Unsupervised indoor localization based on Smartphone Sensors, iBeacon and Wi-Fi

Yi Zhang[1], Jing Chen[2], Wei Xue
School of Internet of Thing Engineering, Jiangnan University
Engineering Research Center of Internet of Things Technology Applications Ministry of Education
Wuxi, China
Email: nezhaxiaozi@outlook.com[1], chenjing@jiangnan.edu.cn[2](corresponding author)

***Abstract*—In this paper, we proposed UILoc, an unsupervised indoor localization scheme that uses the combination of smartphone sensors, iBeacons and Wi-Fi fingerprints for reliable and accurate indoor localization with zero labor cost. Firstly, compared with fingerprint-based method, UILoc system can build the fingerprint database automatically without any site survey and the database will be applied in the fingerprint localization algorithm. Secondly, since the initial position is vital to the system, the UILoc will provide the basic location estimation through the PDR method. To provide accurate initial localization, this paper proposed an initial localization module, a weighted fusion algorithm combined KNN algorithm and Least squares algorithm. In UILoc, we also designed a reliable model to reduce the landmark correction error. The experimental results show that the UILoc can provide accurate positioning and the average localization error is about 1.1 meters in the steady state and the maximum error is 2.77 meters.**

Keywords*—indoor localization; iBeacon; initial localization; reliable model; fingerprint database

I. Introduction

Indoor localization has got a lot of attention for its wide application since people developed various methods for localization such as Wi-Fi [1-2], RFID [3], and so on. However, Wi-Fi is restricted by fixed power and indoor environment and RFID needs corresponding equipment. Bluetooth low energy (BLE) has risen suddenly and strongly in the last two years for its low energy consumption and low cost [4-6]. iBeacon, a kind of equipment based on the BLE, inherits the advantages of BLE which makes it much better than Wi-Fi in terms of energy consumption and cost. What's more, deployment of iBeacon is much easier than RFID and Wi-Fi. Thus, iBeacon can setup a network for indoor localization easily.

Received Signal Strength (RSS) of Wi-Fi is used for localization frequently nowadays. Fingerprinting is a RSS-based localization and has been the most accurate approach to indoor localization among other RSS-based techniques [7], since it can help solve the problem of multipath and NLOS propagations. However, fingerprinting approach is labour-intensive, it needs plenty of labors to collect the RSS.

Since the shortcoming of the fingerprinting approach above, recently many papers [8-11] have put forward the idea that construct fingerprint by crowdsourcing. The database is updated by accepting the feedback from the user. However, this idea needs large quantities of feedback information to ensure the accuracy, so enough cooperate is important. Meanwhile, the feedback from users may have some mistakes. Therefore, the stability of crowdsourcing is low.

Another widely adopted localization technique is Pedestrian Dead Reckoning (PDR) [12]. It includes the following modules: step detection, step length and walking direction. But PDR has the problem of accumulative error and can't estimate the initial point.

Some research and papers [5,13-16] use landmarks to correct user's location, but these landmarks can not ensure whether user is in landmarks, they can only guess that user is close to the landmarks. So the system always tries to haul back to the location of landmarks at the time of unnecessary.

In this paper, we proposed UILoc, a new framework of combining PDR, iBeacon and Wi-Fi. For the PDR approach, since relative information is employed, it will drift with walking distance. Therefore, this paper intend to employ iBeacon measurements to correct the drift of the PDR approach occasionally. The fusion algorithm is applied to estimate the initial position, also the fingerprint database auto-building module is designed to improve the accuracy of the fusion algorithm. The Reliable Model is designed to provide reliable location estimation and reduce the error scope during the landmark rectification stage.

The rest of the paper is organized as follows. The system overview is presented in Section 2, Section 3 shows the detail of UILoc. The prototype implementation and experiments are discussed in Section 4. We conclude the work in Section 5.

II. Summary of UILoc

A. System architecture

The system architecture of the UILoc is shown in Fig.1, which mainly consists of PDR module, particle filter module, reliable module, fingerprint database auto-building module and initial localization module. Compared to fingerprint approaches, the UILoc build the fingerprint database automatically in the online phase, which is similar to Zee [9], instead of sampling by plenty of labours in the offline phase. In UILoc, the location is

978-1-5386-3755-5/18/$31.00 ©2018 IEEE

calculated by PDR. Particle filter module is used in PDR to correct the walking length and walking direction. Because PDR method needs the initial position, the UILoc uses initial localization module to provide the initial position. The UILoc uses reliable module to provide accurate correction service to solve the problem of accumulated error. Then the UILoc combines position information and fingerprint information to establish the fingerprint database which is served for initial localization module to improve the initial positioning accuracy.

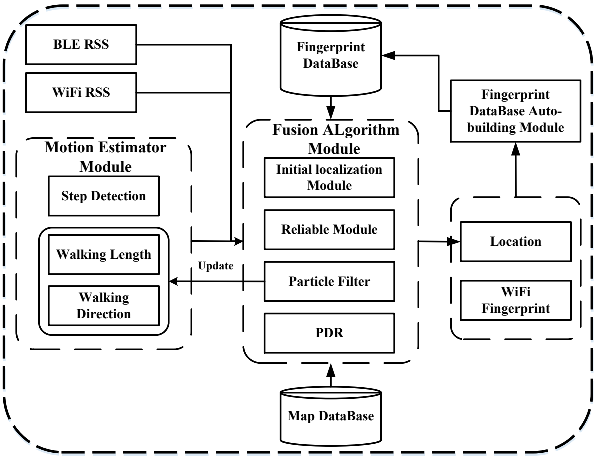

Fig. 1. Overview of UILoc system.

