

FineLoc: A Fine-grained Self-calibrating Wireless Indoor Localization System

Xinyu Tong, Ke Liu, Xiaohua Tian, *Member, IEEE*, Luoyi Fu, *Member, IEEE*, Xinbing Wang, *Senior Member, IEEE*

Abstract—Self-calibrating wireless indoor localization systems construct the radio map even the indoor floor plan automatically, which avoids the labor-intensive site survey process; however, existing systems utilizing the feature of Wi-Fi signals can only provide coarse-grained indoor maps, which hinders improvement of localization accuracy. In this paper, we present FineLoc, a fine-grained self-calibrating localization system based on the freely-deployed Bluetooth low energy (BLE) nodes and crowdsourced data, which can profile more detailed layout information of the indoor space. We first reveal that existing systems can only generate inaccurate floor plans owing to the coarse-grained Wi-Fi reference information. Then we utilize the increasingly popular BLE beacon nodes as the source of reference information, with which a series of dead-reckoning optimization and new schemes particularly for finer-grained indoor map construction are presented. We implement a prototype FineLoc system, which is deployed in around $11,000m^2$ areas. Our experimental results with the prototype show that FineLoc can achieve 80% localization errors within $1.6m$, $1.4m$ and $1.1m$ in the library, classroom building and office building respectively, with an average density of deployed BLE nodes less than $2.6/100m^2$.

Index Terms—Simultaneous localization and mapping (SLAM), localization, bluetooth low energy (BLE)



1 INTRODUCTION

THE past decades have witnessed extensive research work on wireless indoor localization [1]–[6]. Although the recently proposed techniques leveraging the wireless channel state information (CSI), Radio Frequency Identification (RFID) and the acoustic signal can achieve amazing accuracy, the received signal strength indicator (RSSI) based indoor localization is a more practical solution for large-scale deployment. This is because the special hardware enabling those high-accuracy localization techniques, such as special APs that can retrieve CSI [7]–[9], the expensive RFID readers [10] and acoustic anchors [11] are not widely deployed. By contrast, the RSSI is handily available from commodity mobile devices, and the RSSI based localization approach just utilizes pervasive wireless infrastructure such as Wi-Fi and iBeacon [1], [6], [12].

The major obstacle hindering large-scale deployment of the RSSI based indoor localization system is the tedious and costly offline training process, which requires manual efforts for radio map construction and periodical calibration [1], [6], [13]. To address the issue, mechanisms are proposed to construct and update the radio map in a crowdsourcing manner [4], [14], [15], where the inertial measurement unit (IMU) embedded in the users’ mobile devices is exploited. While such schemes enable radio map construction with crowd workers’ non-participatory opportunistic sensing, they require precise information of the floor plan to derive landmarks [14], [16], [17]; however, both the floor plan and the landmarks are not always easily available in practice.

Self-calibrating indoor localization systems are capable of constructing the radio map and even the floor plan automatically [18]–[20], where the opportunistically sensed data from users are smartly combined. The core idea of such systems is to exploit the notable feature of ambient Wi-Fi signals serving either as landmarks in the pathway environment [18] or as labels of users’ trajectories [19], [20]. However, such systems can only construct particular floor plan in a coarse granularity, where pathways are aggregated into line segments, thus resulting in $1 \sim 3m$ average localization error in favored environments. We note that the recently proposed BatMapper system can construct the floor plan with 80% room geometry error within $0.3m$ [13]; however, the acoustic sensing based technique is constrained by the short transmission range of acoustic signals, and it still requires the user-participatory measurement, which obstructs the large-scale deployment.

In this paper, we present FineLoc, a fine-grained self-calibrating indoor localization system based on the RSSI of freely-deployed bluetooth low energy (BLE) beacon nodes and the IMU data, which can derive more detailed layout information of the indoor space with achieving the 80% localization error of $1.6m$ even in unfavored indoor spaces. Our contributions are as follows.

First, we reveal the fundamental reason why existing self-calibrating indoor localization systems can only generate coarse-grained floor plans (§Section 3). The reason is that the features of the Wi-Fi signal serving as the cornerstone reference information in those systems function well only in certain conditions, where the length of the user’s trajectory must be notably greater than the distance between the trajectory and the Wi-Fi AP. Moreover, due to the comparatively large coverage of the Wi-Fi AP, it is difficult to distinguish the relative positions of different trajectories; therefore, Wi-Fi based systems aggregate multiple trajectories into a single

- Xinyu Tong, Ke Liu, Xiaohua Tian, Luoyi Fu and Xinbing Wang are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200240. E-mail: {tongxinyu1993, kqft2540253972, xtian, yiluofu, xwang8}@sjtu.edu.cn.

line segment, which is unable to derive detailed layout information such as rooms, walls and obstacles. Wi-Fi signals are unable to provide fine-grained reference information, which is determined by the design principles of the Wi-Fi system. This motivates us to utilize BLE beacons for reference information, which is also a pervasive infrastructure gaining increasing popularity [21]–[25].

Second, we present design of FineLoc, where the widely used dead-reckoning mechanisms are optimized, and new data fusion schemes particularly for fine-grained map construction are developed (§Section 4). We present a method to recognize the user’s posture of holding the smartphone when performing non-participatory sensing, which provides more precise trace (§Section 5.1). We develop a labeling scheme to associate the BLE beacon nodes with users’ trace, which addresses the issue of unstable BLE reference information due to random radio propagation and heterogeneity of both mobile devices and beacon nodes (§Section 5.2). Then we design an efficient trace merging mechanism, which integrates trace data under different coordinate systems into the skeleton of the indoor map under the unified coordinate system; such mechanism can also detect displacement of BLE nodes and eliminate the incurred error with automatic map update (§Section 5.3). To further eliminate IMU error in direction recognition, we embed our trace revising algorithm into system to improve the reliability (§Section 5.4). Moreover, we propose a map pixel classification scheme, which categorizes areas not covered by trace data into different classes such as pathway, obstacle and wall based on nearby trace information, so that a more detailed indoor map can be derived (§Section 5.5). Finally, we show how FineLoc works (§Section 5.6).

Third, we implement the FineLoc system in three types of indoor spaces covering more than 11,000 m^2 areas, and conduct extensive experiments to validate proposed mechanisms. It shows that FineLoc can construct fine-grained floor plans not only in the environment where small-sized sub-areas densely distributed such as the office building, but also in the open wide environment such as the library; FineLoc can illustrate more detailed information of pathways, rooms and obstacles in the constructed map. A map can be largely constructed after 20 ~ 120mins of free walk; moreover, our system limits 80% localization errors within 1.6m, 1.4m and 1.1m in the library, classroom building and office building environment respectively, with the average density of deployed BLE nodes less than 2.6/100 m^2 (§Section 6).

2 RELATED WORK

Ubiquitous use of mobile devices stimulates the spring-up of wireless indoor localization techniques [1], [6], where the RSSI has been utilized since emergence of Wi-Fi based systems [2], [3]. Those systems first perform offline training associating the RSSI with the location where the data is sampled, and then estimate the user’s location by comparing the user’s reported RSSI with the radio map obtained in the training phase, in order to derive the most matched location. This is known as the fingerprinting based approach, where the issue of labor-intensive training phase has been noted; the corresponding countermeasure is to exploit the radio propagation model to populate part of the training database [2]. Such a model based approach is optimized and applied

to later indoor localization systems [4], [26] in special scenarios; however, a large body of work has shown that the radio propagation model is inaccurate for localization [6].

With more IMU sensors embedded in the cell phone, the IMU data are utilized to save the offline training efforts under the crowdsourcing paradigm, where the basic idea is to analyze IMU data and RSSI to construct the radio map. LiFS [16] uses multi-dimensional scaling to match fingerprints to locations. Zee [14] exploits war-driving to infer locations, which is facilitated by the constraints imposed by the floor plan. WILL [15] leverages the IMU data to infer the user’s location by comparing the corresponding RSSI characteristics and the floor plan. KAILOS provides a crowdsourcing platform and the toolkit for contributors to upload indoor maps and construct radio maps at target buildings [27]. Those systems relieve the site survey overhead, but the ground truth floor plan must be previously acknowledged.

