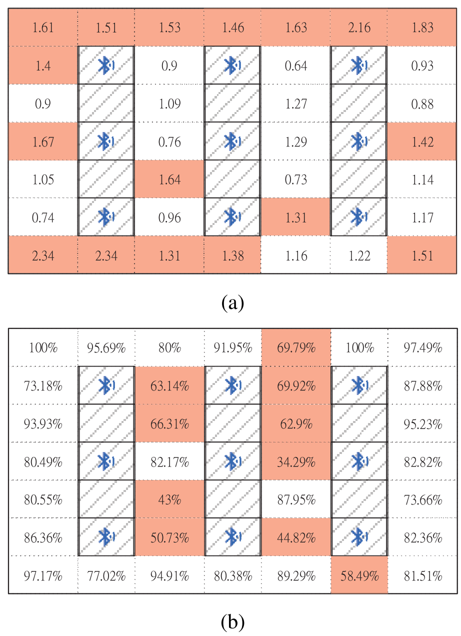


FIGURE 11. Localization performance measured at each reference point via pure fingerprinting. (a) Mean distance error (meters). (b) LA prediction accuracy.

Due to environmental factors (e.g., sight obstructions which prevent direct line-of-sight signals), the accuracy of the estimated position depends strongly on where the user resides (i.e., the particular cell at which the user resides in). Accordingly, a further series of experiments were performed in which the user remained stationary at each of the 34 sample locations in the testbed, and localization was performed using both the pure fingerprinting method and the BTrack system. The performance metrics obtained by the two methods are presented in Figs. 11 and 12, respectively.

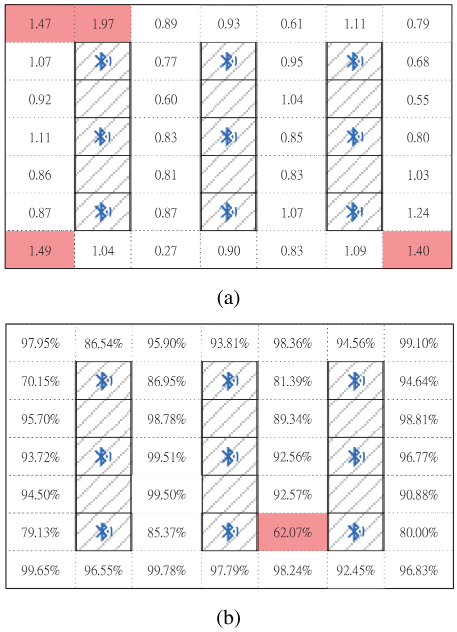


FIGURE 12. Localization performance measured at each reference point in BTrack. (a) Mean distance error (meters). (b) LA prediction accuracy.

The results confirm that, for both methods, the localization performance is significantly dependent on the environmental factors. In particular, the mean distance errors at the sample locations not surrounded by the bookshelves (i.e., further away from the deployed beacons) are much higher than those located between two bookshelves. For example, in Fig. 11(a), the mean distance errors at the sample locations in the first and final rows of the grid (see the highlighted areas) are usually very high varying in the range of $1.31 \sim 2.34$ m. By contrast, the reverse tendency is noted for the LA prediction accuracy. For example, as shown in Fig. 11(b), the prediction accuracy has a lower value vary in the range of $34.29\% \sim 69.92\%$ (see the highlighted area) for the sample locations located between two rows of bookshelves. This finding is reasonable since, in such positions, there is a greater chance that the line-of-sight of the tracker is blocked by the bookshelves. In other words, for objects located at sample locations between the bookshelves, even though the mean distance error is less than 1 m, the user may be hidden from the direct line-of-sight by bookshelves.

Fig. 12 shows that BTrack significantly improves both the mean distance error and the prediction accuracy. The performance improvement stems primarily from the use of the modified particle filter algorithm (Algorithm 2), which transforms the estimated locations into a probability distribution over sample locations as the tracked user moves over time by suppressing infeasible predictions (e.g., those associated with movements blocked by the shelves). Compared to the pure fingerprinting scheme, the performance of BTrack is also slightly improved by scaling RSS in Algorithm 1.