B. Pedestrian Dead Reckoning Module

In the PDR approach, the current position is calculated from the previous position and can be expressed as follows:

$$X_{t+1} = X_t + L_t \begin{bmatrix} sin\theta_t \\ cos\theta_t \end{bmatrix} \tag{1}$$

where X_t is the 2D location of the pedestrian, L_t is the walking length, and θ_t is the walking direction at time of step t. Some critical issues need to be addressed, including walking length estimation and walking direction estimation.

In this paper, we used the AD-FSM algorithm [17] to detect the step, because AD-FSM algorithm can restrain noise interference effectively and provide stable and accurate step detection. And the UILoc utilized particle filter to correct walking direction and walking length.

C. Particle Filter Module

Particle filter [18-19] is used in walking length estimation and walking direction estimation. As every particle has attributes of step length deviation and walking direction deviation, the UILoc can correct the step length and walking direction.

As shown in Fig.2, it is shown how the step length and walking direction deviation is corrected by particle filtering. In the initial phase, all particles are initialized at the initial position as a Gaussian distribution. Every particle has the attribute of step-length-bias and walk-direction-bias. With the user's movement, the position of the particles will be different, and many erroneous particles will die because they hit the wall. The mean of step length deviation and walking direction deviation of the remaining particles are the user's real walk state.

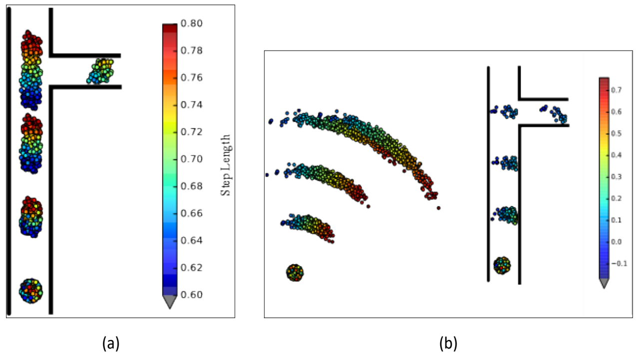

Fig. 2. Showing the use of particles to correct the step length and direction.

D. Reliable Model Module

Reliable model is proposed to provide reliable location estimation and reduce the error scope during the landmark rectification stage. This model will be described in the next section.

E. Fingerprint Database Auto-building Module

Fingerprint database auto-building module combines the position after fusion localization and the fingerprint to update the fingerprint database. This module is designed to serve for initial localization module.

F. Initial Localization Module

Initial position is very important for the whole localization because the initial position influence the position error. Initial localization module combines the fingerprint approach and propagation model localization. And it can provide accurate initial position.

The details of these modules are described in the next section.

III. Proposed Method

A. Fingerprint Database Auto-building Module

Fingerprint database auto-building module build and update the Wi-Fi fingerprint database automatically under unsupervised state. The position information is mainly provided by the fusion algorithm module. The UILoc adopts this module for more accurate initial position.

Fingerprint database has no data in the initial stage. At the beginning, the positioning of the system depends mainly on the propagation model of iBeacon. Then UILoc can estimate position of user by PDR which is combined with particle filter. Also the UILoc deploys some iBeacons as landmark points in indoor environment for correcting cumulative error caused by PDR. With the continuous construction of the fingerprint database, the initial location provided by initial localization module which fuses fingerprint and propagation model method will become more and more accurate.

The main process of building the fingerprint database is as follows:

1) Firstly, since we can get the location (loc_{online}) and Wi-Fi information through this system at online phase, then we need find the fingerprint in the fingerprint database whose *location can be expressed as* loc_{db}*, and the distance between these two points needs to be less than 1 meter. Otherwise we insert the information into fingerprint database directly.*

2) Since we can find the fingerprint in the database, then the position and the value of RSSI in the fingerprint will be averaged and update the database.

B. Initial Localization Module

This paper proposed initial localization module to reduce the error of initial position. Because there is no fingerprint database in the initial stage, the approach commonly used in iBeacon-based position is propagation model method. Most users focus on the location of the first localization when they use the positioning service. And initial localization is very important to the localization system. Although the deviation of the initial localization is large when the fingerprint database in the initial stage is not complete, with the completion of the fingerprint database, localization system based on the fingerprint can provide more accurate initial position relatively. A new method which is the combination of propagation method and fingerprint method is proposed to improve the accuracy. This algorithm is briefly explained in the following subsections.

As shown in Fig.3, when user requests the localization at the point O, UILoc system will scan all signals in the environment firstly. Then the signal strength of each iBeacon will be transformed into distance. The relationship between RSS value and distance is obtained by polynomial fitting [4].

The mapping relations between RSS data and distance in our environment is defined as following:

$$\text{Distance} = 0.568 * rssi + 0.0079 * rssi^2 + 10.2 \quad (2)$$

Since smartphone has the data of user's orientation and the system has all positions of landmarks (iBeacons), we can find the landmark which has strongest RSS. Then the initial localization module use landmark position and the orientation to infer the possible position of the user.