The floor plan however is not always easily available, for which efforts have been dedicated to perform localization without site survey and floor plan. UnLoc [17] leverages the identifiable characteristics in the building as landmarks, and estimate the user’s location with the landmark information and the mobile device’s IMU data. Walkie-Markie [18] proposes to use the dynamic feature of the Wi-Fi signal as the landmark to construct the indoor pathway map. In particular, the landmark is the pathway location at which the RSSI from a certain AP changes from increase to decrease. The changing of the RSSI from Wi-Fi APs also can be used to determine whether two users’ trajectories are from the same path segment, which provides the basis of floor plan derivation in a crowdsourcing manner [19].

Our work in this paper shares the objective of Walkie-Markie and PiLoc; however, we improve the granularity of the floor plan to be constructed and thus improve the localization accuracy. Our system serves general localization activities in contrast to special leader-and-follower navigation needed as in FOLLOWME [28]. Wireless localization techniques based on CSI, RFID, and acoustic signals, as well as that based on visible light [29] can achieve amazing accuracy [8]–[11], which, however, still need site survey and accurate floor plans. We note that the recently proposed BatMapper scheme can construct the floor plan with 80% room geometry error within 0.3m [13]; however, the acoustic sensing based technique is constrained by the short transmission range of acoustic signals, and it still needs involving participatory user measurement and thus is difficult for large-scale deployment. In contrast, we use the crowdsourcing paradigm to aggregate crowd workers’ traces into a map.

FineLoc is a fine-grained self-calibrating indoor localization system. To understand the concept of finer-grained and self-calibrating better, we compare the proposed system with existing ones as shown in Table. 1. We can see that some systems [18] [19] can only generate pathway floor plan without any detailed information. Systems with comparatively finer granularity [30] [31] can distinguish one room from another. If the system can recognize obstacles in the indoor space, then we can say the granularity is even finer. Non-calibration means the necessity of prior knowledge of landmark placement while unique calibration

TABLE 1: Finer-grained Self-calibrating Indoor Localization System Comparison

System Name	FineLoc	Walkie-Markie [18]	PiLoc [19]	Jigsaw [30]	BatMapper [13]	SenseWit [31]
Technology	BLE	Wi-Fi	Wi-Fi	Vision	Acoustic	only INS
Pathway Floor Plan	Yes	Yes	Yes	Yes	Yes	Yes
Sub-room Classification	Yes	No	No	Yes	No	Yes
Obstacle Recognition	Yes	No	No	No	No	No
Non-Calibration	Yes	Yes	Yes	Yes	Yes	Yes
Unique Calibration	Yes	Yes	Yes	Yes	No	No

means that the uniqueness of the landmark can be provided by the system. Among the schemes in Table. 1, Walkie-Markie [18] and PiLoc [19] are based on Wi-Fi and provide both mapping and localization schemes while BatMapper [13], Jigsaw [30] and SenseWit [31] can only sketch floor plan. FineLoc is the only fine-grained self-calibrating indoor localization system.

3 MOTIVATION

Constructing indoor maps requires reference information that can reflect the physical layout of buildings; moreover, the reference information must be uniquely identifiable, and functioning with no need to deploy special infrastructures, considering the scalability and cost. The natural signatures such as the fluctuation of magnetic fluctuation in certain part of the building as in UnLoc [17] need no dedicated infrastructure but can not be uniquely identified in multiple buildings. The spatial or timing feature of Wi-Fi signals can be uniquely identified with the help of the Wi-Fi AP's MAC address [18], [19]; however, such reference information has limited capability to reflect the building's physical layout. This is why the floor plan constructed with Wi-Fi based reference information is with coarse granularity. This section first reveals the fundamental reason why the Wi-Fi based reference information leads to coarse-grained indoor map, and then presents our choice of reference information source for finer-grained indoor map.

3.1 Analysis of Wi-Fi Landmarking

Wi-Fi signals can reflect the building's physical layout to some extent. When a user passes through the covered area of a Wi-Fi AP, the user's mobile device can observe that the RSSI of the AP increase as the user moves closer towards the AP, and decrease as the user moves past the AP. Fig. 1(a) shows the scenario, where it shows that the RSSI increases to a peak point and then decreases as the user moves from A to B . The *RSSI tread tipping point* (RTTP) corresponds to a fixed position on the way that is closest to the AP, which can serve as a landmark to provide location reference information in map construction [18]. In reality, the RTTP can be obviously identified only if the length of trace AB (30 steps) is notably greater than the distance between the AP and the trace (9 steps). In contrast, once the length of trace EF (10 steps) is comparable to the distance between the AP and the trace (7 steps) in Fig. 1(b), we can see from Fig. 1(d) that the RTTP is difficult to identify. The hidden reason is that Wi-Fi RSSI changes slightly when the distance between the user and the AP varies a little. The observation indicates that the Wi-Fi landmarks only appear when the user is walking along a long path, which explains why Walkie-Markie can only construct maps for pathways; because the size of room is usually limited, where the users cannot walk along a long straight path for RTTP identification.

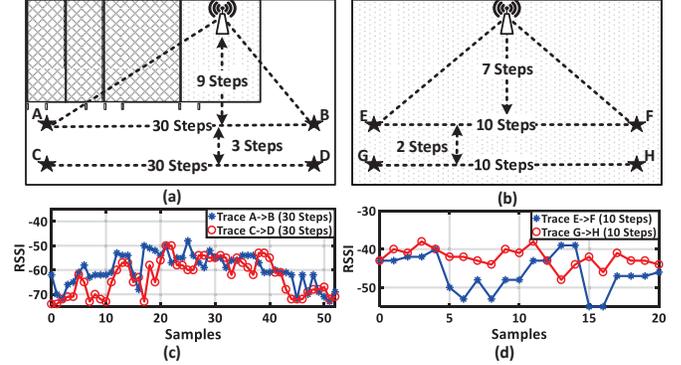


Fig. 1: Wi-Fi RTTP shows less scalability.

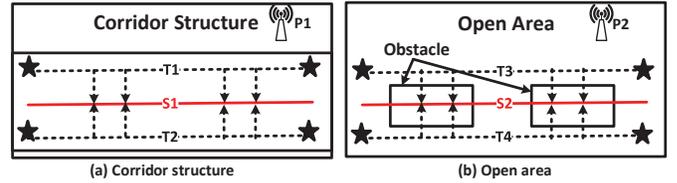


Fig. 2: Existing schemes cannot recognize obstacles due to coarse-grained merging.

In PiLoc [19], the user's mobile device records the observed Wi-Fi RSSI while walking; the recorded time series of the RSSI from different users are compared for correlation, which is a critical reference information to determine whether the RSSI series are observed from the same location in the building. Although claiming independent of landmarks, PiLoc factually leverages the RTTP feature to determine the correlation of two traces in an implicit manner. The similar RTTP observed in two traces is the most prominent feature of the two traces, which only appears when the length of the trace is notably larger than the distance between the AP and the trace. Without the RTTP feature, the correlation between the two traces is dominated by the environment noise, which is unable to determine the trace similarity. Fig. 1(d) shows the scenario, in which the correlation of the two traces is 0.143.

The experimental results presented in Walkie-Markie and PiLoc factually corroborate our analysis. In particular, Figure 4 in [18] and Fig. 7-9 in [19] show that the RTTP appears when the length of the trace is at least 30 steps. This is far beyond the dimension of regular rooms in the building; therefore, both Walkie-Markie and PiLoc focus on constructing pathway maps.