VI. CONCLUSIONS

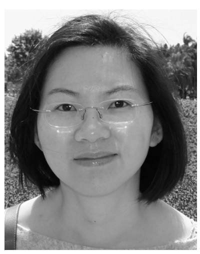
BLE beacons have emerged as a viable platform for various proximity-based services, including brand promotion and localization. The present study has proposed a user tracking and localization system based on BLE beacon technology for indoor location based services. The problem of improving the localization performance for the case where the propagation of the beacon signals is impaired by the presence of physical obstacles is also considered. The feasibility of the proposed system, designated as BTrack, has been demonstrated in a real-world environment using both the mean distance error and the location area prediction accuracy metrics. It is shown that compared with existing methods, BTrack reduces the average localization error by 25% and improves the average prediction accuracy by more than 16% given a random mobility pattern through the testbed area.

As a final remark, the beacon deployment density significantly affects the localization performance. The experimental results are mainly based on a 9-beacon-deployment plan. In the future, we would improve the performance in a low density beacon environment, such as a 3-beacon-deployment plan, i.e., each shelf only have a minimum of one beacon. Also, testing of BTrack in environment with random obstructions as well as randomly deployed beacons will be considered in the future work. Additional beacons should be deployed at other strategic locations within the environment, such as near the entrance/exit, next to the elevators, and so on. A future direction includes the study of the coexist of Wi-Fi/BLE indoor navigation, in which further performance improvement might be achievable under a lower density of beacon deployment plan.

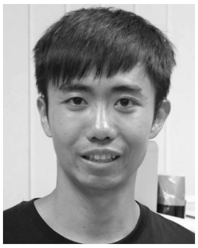
REFERENCES

[1] F. Zafari, I. Papapanagiotou, and K. Christidis, "Microlocation for Internet-of-Things-equipped smart buildings," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 96–112, Feb. 2016.
[2] O. Rashid, P. Coulton, and R. Edwards, "Implementing location based information/advertising for existing mobile phone users in indoor/urban environments," in *Proc. Int. Conf. Mobile Bus.*, 2005, pp. 377–383.
[3] R. Edwards. *2018 Best Wearable GPS Trackers for Kids Buyers Guide*. Accessed: Nov. 27, 2018. [Online]. Available: https://www.safewise.com/resources/wearable-gps-tracking-devices-for-kids-guide/
[4] V. Moghtadaiee, A. G. Dempster, and S. Lim, "Indoor localization using FM radio signals: A fingerprinting approach," in *Proc. IEEE Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2011, pp. 1–7.
[5] W. Farjow, A. Chehri, M. Hussein, and X. Fernando, "Support vector machines for indoor sensor localization," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2011, pp. 779–783.
[6] X. Du, K. Yang, and D. Zhou, "MapSense: Mitigating inconsistent WiFi signals using signal patterns and pathway map for indoor positioning," *IEEE Internet Things J.*, to be published. [Online]. Available: https://ieeexplore-ieee-org.er.lib.ncku.edu.tw/document/8267203
[7] F.-J. Wu and G. Solmaz, "We hear your activities through Wi-Fi signals," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 251–256.
[8] Z. Feng, L. Mo, and M. Li, "Analysis of low energy consumption wireless sensor with BLE," in *Proc. IEEE SENSORS*, Busan, South Korea, Nov. 2015, pp. 1–4.
[9] D.-Y. Liu, C.-S. Wang, and K.-S. Hsu, "Beacon applications in information services," in *Proc. Int. Conf. Adv. Mater. Sci. Eng.*, Nov. 2016, pp. 438–440.