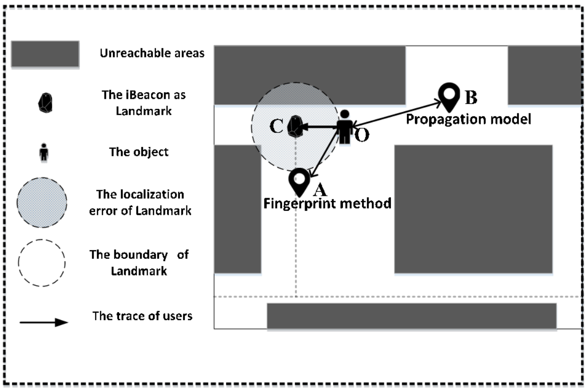

Fig. 3. The principle of Initial Localization Module.

According to KNN calculation formula, we calculate the position of A. And the formula is defined as following:

$$D_j = \sqrt{\sum_{i=1}^{n}\left(RSS_{online}(i) - RSS_j(i)\right)^2} \quad (3)$$

where D_j represent Euclidean distance of the *j*th RP (Reference Point), i is the number of AP.

In the NN algorithm, the device is estimated at the most similar measuring point (i.e. with the minimal Euclidean distance). The KNN computes the center of k closest neighbors and uses Euclidean Distance as the weight of each neighbor. The formula is defined as following:

$$Loc_{KNN} = \frac{1}{K}\sum_{i=1}^{k} loc_i \quad (4)$$

Of course, through the least squares method we can also estimate the position. The specific formula is as follows:

We assume that the user's location is $O(x, y)$ and iBeacons which can be detected by user are located at (x_i, y_i), so the distances between user and iBeacons can be expressed as:

$$r_i^2 = (x_i - x)^2 + (y_i - y)^2 \quad (5)$$

$$r_i^2 = K_i - 2x_i x - 2y_i y + x^2 + y^2 \quad (\text{i} = 1,2,\cdots,\text{N}) \quad (6)$$

where $K_i = x_i^2 + y_i^2$.

We define $r_{i,1}$ as follows:

$$r_{i,1} = r_i - r_1 \quad (7)$$

$$r_1 = r_{i|i=1} = \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \quad (8)$$

So we can get:

$$r_i^2 = r_{i,1}^2 + 2r_{i,1}r_1 + r_1^2 \quad (9)$$

Combine the equation 6 with equation 9, we can get

$$r_{i,1}^2 + 2r_{i,1}r_1 = -2x_{i,1}x - 2y_{i,1}y + K_i - K_1 \quad (10)$$

where $x_{i,1}$ and $y_{i,1}$ represent $x_i - x_1$ and $y_i - y_1$. Note that the unknowns r_1, x and y has a linear relation. So can be expressed as matrix form:

$$\frac{1}{2}\left[r_{i,1}^2 - (K_i - K_1)\right] = -\left(r_{i,1}r_1 + x_{i,1}x + y_{i,1}y\right) \quad (11)$$

$$-\begin{bmatrix} x_{2,1} & y_{2,1} & r_{2,1} \\ x_{3,1} & y_{3,1} & r_{3,1} \\ \vdots & \vdots & \vdots \\ x_{N,1} & y_{N,1} & r_{N,1} \end{bmatrix}\begin{bmatrix} x \\ y \\ r_1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} r_{2,1}^2 - (K_i - K_1) \\ r_{3,1}^2 - (K_i - K_1) \\ \vdots \\ r_{N,1}^2 - (K_i - K_1) \end{bmatrix} \quad (12)$$

This can be expressed as:

$$GZ = H \quad (13)$$

So the location of user is calculated as:

$$Loc_{LS} = Z = (G^T G)^{-1} G^T H \quad (14)$$

Then the fusion algorithm is expressed as follows:

$$D_{fp} = {}^{1}/_{e^{dis_{KNN}}} \tag{15}$$

$$D_{ls} = {}^{1}/_{e^{dis_{LS}}} \tag{16}$$

where dis_{KNN} represents the distance of $|OA|$, dis_{LS} represents the distance of $|OB|$. Since the larger distance indicates that the greater error, we use the exponent of e to amplify the deviation, and then get the reciprocal of it because the greater of the error, the smaller of the weight to be given. Then the weights need to be normalized as follows:

$$W_{fp} = \frac{D_{fp}}{D_{fp}+D_{ls}} \tag{17}$$

$$W_{ls} = \frac{D_{ls}}{D_{fp}+D_{ls}} \tag{18}$$

The final position after normalizing the weight is estimated as follows:

$$Loc_{out} = W_{fp} * Loc_{KNN} + W_{ls} * Loc_{LS} \tag{19}$$

C. Reliable Module

Reliable Module can shrink the error which is produced by landmarks' correction and provide stable and accurate localization. We proposed reliable model to provide reliable location estimation and reduce the error scope during the landmark rectification stage.

Figure 4 depicts the scenario using landmark in indoor environment. There are many landmarks which can correct users' locations in the inner area of a building. But the identify-boundary of a landmark maybe a rough scope that position can not be measured precisely. And the landmark can't make corrections efficiently when the user keeping close to the landmark, because of the fluctuation of BLE signal. Landmarks may correct the error caused by PDR, but the error from itself still remains. So our aim is to minimize the error from landmarks.