If the RTTP only appears in the long trace, will the reference information based on Wi-Fi signals be suitable for wide open areas such as libraries or museums? Unfortunately, the answer is negative. If the two paths are labeled with the RTTP from the same AP, it will be difficult to determine distances between the paths and the AP respectively. As shown

in Fig.1(c), paralleled trace AB and CD can obtain almost the same RTTP. Thus it's impossible to recognize the relative distance between AB and CD (3 steps). Fig.2(a) illustrates the two traces T_1 and T_2 in real situation, where the space between the two traces can be observed. Due to coarse-grained RTTP recognition, the current solutions in [18], [19] just merge the two traces to form a combined trace as shown in S_1 , which makes the space between T_1 and T_2 unable to be reflected in the resulted floor plan. Regular pathways may be modeled with line segments; however, the trace merging schemes can prevent the constructed floor plan from reflecting the obstacles such as the bookshelf, showcase and booth between two traces, as shown in Fig.2(b), which results in coarse-grained map with little usefulness.

The trace merging method is adopted in both Walkie-Markie and PiLoc, because the nature of the Wi-Fi signal can provide only coarse-grained reference information. The infrastructure Wi-Fi AP is designed to cover a hundred-meter-radius area; APs are normally sparsely deployed to save the cost and avoid MAC layer collisions. Even if some traces can observe strong RSSI with respect to an AP, it is still hard to determine the distance between the trace and the reference point such as RTTP generated by the Wi-Fi AP. Consequently, it is reluctant for the coarse-grained relative location information of the user's trace with respect to instable reference information to provide basis to construct a fine-grained indoor map. It is necessary to find finer-grained reference information.

3.2 BLE Landmarking

We find that BLE nodes such as iBeacons can be a promising source of reference information for indoor map construction, which can be uniquely identified and have been deployed in many public places. For example, major league baseball (MLB) has been using iBeacons since the start of the 2014 baseball season to track MLB app users and send relevant messages to enhance the ballpark experience [21]; airline companies and retailers have been using iBeacons to send flight information and coupons [22], [23]; the mobile social app WeChat owning hundreds of millions of active users has been providing users with the mini program interface to discover iBeacons [24]; there have been iBeacon based localization systems deployed leveraging the proximity approach [23], [24]. Research efforts also have been dedicated to building BLE based localization systems [12], [32], [33], where it has been verified that BLE has some favored characteristics in localization compared with Wi-Fi; however, how to leverage BLE nodes for indoor map construction is still an open issue.

The fundamental reason that BLE nodes can provide finer-grained reference information is that the nodes can be close to the user's mobile device, which is rooted in the BLE design principles to provide smaller coverage for power saving. Such a favored feature leads to that the relative location of the user's trace with respect to the BLE node can be more accurate and reliable. This provides opportunities to identify spaces between traces and thus can reflect more detailed information of the building's physical layout.

However, to enable the fine-grained self-calibrating localization system, we are confronted with the following new challenges:

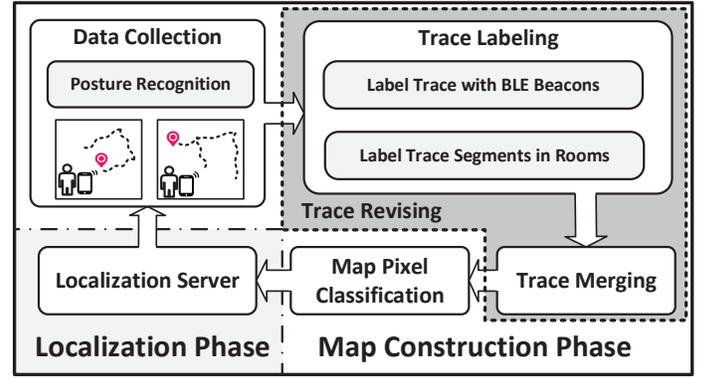


Fig. 3: System Architecture

Challenge 1: Fine-grained map construction. As shown in Table. 1, FineLoc is the only fine-grained self-calibrating indoor localization system. In contrast to previous floor plan construction methods, integrating users' traces with more detailed information instead of just illustrating the skeleton increases the difficulty to realize map construction. Since there are many factors such as coarse-grained RTTP recognition and IMU error affecting the construction performance, it is challenging to construct fine-grained floor plan.

Challenge 2: Dynamic landmarks environment. Since BLE nodes can be in reach of people, it may happen that the BLE nodes are moved accidentally, which results in an environment with dynamic landmarks. As the most important information to label different traces, dynamic landmarks might lead to false floor plan construction. Therefore, it is necessary to design a new trace merging method to recognize landmarks movement and update the map dynamically.

4 SYSTEM OVERVIEW

FineLoc constructs indoor maps in a crowdsourcing manner, where the BLE reference information associated with the crowd worker's traces are used to integrate different traces into a map. There is no need to have prior knowledge of the BLE nodes' locations, and FineLoc tolerates that BLE nodes can be moved or running out of power. We develop mechanisms to categorize traces submitted by different users, from which we derive detailed information reflecting the building's physical layout. FineLoc supports map construction in wide open areas such as the library, where we can recognize the position of obstacles.

FineLoc is shown in Fig. 3, which shows that the system consists of two phases. FineLoc first collects trace data from mobile users and processes the data with several sequential mechanisms to construct the indoor map, which is termed as the *map construction phase*; the constructed map is then used for positioning in the *localization phase*. According to our experiments to be presented in §Section 6, we can construct finer-grained floor plan in contrast to previous works [18]–[20]. Our work in this paper uses BLE beacon based approach in the localization phase, which presents 80% localization errors of 1.6m, 1.4m and 1.1m in the library, classroom and office building respectively. The average density of the BLE nodes is less than 2.6/100m².

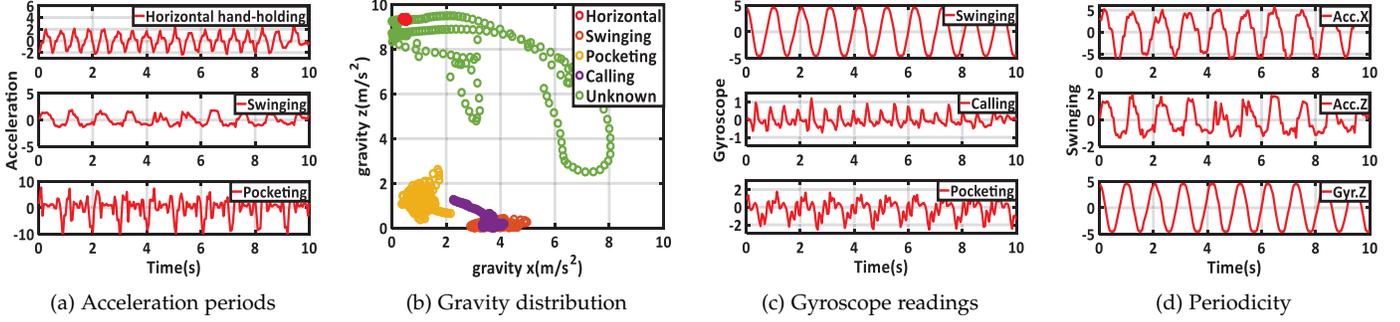


Fig. 4: Posture recognition.

5 SYSTEM DESIGN

5.1 Data Collection

5.1.1 Trace Data Format

FineLoc supports crowdsourcing in a non-participatory manner, where crowd workers can collect data when walking around within the indoor space in daily life. The trace data set is denoted by $T = \{\tau_i, i = 1, 2, \dots, m\}$, where $\tau_i = \langle id, p, f \rangle$ is a specific trace. The element id is factually a tuple $\langle userid, traceid \rangle$ meaning the trace data is collected by which user. Note that a user can submit multiple traces labeled by different $traceids$, where each labels the segment of the user's trace when the user is in a specific posture, and the $traceid$ will change if the user switches the posture during walking. This is because the same IMU trace data indicate different walking distances when the user is in different postures. The element $p = \{p_t, t = 1, 2, \dots, l\}$ records the user's position in each user's relative coordinate system in each sampling time slot with $p_t = \langle x, y \rangle$. We set the initial position and the direction of a newly generated trace as $(0, 0)$ and along x -axis in the relative coordinate system, respectively. In contrast to the previous work [17], [19], [35] assuming the availability of heading direction and stride length of each trace, which is not always true in practice, our trace data are in different relative coordinate systems and will be merged into the same coordinate system in the trace merging process to be presented in §Section 5.3. The element $f = \langle mac, rssi \rangle$ records detected BLE MAC address and RSSI series during the trace, which is to be utilized for clustering and merging traces later.