[10] H. Zhang, J. Li, B. Wen, Y. Xun, and J. Liu, "Connecting intelligent things in smart hospitals using NB-IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1550–1560, Jun. 2018.
[11] P. Martin, B.-J. Ho, N. Grupen, S. Muñoz, and M. Srivastava, "An iBeacon primer for indoor localization: Demo abstract," in *Proc. 1st ACM Conf. Embedded Syst. Energy-Efficient Buildings (BuildSys)*, Memphis, TN, USA, 2014, pp. 190–191.
[12] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 1st Quart., 2015.
[13] Y. Fu, P. Chen, S. Yang, and J. Tang, "An indoor localization algorithm based on continuous feature scaling and outlier deleting," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1108–1115, Apr. 2018.
[14] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "BLE Beacons for Internet of Things applications: Survey, challenges, and opportunities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 811–828, Apr. 2018.
[15] N. Newman, "Apple iBeacon technology briefing," *J. Direct, Data Digital Marketing Pract.*, vol. 15, no. 3, pp. 222–225, Jan. 2014.
[16] D. Li, B. Zhang, Z. Yao, and C. Li, "A feature scaling based k-nearest neighbor algorithm for indoor positioning system," in *Proc. IEEE Global Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 436–441.
[17] A. Fujihara and T. Yanagizawa, "Proposing an extended iBeacon system for indoor route guidance," in *Proc. IEEE Int. Conf. Intell. Netw. Collaborative Syst. (INCOS)*, Taipei, Taiwan, Sep. 2015, pp. 31–37.
[18] S. Alletto *et al.*, "An indoor location-aware system for an IoT-based smart museum," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 244–253, Apr. 2016.
[19] *BuddyTag*. Accessed: Jan. 8, 2019. [Online]. Available: http://www.mybuddytag.com/
[20] *Linable*. Accessed: Jan. 8, 2019. [Online]. Available: http://lineable.net/
[21] R. Faragher and R. Harle, "Location fingerprinting with Bluetooth low energy beacons," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.
[22] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (Mobicom)*, 2012, pp. 269–280.
[23] I. Bisio *et al.*, "A Trainingless WiFi fingerprint positioning approach over mobile devices," *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 832–835, 2014.
[24] Y.-B. Lin and W.-N. Tsai, "Location tracking with distributed HLR's and pointer forwarding," *IEEE Trans. Veh. Technol.*, vol. 47, no. 1, pp. 58–64, Feb. 1998.
[25] S.-R. Yang, Y.-C. Lin, and Y.-B. Lin, "Performance of mobile telecommunications network with overlapping location area configuration," *IEEE Trans. Veh. Technol.*, vol. 15, no. 2, pp. 1285–1292, Mar. 2008.
[26] S.-I. Sou, J.-Y. Jeng, and P. Lin, "Improving session continuity through user mobility tracking for EPS inter-serving gateway handover," *Wireless Commun. Mobile Comput.*, vol. 12, no. 12, pp. 1077–1090, Aug. 2012.
[27] C.-Y. Hsiao, S.-I. Sou, and C.-S. Lin, "Small scale tracking scheme for low mobility devices in machine type communications," in *Proc. IEEE VTC Spring*, Seoul, South Korea, May 2014, pp. 1–5.
[28] P. Meissner and K. Witrisal, "Multipath-assisted single-anchor indoor localization in an office environment," in *Proc. 19th Int. Conf. Syst., Signals Image*, 2012, pp. 22–25.
[29] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[30] F. Zafari and I. Papapanagiotou, "Enhancing iBeacon based micro-location with particle filtering," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–7.
[31] H. Yan, C. M. DeChant, and H. Moradkhani, "Improving soil moisture profile prediction with the particle filter-Markov chain Monte Carlo method," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6134–6147, Nov. 2015.
[32] H. Bi, J. Ma, and F. Wang, "An improved particle filter algorithm based on ensemble Kalman filter and Markov chain Monte Carlo method," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 2, pp. 447–459, Feb. 2015.
[33] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, and M. T. Lim, "Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/FIR filtering," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1089–1098, Oct. 2015.
[34] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lu, "A reliability-augmented particle filter for magnetic fingerprinting based indoor localization on smartphone," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 1877–1892, Aug. 2016.
[35] P. Yang and W. Wu, "Efficient particle filter localization algorithm in dense passive RFID tag environment," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5641–5651, Oct. 2014.
[36] M. Talha and R. Stolkin, "Particle filter tracking of camouflaged targets by adaptive fusion of thermal and visible spectra camera data," *IEEE Sensors J.*, vol. 14, no. 1, pp. 159–166, Jan. 2014.
[37] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1819, Nov. 2005.
[38] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 240–247.
[39] C. G. Cifuentes, J. Issac, M. Wüthrich, S. Schaal, and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 577–584, Apr. 2017.
[40] C.-C. Hsu, S.-S. Yeh, and P.-L. Hsu, "Particle filter design for mobile robot localization based on received signal strength indicator," *Trans. Inst. Meas. Control*, vol. 38, no. 11, pp. 1311–1319, 2016.
[41] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, May 2001, pp. 1665–1670.
[42] V. Kılıç, M. Barnard, W. Wang, and J. Kittler, "Audio assisted robust visual tracking with adaptive particle filtering," *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 186–200, Feb. 2015.
[43] J. Yuan, H. Chen, F. Sun, and Y. Huang, "Multisensor information fusion for people tracking with a mobile robot: A particle filtering approach," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 9, pp. 2427–2442, Sep. 2015.
[44] J.-G. Park, D. Curtis, S. Teller, and J. Ledlie, "Implications of device diversity for organic localization," in *Proc. IEEE Proc. INFOCOM*, Apr. 2011, pp. 3182–3190.
[45] F. Dong, Y. Chen, J. Liu, Q. Ning, and S. Piao, "A calibration-free localization solution for handling signal strength variance," in *Mobile Entity Localization and Tracking in GPS-Less Environnments*. Berlin, Germany: Springer, 2009, pp. 79–90.
[46] A. K. M. Hossain, H. N. Van, Y. Jin, and W.-S. Soh, "Indoor localization using multiple wireless technologies," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, Oct. 2007, pp. 1–8.
[47] S. Han, C. Zhao, W. Meng, and C. Li, "Cosine similarity based fingerprinting algorithm in WLAN indoor positioning against device diversity," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 2710–2714.
[48] A. Haeberlen *et al.*, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 70–84.
[49] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 53–82, Jul. 2010.
[50] TranSafety, Inc., "Study compares older and younger pedestrian walking speeds," *Road Manage. Eng. J.*, 1997. [Online]. Available: http://www.usroads.com/journals/p/rej/9710/re971001.htm