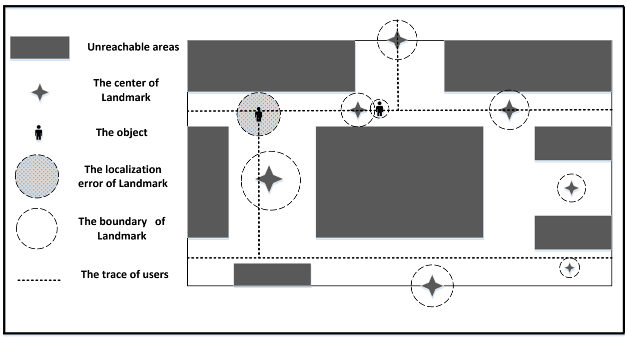

Fig. 4. The scenario using Landmarks.

Most of the papers [5, 14] use the method to detect the strongest signal. When the algorithm finds that the value of RSS is greater than the threshold, the user's location will pull back to the center point of landmark, but at that time the user may just come close to the landmark area or has gone out of the landmark area.

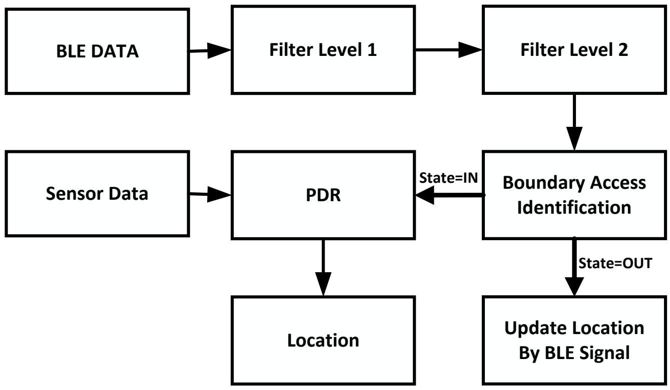

Fig. 5. The design of the Reliable Model.

To smooth the signal and restrain noise, we designed reliable model which can improve the location accuracy. As Fig.5 shows the algorithm consists of the boundary access identification module, L1 and L2 filter and the PDR module. Firstly, the L1 level filter is utilized to eliminate the outliers. Then, the L2 level filter is used to restrain noise and smooth signal. Next, the boundary access identification module judges the states of users about specific landmarks, if a user hasn't entered into the landmark area, the user's location will be corrected by landmark and the location will be produced by PDR. Lastly, once the user has got into the reliable model, the PDR module will provide the output of localization.

Figure 6(a) shows the process that the user comes close to the landmark then goes away from the landmark. Because of the fluctuation of BLE RSS, as the red circle labeled in the figure, at that time the user already gone away from the landmark, but the value of RSS is larger than the threshold again, so the algorithm will correct the user's position. In this case, the raw data will enhance the volatility and lower the robustness in the process of localization.

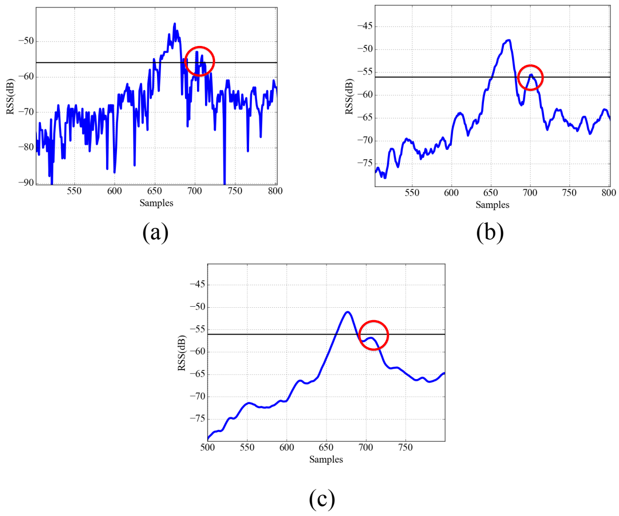

Fig. 6. The signal curves of the iBeacon when the user comes close to then goes away from the landmark. Fig (a) raw data of BLE. Fig (b) data of BLE after L1 filter. Fig (c) data of BLE after L2 filter.

Firstly, we considered the sampling frequency is 10 Hz, so the length of window for dynamic filter was set to 10. The L1 level filter is designed as below:

$$\begin{cases} \boldsymbol{RSS} = w(rss_{i-9}, rss_{i-8}, \ldots, rss_i) \\ \boldsymbol{RSS}' = filter(\boldsymbol{RSS}) \\ rss_i = \frac{1}{n}\sum_{j=1}^{n} rss_j \end{cases} \tag{20}$$

where $\boldsymbol{RSS}$ is a dynamic vector about the value of rss. We used Tukey's test method to remove the outliers for $\boldsymbol{RSS}$ and get $\boldsymbol{RSS}'$. Finally, the mean value of $\boldsymbol{RSS}'$which will be given to rss_i is calculated, where rss_i represents the rss at the moment i. The Tukey's test method is expressed as following:

$$\begin{cases} \boldsymbol{min} = Q_1 - 1.5(Q_3 - Q_1) \\ \boldsymbol{max} = Q_3 + 1.5(Q_3 - Q_1) \\ \boldsymbol{min} < rss_k < \boldsymbol{max} \end{cases} \tag{21}$$

where Q_1 and Q_3 are expressed in Equation (22). And each reasonable rss_k in $\boldsymbol{RSS}$ must be between the $\boldsymbol{min}$ and the $\boldsymbol{max}$.

$$\begin{cases} Q_1 = sorted(\boldsymbol{RSS})_{1/_{4*\boldsymbol{length}(\boldsymbol{RSS})}} \\ Q_3 = sorted(\boldsymbol{RSS})_{3/_{4*\boldsymbol{length}(\boldsymbol{RSS})}} \end{cases} \tag{22}$$

After the L1 level filter, the signal curve is shown in Fig.6(b). It is clear that there are a lot of Gaussian noise which will influence the localization, so we proposed to use Kalman filter to eliminate the Gaussian noise. The algorithm contains two processes: predicting and updating.