5.1.2 Posture Recognition in Dead Reckoning

Posture Recognition: The trace information is derived from the IMU data, where the orientation and step count are fundamental for determining the direction and length of the trace. How to recognize change of the walking direction has been well studied in [17], which is also adopted in our work. For the step count determination, detecting acceleration period is a widely adopted approach [14], [17], [19], [28], the basis of which is the observation that the acceleration readings in the three axes of the accelerator present periodicity.

However, such an approach pays limited attention to the practical scenario that users may switch their postures of holding the mobile device during walking. Fig. 4(a) illustrates periods of accelerator readings when the user is using different postures, where the acceleration period

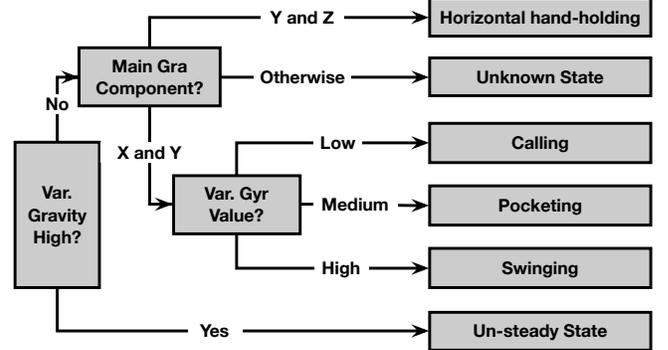


Fig. 5: Posture recognition classifier.

can be identified by observing the accelerator's peak value. However, when the user holds the smartphone horizontally, one period of the accelerator's readings indicates that the user moves one step, but when swinging the smartphone, one period means two steps.

This is because the readings in X , Y and Z axes of the accelerator are dependent on the orientation of the accelerator's coordinate system with respect to the direction of gravity. When holding the smartphone horizontally, each step can result in a slight vibration in the gravity direction, which indicates a period of walking. The gravity direction happens to be in accordance with the direction of Z axis in this case, thus each period of the Z -axis readings represents one step. However, when the user is swinging the smartphone during walking, the gravity direction is in accordance with the direction that is perpendicular to the gravity direction. It can be found that the readings in Z axis in this case show the following pattern: peak (right foot) \sim zero (one step) \sim valley (left foot) \sim zero (two steps) \sim peak (the next period). This period takes twice time compared with the period in the horizontal hand-holding case, so a period represents two steps.

We propose to improve the accuracy of the step-counter by recognizing the user's posture of holding the mobile device. Normally, the user's postures can be categorized into the following classes: horizontal hand-holding, swinging, pocketing and calling, among which the horizontal hand-holding posture is usually used when the user is performing localization. Since the user may have unpredictable postures other than the 4 classes, we categorize the rest of possible postures into the unknown state. Our posture recognition scheme is based on the observation that the gravity accel-

erator readings are different under different postures of the user's, as shown in Fig. 4(b). It can be seen that most of the postures can be well recognized, but some of them have very similar features. For example, mobile device is usually placed vertically when calling, swinging and pocketing. To address this issue, we find that referring to the gyroscope data can help cross check the user's posture, as shown in Fig. 4(c). It is found that the amplitude and pattern of the gyroscope readings vary under different postures. Based on this observation, we have designed a classifier to recognize different postures as shown in Fig. 5. To separate disturbed states (for example, irregular shaking) from regular states, we first calculate the variance of gravity to evaluate the stability of smartphone placement. This is because smartphone placement is relatively steady under regular states. Then we obtain smartphone placement relative to absolute gravity direction of the earth by analysing main gravity component, where Y denotes the vertical placement and Z denotes the horizontal one. Finally, with gyroscope readings, we can further determine smartphone postures.

Step Counter: With the user's posture determined, we find that the readings in X , Y and Z axes of both accelerometer and gyroscope may all present periodicity in Fig. 4(d). Besides, for a certain posture, different readings suffer from noise pollution in varying degrees. We could choose the readings with more obvious periodicity to obtain a more accurate step counts, which is in contrast to existing schemes considering sum of square of readings in all the 3 axes of the accelerometer [14] [17] [28], which could amplify the noise by aggregating the noise in each axis.

Heading Angle and Stride Length: Though many researches have been conducted to improve heading angle recognition and stride length evaluation [18] [19] [34], these two problems remain serious. For heading angle recognition, compass readings would be interrupted by magnetic field and gyroscope can hardly provide absolute direction under different postures; For stride length evaluation, since a certain person might cover varied stride length, a fixed or even trained stride length is still unreliable. In this paper, we propose an adaptive merging scheme (§Section 5.3), which can automatically adjust the orientation and stride length of each trace. Therefore, only relative direction measured by gyroscope readings [17] and a fixed stride length are required in this section for trace generation. Then these two factors would be considered and synchronized.

Remark: In the data collection phase, we want to record the user's trace as long as possible, because it is more likely that a longer trace contains more position information and landmark information. On the one hand, the distance between two parallel traces could be identified if they are long enough to include turns. For example, if the two traces contain movements of taking turnings, the different step counts when the two crowd workers take the turning towards another direction can help determine the distance between the two traces. Our design philosophy is in contrast to the previous work [19] preferring the short traces along one direction, since our work tries to reflect the physical layout of the indoor space as detailed as possible, but the previous work is implicitly designed to construct the pathway layout, where it is convenient for the 1-D short traces in parallel to be integrated into line segments. On the

other hand, a longer trace may go through more landmarks while their relative positions could be determined to reflect the physical layout. Such information could be utilized to recognize dynamic landmarks in §Section 5.3.

5.2 Trace Labeling

5.2.1 Labeling Traces with BLE Beacons

With the collected trace data, we now need to associate each trace with BLE beacon nodes using the beacons' MAC addresses. We associate the trace with a beacon if a part of the trace is within $1m$ of the beacon, where "associate" means that the beacon is considered in the way of the crowd worker and can be regarded as a label of the trace. This is because the BLE node's coverage is very limited, the mobile device can observe strong RSSI in the $1m$ neighborhood of the beacon. We could leverage the labeled traces to derive the skeleton of the floor plan. However, the challenge is that the corresponding RSSI when the mobile device is in the $1m$ neighborhood of the beacon varies due to the random wireless signal propagation and the heterogeneity of beacons and the mobile devices. It is impractical to set a fixed threshold of the RSSI to determine if the mobile device and the beacon are close enough to each other.

To address the issue, we propose a dynamic threshold updating scheme, where the basic idea is to update the maximum RSSI observed by crowd workers, and adjust the information of the traces that are collected before updating accordingly. In particular, consider a BLE landmark i , for which the maximum observed RSSI of the landmark is currently recorded in the server as r_{max}^i . If we have a trace with a series of RSSI $f^i = \{r_1^i, r_2^i, \dots, r_m^i\}$ observed by a crowd worker along the trace $p = \{p_1^i, p_2^i, \dots, p_m^i\}$ with respect to the landmark i , we first define $w(r_k^i, r_{max}^i)$ to represent the weight r_k^i with respect to r_{max}^i :

$$w(r_k^i, r_{max}^i) = \frac{(r_k^i - r_{max}^i - \gamma)^2 I(r_k^i > r_{max}^i - \gamma)}{\gamma^2}, \quad (1)$$

where I is an indicator function, and γ is a protecting threshold to determine the transmission range of the landmark with an empirical value $\gamma = 15$. Then we sum the weight of each value and assign a weight to the trace defined by

$$\mathcal{W}(l^i) = \sum_{k=1}^m w(r_k^i, r_{max}^i), \quad (2)$$

where l^i is the BLE landmark i . The trace has a weight with respect to l^i only if it contains a RSSI observation with respect to l^i such that $r_k^i > r_{max}^i - \gamma$. The greater the RSSI observed in the series, the higher weight the trace is with l^i , thus the closer the trace is to l^i . Note that the square of both the numerator and denominator is to better distinguish weights of two traces. Our empirical study shows that if the weight is greater than 0.8, the trace should be associated with the landmark.