SOK-IAN SOU (M'05) received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1997, 2004, and 2008, respectively. She was a Visiting Scholar with Carnegie Mellon University, in 2009. She is currently a Professor with the Institute of Computer and Communication Engineering and the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. She was a recipient of the Investigative Research Award from the Pan Wen Yuan Foundation, in 2009, the Young Researcher in Service Science Award from the Sayling Wen Cultural and Educational Foundation, in 2012, and the Outstanding Teacher Award, in 2017. She has co-authored the book *Charging for Mobile All-IP Telecommunications* (Y.-B. Lin; Wiley, 2008). Her current research interests include the Internet of Things (IoT), mobile networks, wireless sensing, the design and analysis of mobile communication services, and intelligent IoT applications.



WEN-HSIANG LIN received the B.S. degree from the Department of Computer Science and Information Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, in 2015, and the M.S. degree from the Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan, in 2017.

KUN-CHAN LAN received the B.A. degree in industrial management science from National Cheng Kung University, Taiwan, the master's degree in computer science from SUNY at Stony Brook, in 1997, advised by Prof. T.-C. Chiueh, and the Ph.D. degree in computer science from the University of Southern California, in 2004, advised by Prof. J. Heidemann.

From 1998 to 2004, he was with the Computer Networks Division, ISI, as a Graduate Research Assistant. During this period, he maintained and contributed codes to the popular network simulation software ns-2. From 2004 to 2007, he was with the Network and Pervasive Computing Program, National ICT Australia, Sydney, as a Researcher, and lead the effort of building one of the world's first outdoor wireless test beds for traffic light communication in the context of the Smart Transport and Roads Project (STaR in press). He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University.

CHUAN-SHENG LIN received the M.S. degree from National Cheng Kung University (NCKU), Tainan, Taiwan, in 2010, and the Ph.D. degree from the Institute of Computer and Communication Engineering, NCKU, in 2018.