Firstly, the Kalman filter has some parameters that have to be determined. As the value of RSSI changes randomly, therefore, the transition matrix F and the measurement matrix H are set to one. Furthermore, there is no external control input, so Bu_{t-1} is also set to zero.

Predicting:

$$\begin{cases} rss_{t|t-1} = Frss_{t-1|t-1} + Bu_{t-1} \\ P_{t|t-1} = P_{t-1|t-1} + Q_{t-1} \end{cases} \tag{23}$$

Updating:

$$\begin{cases} K_t = P_{t|t-1}H_t^T\left(H_tP_{t|t-1}H_t^T + R_t\right)^{-1} \\ rss_{t|t} = rss_{t|t-1} + K_t(rssi_t - H_t rss_{t|t-1}) \\ P_{t|t} = (1 - K_tH_t)P_{t|t-1} \end{cases} \tag{24}$$

In Equation (24), P is the estimate covariance. In UILoc, the initial value of P is set to $P_{t-1|t-1} = 1.0$. K is the Kalman gain. The values $Q = 0.015$ and $R = 1.5$ are determined experimentally.

After L1 and L2 level filter, the signal curve is shown in Fig.6(c).

The basic principles of the reliable module are described in detail above. Algorithm 1 is the pseudo-code form of the reliable module.

The input data of the reliable module is the BLE signal: $\{UUID_i: RSSI_i\}$ and the direction data in the environment. The output is the location of the user. Once the reliable module get the data, the BLE data will be filtered by L1 and L2. Then we need to set the states of users about specific landmarks from IN to OUT when the RSSI from specific beacon is less than -65dB (about 6m by Eq.(2)). At that moment, we can think that the user has gone out of the landmark area completely. Next, we need to find the beacon as $beacon\_i$ which has the strongest signal. If the state of $beacon\_i$ is OUT and the strength of $beacon\_i$ is greater than -52dB(about 2m by Eq.(2)), then we need to update user's location as follows. Also we need to set the state of $beacon\_i$ to IN, so even if the user walks into the landmark area, the reliable module can prevent the problem of repeated positioning. And in the landmark area, the location will be calculated by PDR.

Algorithm 1: Reliable Module

```
Input: ble_data, ori
Output: location
Loop:
ble_data = filteredByL1(ble_data)
ble_data = filteredByL2(ble_data)
/* boundary access identification module start*/
beacon_i = find the beacon whose state is IN
if beacon_i_state is IN and beacon_i_rssi < -65:
    beacon_i_state = OUT
beacon_max = find the beacon which has max rssi
if beacon_max_state is OUT and beacon_max_rssi > -52:
    distance = map beacon_max_rssi to diatance
    X = beacon_max_postion_x - distance * sin(ori)
    Y = beacon_max_postion_y - distance * cos(ori)
    beacon_max_state = IN
    return X,Y
End Loop
```

D. UILoc Algorithm Explanation

All the sub modules have been described in detail above. Next, we will describe the details of the module integration. The algorithm 2 is the pseudo-code form of the UILoc.

Algorithm 2: UILoc

```
Input: data
Output: location
Loop:
use data to update acc,ori,wifi_entry,ble_entry and ble_queue_window by
data's type
if type is ACC:
  acc = use kalman filter to filter acc
  if AD-FSM.countStep(acc) and location is not null:
    location = use particle filter to estimate the location
else if type is BLE:
  if location is not null:
    location = ReliableFilter.process(ble_queue_window,ori)
else if type is WIFI:
  if location is null:
    location = InitLocalization(wifi_entry,ble_entry)
    initialize the particle filter module
  else:
    update fingerprint database by location and wifi_entry
return location
End Loop
```

The input of the UILoc is data $\{type: senor\_data\}$, including acceleration data, orientation data, ble data and wifi data. The output of the algorithm is the location of the user.

Firstly, the parameter ble_entry is the data like$\{UUID_i: RSSI_i\}$. And $wifi\_entry$ is a fingerprint data like

$\{MAC_i\!:RSSI_i\}$ and the parameter ble_queue_window is an array contains 10 historical ble data.

If the type of data is ACC, we do Kalman filter for acc data, then we use AD-FSM to determine whether a step is formed. If the $location$ has been initialized and the acc data formed a step, then the location be calculated by particle filter which has fused PDR module. If the data is type of BLE, we use reliable module to provide a stable localization. At last, if the data is type of WIFI, we use initial localization module to estimate user's initial location. We also can create and update fingerprint database.

IV. Experimental Work And Results

In this section, we introduce the experiment setup, perform the proposed model in a real building (i.e. other people may walk through this area during the experiments), then present and discuss the experimental results.

A. Experiment Setup

To evaluate the performance of the UILoc model, we conducted experiment on one floor of a typical office building at the campus of JNU (Jiangnan University). The size of the target area is about 3000 m^2($60m$ by $50m$). Figure 7 shows the layout of the research lab. We deployed 18 beacons (average: 1 beacon per $10m$) in the experiment. All these beacons were installed on the wall at a height of approximately $1.5m$. To balance the power consumption and accuracy, each beacon was set to 10Hz sample rate with 0 dB transmit power. The device involved in the experiment is a HUAWEI P9 smartphone running Android 6.0 operation system. In order to compare the fingerprint localization algorithm, we perform our experiment with 184 RPs. The average distance between 2 RPs is $1m$.