As more crowd workers pass by the landmark l^i , the FineLoc server may get updated by greater r_{max}^i , which means that the weight of the trace with respect to l^i should also be updated. A formal description of the update is

$$\mathcal{W}(l^i | r'_{max}) = \mathcal{W}(l^i | r_{max}) w(r_{max}, r'_{max}), \quad (3)$$

where $\mathcal{W}(l^i | r_{max})$ can be updated to $\mathcal{W}(l^i | r'_{max})$ and $w(r_{max}, r'_{max})$ could be calculated through Eq. 1.

After determining weight of the landmark, we could acquire the landmark's position in the current coordinate system. With acquired weights w_k and positions $p = \{p_1^i, p_2^i, \dots, p_m^i\}$, we can obtain the position $l^i = [x_i, y_i]$ by

$$l^i = \frac{\sum_{k=1}^m p_k w_k}{\mathcal{W}(l^i)}, \quad (4)$$

5.2.2 Labeling Trace Segments in Rooms

A fine-grained indoor map should be able to reflect room information, which requires the trace data to be categorized in terms of sub-areas. The solution in WILL system [15] calculates Euclidean distance between two fingerprints to determine whether the fingerprints are observed from the same room, which is based on the observation that the RSSI changes due to the wall blockage. In contrast to WILL utilizing Wi-Fi signals, FineLoc uses BLE signals, which provide an opportunity to derive more accurate trace segments classification. However, since RSSIs change rapidly when the user is near to the node, tradition Euclidean distance could not be employed directly. Here, we use a discrete function $\mathcal{F}_V(\cdot)$ to map the RSSI into corresponding level. In particular, the RSSI falling into the following ranges $\{>-65, -65\sim-75, -75, -75\sim-85, -85, -85\sim-95, \text{otherwise}\}$ will be mapped into the following levels $\{1, 2, 3, 4, 5, 6, 7\}$. We calculate dissimilarities among the RSSI series along the trace by

$$\mathcal{V}_{k,k+1} = (\mathcal{F}_V(f_k) - \mathcal{F}_V(f_{k+1}))^2, \quad (5)$$

where f_k denotes RSSIs with respect to all observable BLE nodes in position p_k . If $\mathcal{V}_{k,k+1} > 30$, then a new cluster will be generated. In FineLoc, trace segments represent the traces with respect to different rooms. Since there is usually one door between adjacent rooms, we can define the position satisfying $\mathcal{V}_{k,k+1} > 30$ as the *segmented position* for door recognition. Detail methods to recognize the door would be presented in §Section 5.5.

In this way, each step in the trace will be categorized into a cluster, and steps with RSSIs near to each other are put in the same cluster. All clusters reflect the spatial characteristics of the RSSI in the building, which lay the foundation of recognizing rooms. We now could obtain a set $S_l = \{l^1, l^2, \dots, l^m\}$, which includes all landmarks observed along the trace segment. Based on our empirical study, a landmark l^i can be included into subset S_l if one of the following event occurs: First, the crowd worker passes by the landmark and observed RSSI makes the indicator function $I(\cdot) = 1$ (Eqn. 1); second, more than 60% of the RSSI observed along the trace segment satisfies $r_k^i \leq -75$.

Remark: Instead of labeling remote traces (about 5 ~ 10m) based on Wi-Fi RTTP [18] or RSSI correlation [19], [20] to reveal the rough skeletal structure, our system constructs the accurate floor plan with precise information of traces and landmarks. We associate the trace with a beacon if a part of the trace is near to the beacon (about 0 ~ 2m) and calculate the beacon's position. We develop a weighted labeling method based on dynamic threshold and then design corresponding methods to update derived labeling weight. Proposed methods make FineLoc more adaptive to the random wireless signal propagation and the heterogeneity of devices.

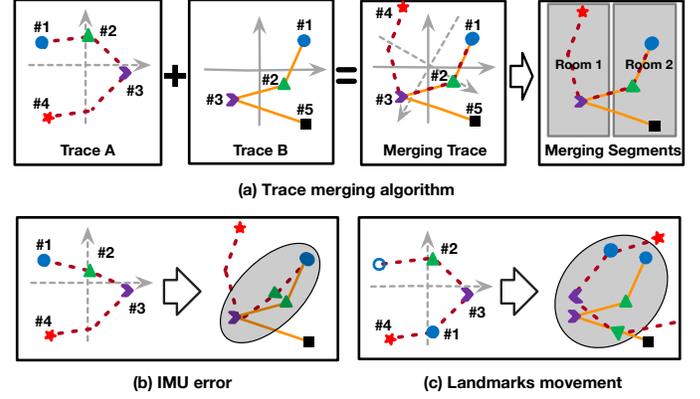


Fig. 6: Process of trace merging.

5.3 Trace Merging

We now integrate the clustered trace segments obtained from the previous processes into a skeleton of the floor plan. This is factually to merge the traces in different relative coordinate systems to a unified coordinate system. Our basic idea is to merge two traces first, and then merge the resulted trace with another trace and so on.

Fig. 6 illustrates an example scenario of the merging process. We need to first translate the entire trace so that their geometric center happens to be the origin of the relative coordinate system, and then rotate the coordinate systems so that the landmarks in the two systems can exactly overlap. Although the BLE landmarks can provide references to combine the two traces, the relative distances among those landmarks in the two systems can be different, thus we need to calculate the misalignment of the landmarks. Such operations can be tedious to realize in practice especially when there are a large number of traces needed to be merged. To address this issue, we present the following matrix transformation based method.

We use $\bar{P}^A = (\bar{x}^A, \bar{y}^A)$ and $\bar{P}^B = (\bar{x}^B, \bar{y}^B)$ to denote the coordinates of the geometric centers of landmarks of trace A and B, respectively, which can be obtained by averaging the horizontal and the vertical coordinates of those landmarks. The two traces have some landmarks in common, and we use \mathcal{L}^A and \mathcal{L}^B to denote the coordinates vectors of those landmarks in the two traces, respectively. Then we have the normalized coordinates vectors of those common landmarks with respect to \bar{P}^A and \bar{P}^B , which are denoted by $\hat{\mathcal{L}}^A$ and $\hat{\mathcal{L}}^B$, respectively. We use K to denote the scaling coefficient for translation, and \mathcal{R} the rotation matrix:

$$\mathcal{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \quad (6)$$

where θ is the angle trace A needs to rotate counterclockwise to comply with trace B. Trace merging is actually finding appropriate K and \mathcal{R} so that

$$\hat{\mathcal{L}}^A K \mathcal{R} = \hat{\mathcal{L}}^B. \quad (7)$$

To this end, we first construct $\mathcal{L}_k^M = [x_k^A + y_k^A, x_k^A - y_k^A]$ and $\mathcal{L}_k^N = [x_k^B + y_k^B, x_k^B - y_k^B]$. Then we can transform equation (7) to the following: $\mathcal{L}^M [K \cos\theta \ K \sin\theta]^T = \mathcal{L}^N$, and obtain that

$$[K \cos\theta \ K \sin\theta]^T = ((\mathcal{L}^M)^T \mathcal{L}^M)^{-1} (\mathcal{L}^M)^T \mathcal{L}^N. \quad (8)$$

With such results, we can merge trace A and B. We use the *residual error* e to denote the misalignment of original landmarks with respect to the resulted ones, where

$$e = \sqrt{|(\hat{\mathcal{L}}^B - \hat{\mathcal{L}}^A K \mathcal{R})^T (\hat{\mathcal{L}}^B - \hat{\mathcal{L}}^A K \mathcal{R})|}. \quad (9)$$

Two traces can be merged if both of the following conditions are satisfied: First, they must have at least 3 landmarks in common; Second, the resulted error $e < T_e$, where T_e is set to be 8 steps. This is because if the two traces only have 1 or 2 landmarks in common, they definitely can be perfectly merged no matter how they deviate from each other.

In FineLoc, e might result from the malfunctioning step counter or displaced landmarks. In reality, since step counter error is slowly cumulative and evenly distributed, it usually leads to a smaller e compared with displaced landmarks as shown in Fig. 6(b). In contrast, once a landmark has been moved to another position as shown in Fig. 6(c), we can hardly merge the traces collected, because landmark movement leads to a large residual error e . Therefore, by observing the abnormal residual errors larger than T_e in the trace merging process, FineLoc is able to detect unreliable landmarks, which are to be updated. This will ensure the robustness of the system in dynamic environment.

We have merged the two traces as of now, the next step is to cluster the segments of the traces, so that the trace segments in different rooms can be recognized. Recall that each segment of a trace is associated with a set of landmarks S_l . We define the similarity of S_l and S'_l as

$$\mathcal{F}_s(S_l, S'_l) = \frac{S_l \cap S'_l}{S_l \cup S'_l}. \quad (10)$$

If $\mathcal{F}_s(S_l, S'_l) > 0.6$, the two segments of traces can be categorized into the same group in our system according to our empirical study.

Remark: In contrast to previous works slicing the trace into segments such as turns or straight lines and then merging the similar segments [18]–[20], we merge complete traces directly. The advantage of our method is that we can realize the overall layout of beacons as shown in Fig. 6. In this way, we present a method to recognize the movement of beacons and adapt FineLoc to dynamic landmarks environment. Note that “segment” in this paper represents the traces relative to different rooms, in contrast to turns or straight lines in previous works [19] [20].

5.4 Trace Revising

In trace labeling and trace merging stages, the system performance highly relies on the accuracy of IMU sensors. Though posture recognition algorithm has been proposed and step counter with 98% accuracy has been realized, floor plan construction also suffers from direction error because the error in direction will be gradually accumulated as the user moves on. This problem might lead to serious error, which however has not been resolved by previous floor plan construction systems [18] [19] [31] yet. In this section, we propose a novel scheme to revise user’s direction where there are two stages: trace self-revising and trace co-revising.

In the trace self-revising stage as shown in Fig. 7(a), there are three BLE landmarks A, B, C and three turning points TP_1, TP_2, TP_3 . When a user goes through a closed-loop trace $A - TP_1 - B - TP_2 - C - TP_3 - A$, we might obtain a measured trace as shown in fig 7(b) due to IMU errors. We here define cumulative bias $\mathcal{D}_{AA'}$ as an indicator of trace accuracy. Intuitively, the system should modify the values of TP_1, TP_2, TP_3 to minimize $\mathcal{D}_{AA'}$ which can be formulated into the following optimization problem:

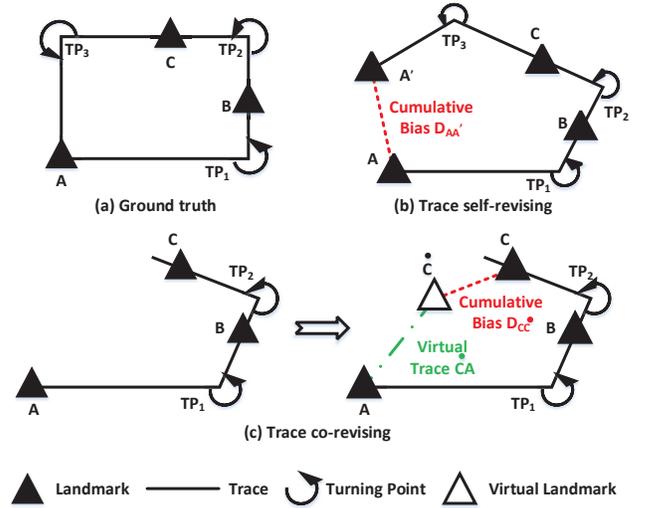


Fig. 7: Process of trace revising.

$E_{TP} = \arg \min_{E_{TP}} \mathcal{D}_{AA'}$, where $E_{TP} = \{TP_1, TP_2, \dots, TP_n\}$ denotes the sequence of turning points. To address this issue, our system utilizes particle filter to select TP_i to generate new particles and finally determine E_{TP} .

In practice, we may not be able to obtain a closed-loop path all the time. To extend our scheme to non-closed-loop scenarios, we combine trace merging stage and trace revising stage, which yields a trace co-revising algorithm. The basic idea is to construct virtual closed-loop path with merging landmarks. As shown in Fig. 7(c), when the user generates a non-closed-loop trace $A - TP_1 - B - TP_2 - C$, self-revising scheme can not work; however, if this non-closed-loop trace is merged into a global trace, we can perform co-revising scheme as follows: 1) Determine rotation matrix \mathcal{R} and scaling coefficient K (Equation 8), and place landmarks A, C, \hat{A}, \hat{C} into the same coordinate system, where \hat{A} and \hat{C} mean landmarks in global trace with a higher reliability; 2) Move \hat{A} to be overlapped with A , and we obtain a virtual position of landmark \hat{C} ; 3) Connect A (or \hat{A}) and \hat{C} to generate virtual trace $\hat{C}A$; 4) Based on the closed-loop path $\hat{C} - A - TP_1 - B - TP_2 - C$, we can define cumulative bias $\mathcal{D}_{C\hat{C}}$ and the problem is simplified into a self-revising problem.

Remark: For the purpose of IMU error elimination, previous works [19] [20] propose inter- and intra- trajectory correcting schemes, which correct traces by merging multiple similar traces into one trace, thus can not recognize obstacles as illustrated in Fig. 2. To address this issue, our data collection and trace merging stages prefer merging complete traces directly instead of slicing them into smaller segments (turns and straight lines). However, merging complete traces suffers more from direction error because the error in direction will be gradually accumulated as the user moves on. Since previous trace revising method is course-grained, it is necessary to propose our trace revising method to eliminate the IMU error.

5.5 Map Pixel Classification

The skeleton of the indoor floor plan can be obtained with the process as shown in Fig. 8. We can divide the map into small square sub-regions, each of which can be

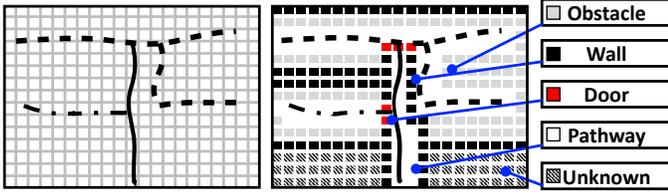


Fig. 8: Map pixel clustering.

regarded as a pixel. Besides pixels covered by those trace segments, there are still a large number of pixels that need to be classified as walls, doors, pathways and so on. We here present our map pixel classification method.

Suppose we have a map skeleton such as the one shown in Fig. 8, for each pixel, we first calculate the distance between the pixel and each kind of trace segment. Give a trace segment of cluster ID i , the distance d_i is the sum of the distances between the pixel and the k nearest points in the trace. Suppose that there are m clusters of trace segments, then we have $\{d_i, i = 1, 2, \dots, m\}$, among which d_{1min} and d_{2min} denote the shortest and the second shortest distances respectively. We define $d_{\Delta} = d_{2min} - d_{1min}$. Then we can classify the pixel according to the following rules:

$$Pixel\ role = \begin{cases} Pathway, & d_{1min} < 0.5k; \\ Obstacle, & 0.5k < d_{1min} < 2k; \\ Unknown, & d_{1min} > 2k; \\ Wall, & d_{\Delta} < 0.5k \ \& \ d_{2min} < 2k. \end{cases}$$

Note that the distance between the pixel and a single point in the trace segment can be erroneous due to the unpredictable abrupt malfunction of the step-counter or the sudden change of BLE RSSEs, that is why we use the k nearest points in the trace to determine the shortest distance, where k is set to be 4 in our system; moreover, since the absolute distance between two pixels is indistinguishable in a fine-grained map, k also can amplify the distance differences, so that the pixels can be classified more accurately.