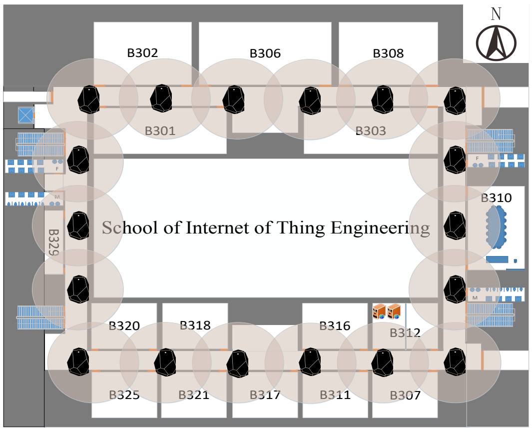

Fig. 7. Layout of the experimental environment.

B. Performance Evaluation

As shown in Fig.8, it depicted the process of the fingerprint database construction. In this experiment, there are some users who moved with smartphone in the indoor environment randomly. And the location and fingerprint data will be continuously collected to our server. From the figure 8 we can find that the number of RP point in the fingerprint database is very few at the beginning. After the UILoc system started for some time, the number of reference point increases rapidly, as shown in (a), (b) and (c), until the environment is evenly distributed with reference points. At last, as shown in (g) and (f), the fingerprint database reaches a steady state.

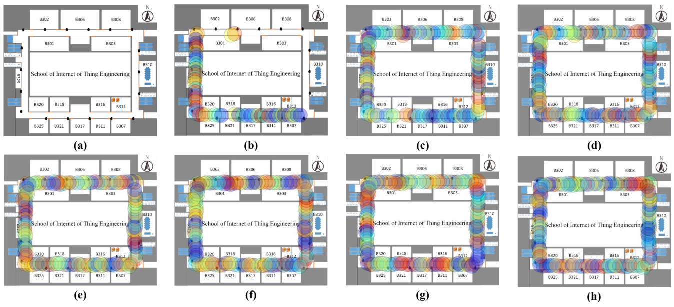

Fig. 8. The process of the fingerprint database construction.

Figure 9 shows the average localization error and the standard deviation of our initial localization model under different 8 stages corresponding to the 8 graphs in Fig.8. As expected, the accuracy of initial localization model is very low in phase 1 and phase 2, because the location at this stage is mainly provided by the least squares method. From phase 3 to phase 5, the number of RP in the database was growing rapidly, so the localization error and the standard deviation are rapidly decreased. At last, positioning error converges at around 2.15 meters.

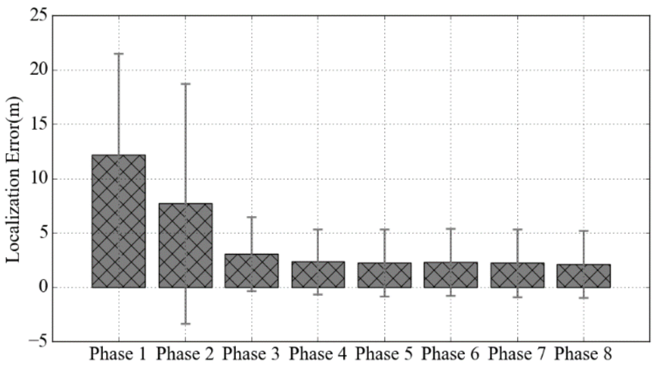

Fig. 9. The average localization error and the standard deviation of the initial localization model under different stages.

Figure 10 is based on the localization error of each initial localization algorithm. It can be seen that the error interval of LS algorithm is the largest because the signal fluctuations. Compared to other algorithms, the number of outliers in our algorithm is the least. Because of the weighted fusion algorithm can suppress the impact of signal fluctuation. In UILoc, our initial localization model can provide a more accurate initial positioning.

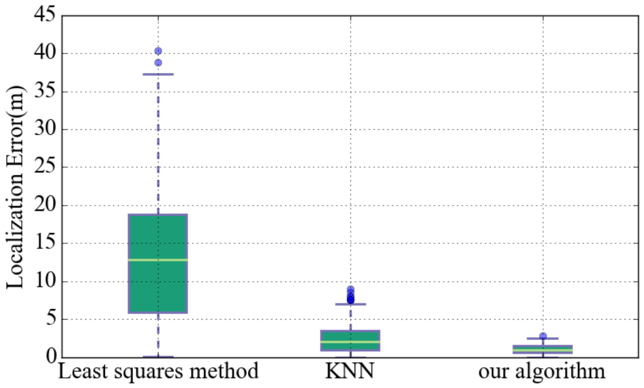

Fig. 10. The localization error of different initial localization algorithm

In order to analyze the localization error of UILoc, we evaluate UILoc algorithm in a real experimental environment. In the experiment, the user walk in the experimental area with a fixed step length, then we use the PDR, PDR and BLE, PDR, BLE and PF (Particle Filter), PDR, BLE and RM (Reliable Model) and our method (ie SuLoc) for trajectory estimation, all of these algorithms are briefly described as PDR, PB, PBP, PBR and UILoc.