The rationales of the rules presented above are as follows. The pixel that is $0.5m$ from a trace is likely to be a pathway pixel, since the width of the space a walking human body needs to occupy is around $0.5m$. The pixel satisfies the second rule can be an obstacle, since normally the dimension of an obstacle in the indoor space is around $2m$ such as desks and bookshelves. This rule can not 100% guarantee identifying the obstacles, since there indeed are some small obstacles such as sofa and chairs; however, if we have sufficient trace data, such small obstacles still can be recognized. If the pixel is comparatively far away ($2m$) from the nearest trace, we temporarily regard it as an unknown point; if no other traces are near to it, probably nobody ever passes by, thus it is indeed unknown. The wall is probably adjacent to or separate two traces, and the wall is actually one kind of obstacle thus we let $d_{2min} < 2k$. Moreover, in §Section 5.2.2, we define segmented position when we observe a new cluster. Since the segmented position represents the border between different rooms, we regard one pixel as the door when the distance between the pixel and the segmented position is lower than $0.5k$.

Remark: Compared with previous works which can only sketch the skeleton of floor plan as shown in the left subfigure of Fig. 8, we for the first time develop the map pixel classification method to realize fine-grained map construction. With proposed methods, we can construct floor plan with limited traces, where subrooms and obstacles could be recognized.

5.6 Map Construction and Localization

After introducing the frame and methodology of Fine-Loc, we here show how to integrate our proposed schemes (§Section 5.1 ~ §Section 5.5) into algorithms to perform map construction and localization.

Trace Generation Algorithm: In Algorithm. 1, we would collect data, recognize smartphone postures and generate user trace. The system generates a new trace when the user's posture switching occurs since posture switching would introduce error in dead reckoning and damage floor plan construction.

Algorithm 1 Trace Generation

Input: Sensor readings

Output: Labeled trace

for each time slot k **do**

Parse acc, gra, gyr from sensor readings;

§Section 5.1.2 - Posture recognition

Recognize smartphone posture $Post_t$;

if $Pos_k \neq Pos_{k-1}$ **then**

Generate a new $traceid$;

else

Update position p_k ;

§Section 5.2.1 - Trace Labeling

Parse RSSI series $f^i = \{r_1^i, r_2^i, \dots, r_m^i\}$;

if $I(\cdot) = 1$ (Eq. 1) **then**

if $r_k^i \leq r_{max}^i$ **then**

Calculate $w(r_k^i, r_{max}^i)$ and $\mathcal{W}(l^i)$ (Eq. 2);

else

Update $r'_{max} = r_k^i$ and $\mathcal{W}(l^i | r'_{max})$ (Eq. 3);

end if

Attach landmark l^i to trace (Eq. 4);

§Section 5.4 - Trace self-revising

if l^i is historical landmark of this trace **then**

Revise trace and obtain E_{TP} ;

end if

end if

§Section 5.2.2 - Label trace segments

Calculate dissimilarities $\mathcal{V}_{k,k+1}$ (Eq. 5);

if $\mathcal{V}_{k,k+1} > 30$ **then**

Generate new cluster and obtain segment set S_i ;

end if

end if

end for

Floor Plan Generation Algorithm: The purpose of Algorithm. 2 is to merge, revise trace and finally finish map pixel classification. Here we define collected trace space E_T^{Local} and merged trace space E_T^{Global} , where E_T^{Local} denotes the traces collected by users and E_T^{Global} represents the traces after merging. With more traces collected and merged, we can enlarge the scale and reliability of E_T^{Global} . Finally,

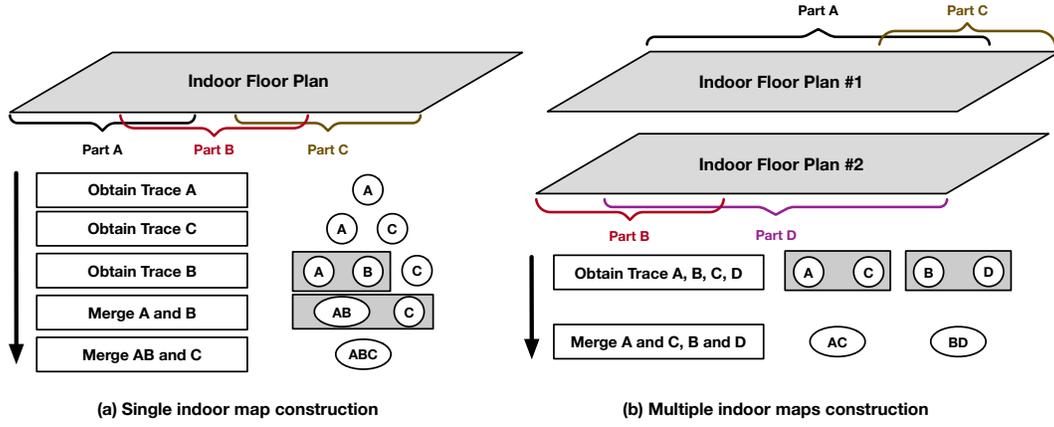


Fig. 9: Process of floor plan generation.

detailed floor plan would be constructed through map pixel classification scheme.

We use Fig.9 to illustrate how the floor generation algorithm works. As shown in Fig.9(a), we collect trace A, C, B sequentially. Since trace A and C have no intersection, we first consider that they belong to different indoor maps. Once trace B is collected, we can merge trace A and B via trace merging module and derive trace AB. Meanwhile, since there exists intersection between trace AB and C, we finally merge all traces and construct more completed floor plan. Proposed methods can be directly employed in multi-floor or multi-building scenarios without any modification. As shown in Fig.9(b), there are four traces A, B, C, D coming from different buildings. In our system, we derive BLE's MAC address to uniquely identify different landmarks. Differing from Wi-Fi which covers a hundred-meter-radius area, BLE can only cover limited area, which means no interface among multiple floors or multiple buildings. Since there can never be landmark intersection between traces coming from different indoor maps, we thus independently construct respective indoor map (AC and BD).

Localization Phase: In the mapping phase, our system utilizes the crowdsourced data to construct finer-grained floor plan. Meanwhile, FineLoc attempts to localize the user with collected data. In previous work, dead reckoning [17] and fingerprinting based [14] [18] [19] localization schemes have been employed for localization. With the improvement of IMU accuracy, dead reckoning shows higher reliability than fingerprinting within a small region. To further eliminate cumulative error of dead reckoning, landmarks could be utilized.

In this paper, localization is based on dead reckoning and BLE landmarks. Compared with traditional landmarks (for example, water dispenser and turns [31]), BLE landmarks show higher reliability and uniqueness. Specifically, the system first generates user's trace and then determines which map the trace belongs to. Due to uniqueness and limited coverage of BLE nodes, it is simple to find corresponding map. It should be noted that when new landmarks are detected, we would perform mapping phase to construct a new region map. Secondly, to localize the user, we just need to merge his/her trace into existing floor plan, where rotation matrix \mathcal{R} and scaling coefficient K are required. Therefore, according to the merging rule in Equation. 7,

Algorithm 2 Floor Plan Generation

Input: Collected trace space E_T^{Local} , merged trace space E_T^{Global}

Output: Finer-grained floor plan

```

for all trace  $T_A, T_B \in E_T^{Local} \cup E_T^{Global}$  do
  Calculate overlapping landmarks  $L^A$  and  $L^B$ 
  if  $size(L^A)$  and  $size(L^B) \geq 3$  then
    §Section 5.3 - Trace merging
    Determine  $\mathcal{R}, K$  and  $e$  (Eq. 8,9);
    if  $e > T_e$  then
      Determine dynamics of landmarks;
    else
      §Section 5.4 - Trace co-revising
      if  $T_A$  or  $T_B \in E_T^{Global}$  then
        Perform trace co-revising
      end if
      Merge trace  $T_A, T_B$  and obtain global trace  $T_G$ ;
      Add global trace  $T_G$  into trace space  $E_T^{Global}$ 
      Merge the same segment; (Eq. 10);
      §Section 5.5 - Map pixel classification
      Perform map pixel classification for trace  $T_G$ ;
    end if
  end if
end for

```

we can obtain $P = (P^A - \bar{P}^A)K\mathcal{R} + \bar{P}^B$, where P^A denotes the position in current trace's coordinate system, P represents the position in global floor plan, $\bar{P}^A = (\bar{x}^A, \bar{y}^A)$ and $\bar{P}^B = (\bar{x}^B, \bar{y}^B)$ denote the coordinates of the geometric centers of landmarks of trace A and B.