Figure 11 shows the intuitive trajectory estimations by some algorithms in our experimental area. In the experiment, we compared the PDR, PBR and UILoc. It can be seen that the cumulative error of the PDR is very serious in the absence of landmarks' correction, but it is shown that the PDR provide continuous position estimation. As the green track shown, we can use landmarks to restrain the cumulative error of PDR. As shown in the bottom right corner of the Fig.11, the PBR algorithm has the phenomenon of repeated correction. But in UILoc, the reliable model prevents the error caused by repeated correction.

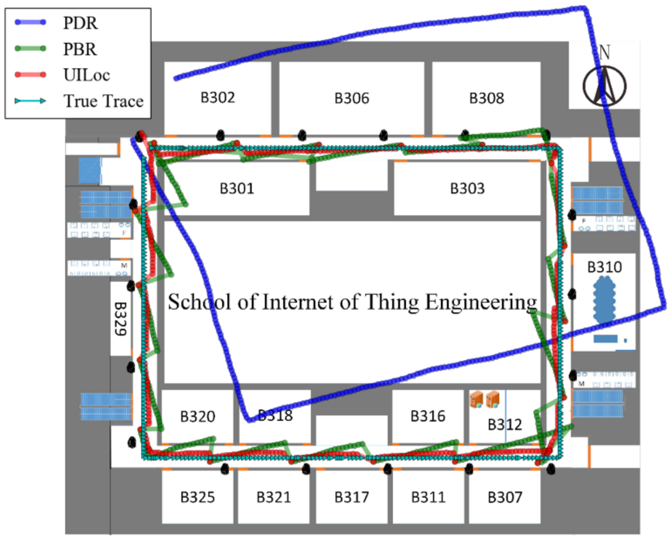

Fig. 11. Estimated trajectories with different algorithms.

Fig.12 describes the real-time localization errors of indoor trajectories using PDR, PB (Landmark with PDR), PBR and UILoc. As can be seen from the figure, without the aid of the reliable model, PB's localization error is significantly larger than PBR algorithm. As the red line shows, in the initial positioning phase, the initial localization error of PBR is much larger than UILoc. So UILoc system can provide more stable position estimation.

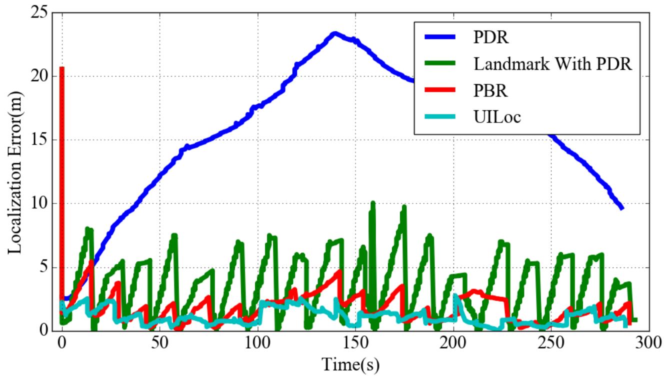

Fig. 12. Localization errors of indoor trajectories using different algorithms.

Table 1 is a summary of localization errors of different algorithms. UILoc's positioning accuracy has been greatly improved compared to the fingerprint positioning algorithms e.g. KNN [20], UnLoc [21], Zee [9]. In algorithm of KNN, the mean of the positioning error is 2.36 meters; In UnLoc[21], the mean of the positioning error is 2.0 meters. In Zee algorithm, the mean error is about 5 meters. Without the reliable model module, the positioning error of the PB algorithm is 3.06*m*, after adding the reliable model module, the localization error is reduced to 1.77m. After adopting reliable model and Initial Localization Model to UILoc, the average positioning error is reduced to 1.11 m and the error of 50% samples is kept within 1 m. Also in this experiment, we get the maximum positioning error of 2.77 meters.

TABLE 1
POSITIONING ERROR OF DIFFERENT METHODS

Algorithm	50%	75%	Mean	RMSE
PDR	16.52	19.29	15.56	16.41
PB	2.75	4.60	3.06	3.76
PBP	2.75	4.50	3.04	3.74
PBR	1.49	2.44	1.77	2.26
KNN	2.0	3.5	2.36	2.97
UILoc	0.96	1.47	1.11	1.26

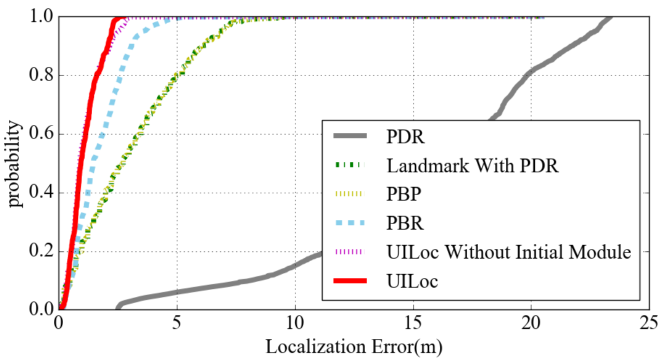

Fig. 13. CDF of location error with different algorithms.

Figure 13 is the cumulative probability graph of each positioning algorithm. Due to the PDR cumulative error, the convergence rate of the PDR is very slow. The convergence rate of PBR is obviously improved after adding RM module. We can see from the figure, the convergence rate of UILoc is faster than that of UILoc without initial module. And the UILoc can keep the positioning error within the range of $3m$ steadily.

At last, we implemented the UILoc on the Android platform and you can get experimental videos from our server [22, 23].

V. CONCLUSION AND FUTURE WORK

In this work, we proposed the UILoc, which builds and updates the fingerprint database automatically. It provides accurate initial position estimation by Initial Localization Module and provides stable and accurate localization by reliable model. Firstly, the popular PDR approach has the well-known drift problem. Since iBeacons can provide a high localization accuracy and can be easily deployed in real situations, we corrected the drift of the PDR approach by iBeacons. Since we used the PDR method to provide the basic position estimate, the accuracy initial position estimation is very important. We proposed the Initial Localization Module (a weighted fusion algorithm that combined fingerprint method and least squares method) to estimate user's initial location. Lastly, the popular Landmark approach just detect the strongest signal and correct user’s position, which may have the problem of repeated correction and result in extra localization error or the instability of localization, so we proposed the reliable model which can provide stable and accurate localization. The preliminary experiment results show that UILoc achieves low human cost, rapid system deployment and high positioning accuracy. Results show that the UILoc achieved an average error of 1.11 meters and can provide the initial localization with the average error of 2.15 meters.

Our ongoing research focuses on making UILoc feasible and pervasive to various applied environments and buildings.

ACKNOWLEDGMENT

This research was supported by the Jiangsu Province Science Foundation for Youths (Grant No. BK20150159) and also supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. SJCX17_0509)

REFERENCES

[1] Bahl P, Padmanabhan V N. RADAR: An in-building RF-based user location and tracking system[C]//INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Ieee, 2000, 2: 775-784.

[2] Yang C, Shao H R. WiFi-based indoor positioning[J]. IEEE Communications Magazine, 2015, 53(3): 150-157.

[3] L. Ni, Y. Liu, Y. Lau, and A. Patil. LANDMARC: Indoor location sensing using active rfid. Wireless Networks, 2004, 10(6):701–710.

[4] Zhuang Y, Yang J, Li Y, et al. Smartphone-based indoor localization with bluetooth low energy beacons[J]. Sensors, 2016, 16(5): 596.

[5] Chen Z, Zhu Q, Jiang H, et al. Indoor localization using smartphone sensors and iBeacons[C]//Industrial Electronics and Applications (ICIEA), 2015 IEEE 10th Conference on. IEEE, 2015: 1723-1728.

[6] Zou H, Jiang H, Luo Y, et al. BlueDetect: An iBeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service[J]. Sensors, 2016, 16(2): 268.

[7] Han D, Jung S, Lee M, et al. Building a practical Wi-Fi-based indoor navigation system[J]. IEEE Pervasive Computing, 2014, 13(2): 72-79.

[8] Lim J S, Jang W H, Yoon G W, et al. Radio map update automation for WiFi positioning systems[J]. IEEE Communications Letters, 2013, 17(4): 693-696.

[9] Rai A, Chintalapudi K K, Padmanabhan V N, et al. Zee: Zero-effort crowdsourcing for indoor localization[C]//Proceedings of the 18th annual international conference on Mobile computing and networking. ACM, 2012: 293-304

[10] Yang S, Dessai P, Verma M, et al. Freeloc: Calibration-free crowdsourced indoor localization[C]//INFOCOM, 2013 Proceedings IEEE. IEEE, 2013: 2481-2489.

[11] Wu C, Yang Z, Liu Y. Smartphones based crowdsourcing for indoor localization[J]. IEEE Transactions on Mobile Computing, 2015, 14(2): 444-457.

[12] Kang W, Han Y. SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization[J]. IEEE Sensors journal, 2015, 15(5): 2906-2916.

[13] Faragher R, Harle R. Location fingerprinting with bluetooth low energy beacons[J]. IEEE journal on Selected Areas in Communications, 2015, 33(11): 2418-2428.

[14] Chen Z, Zhu Q, Soh Y C. Smartphone inertial sensor-based indoor localization and tracking with iBeacon corrections[J]. IEEE Transactions on Industrial Informatics, 2016, 12(4): 1540-1549.

[15] Golestanian M, Poellabauer C. Indoor localization using multi-range beaconing: poster[C]//Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2016: 397-398.

[16] Lee S H, Lim I K, Lee J K. Method for improving indoor positioning accuracy using extended kalman filter[J]. Mobile Information Systems, 2016, 2016.

[17] WANG Gechao, LIANG Jiuzhen, CHEN Jing et al. Step counting algorithm based on FSM using acceleration differential[J]. Journal of Frontiers of Computer Science and Technology, 2016, 10(8): 1133-1142.(in Chinese)

[18] Davidson P, Collin J, Takala J. Application of particle filters for indoor positioning using floor plans[C]// Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010. IEEE, 2010:1-4.

[19] Widyawan, Klepal M, Beauregard S. A backtracking particle filter for fusing building plans with PDR displacement estimates[C]// Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on. IEEE, 2008:207-212.

[20] Moghtadaiee V, Dempstera A G. Indoor location fingerprinting using FM radio signals[J]. IEEE Transactions on Broadcasting, 2014, 60(2): 336-346.

[21] Wang H, Sen S, Elgohary A, et al. No need to war-drive: unsupervised indoor localization[C]//Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM, 2012: 197-210.

[22] Yi Zhang. UILoc experiment[DB/OL]. https://youtu.be/o6880f2uv9o.

[23] Yi Zhang. Summary of different localization algorithms [DB/OL]. https://youtu.be/kLvYlNMLtww.

978-1-5386-3755-5/18/$31.00 ©2018 IEEE