6 PERFORMANCE EVALUATION

6.1 Experimental Setups

We implement FineLoc on different Android phones (HUAWEI Mate 7, InFocus M512, Nexus 5) and conduct experiments in three buildings to evaluate our proposed mechanisms, where the first testing environment is a 10000 m^2 pathway area in a classroom building, the second is a 500 m^2 area in an office building with rooms, and the third is a 1200 m^2 wide area, where the floor plans are shown in Fig 10.

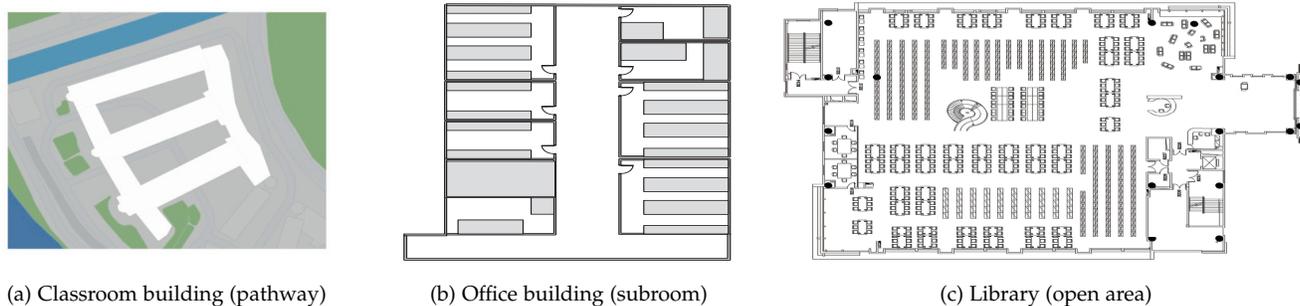


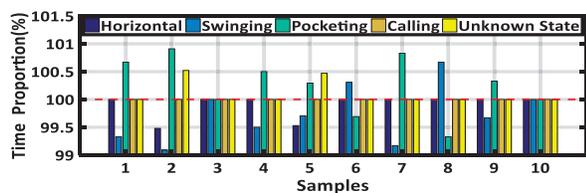
Fig. 10: The ground truth of three scenarios

TABLE 2: Accuracy of Floor Plan

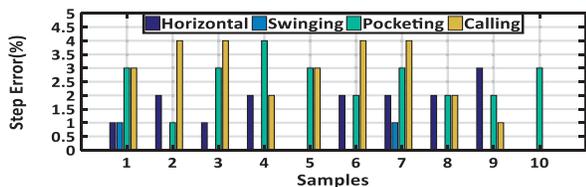
Scenario	Mean error	80% error	\mathcal{P}_{room}	\mathcal{R}_{room}	\mathcal{F}_{room}	$\mathcal{P}_{obstacle}$	$\mathcal{R}_{obstacle}$	$\mathcal{F}_{obstacle}$
Classroom (pathway)	0.97m	1.32m	N/A	N/A	N/A	N/A	N/A	N/A
Office (subroom)	0.81m	0.91m	81.3%	83.8%	82.5%	68.7%	64.8%	66.7%
Library (open area)	1.21m	1.57m	N/A	N/A	N/A	61.6%	53.0%	57.1%

6.2 Accuracy of Posture Recognition

We enroll 5 volunteers to test the accuracy of the proposed posture recognition scheme. Each volunteer is asked to walk 400 steps twice, during which freely switching postures is encouraged. We record the time consumed by each walk, the duration and the number of steps for each posture. Fig. 11(a) shows the experimental results in time, where the horizontal axis is the index of the experiment and the vertical axis is the proportion of time each posture is correctly recognized; we can see that the scheme can correctly identify different postures in more than 99% of the time, where most of the can-not-tell situations occur during the posture switching. Fig. 11(b) shows the results in the number of steps, which corroborates the results in Fig. 11(a). We can see that only situations with less than 4 steps may be incorrectly identified for each posture.



(a) Accuracy in time



(b) Accuracy in num. of steps

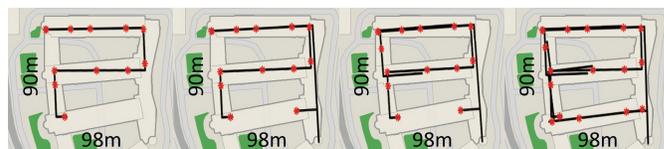
Fig. 11: Accuracy of posture recognition

6.3 Performance of Map Construction

We construct the indoor floor plans of the three testing areas, and randomly select a number of points, including 420 points from the classroom building, 154 points from the office building and 249 points from the library. Those points are then compared with the corresponding points in the real map to find the deviation. There are 18 and 68

obstacles in the office building and the library respectively; the pathways in the classroom building contain no obstacles. We evaluate the proposed map construction mechanism in terms of the 8 metrics as tabulated in TABLE 2. Note that the first 5 metrics are also used in SensorWit [31] and CrowdMap [36], where $\mathcal{P}_{room} = \frac{|S_{generate} \cap S_{true}|}{|S_{true}|}$, $\mathcal{R}_{room} = \frac{|S_{generate} \cap S_{true}|}{|S_{generate}|}$, $\mathcal{F}_{room} = 2 \times \frac{\mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}}$.

We can see that \mathcal{P}_{room} denotes the ratio of correct sub-room area to generated area while \mathcal{R}_{room} means the ratio of correct sub-room area to ground truth. \mathcal{F}_{room} combines \mathcal{P}_{room} and \mathcal{R}_{room} . Since there is no sub-room in classroom and library, the second and fourth rows employ N/A in TABLE 2. Meanwhile, since our proposed mechanism is also able to recognize obstacles, we utilize $\mathcal{P}_{obstacle}$, $\mathcal{R}_{obstacle}$ and $\mathcal{F}_{obstacle}$ to verify the performance of obstacle construction. In particular, the definition of $\mathcal{P}_{obstacle}$, $\mathcal{R}_{obstacle}$ and $\mathcal{F}_{obstacle}$ is also similar to that of \mathcal{P}_{room} , \mathcal{R}_{room} and \mathcal{F}_{room} .



(a) One trace (b) Two traces (c) Three traces (d) Four traces

Fig. 12: Floor plan of classroom building.

Fig. 12 shows the layout of the classroom building and the constructed pathway map. We deploy 16 landmarks along the 500m-long pathway. We only utilize 4 traces to construct the map, which takes less than 20mins. This outperforms the Walkie-Markie and PiLoc, where the performance are 260m within 50mins and 150m within 10mins respectively. This is because we record long trace segments instead of dividing the trace into very small segments in previous systems [18]–[20]. This will save the amount of data required and improve the data processing efficiency.

Fig. 13 shows the layout of testing area in the office building, where there are 8 rooms and 13 landmarks denoted by red stars. In contrast to existing scheme that can only profile pathways, the FineLoc can sketch the outlines of rooms, obstacles, doors and pathways, which are in the color

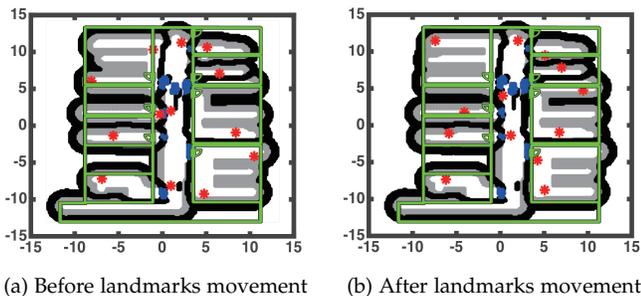


Fig. 13: Floor plan of the office building area.

of black, grey, blue and white in the figure, respectively. The map shown in Fig. 13 (a) is constructed after 20mins of volunteers' free activities, and that shown in Fig. 13 (b) is constructed while 5 out of the 13 landmarks are moved to somewhere else, which shows the capability of automatic landmark updating.

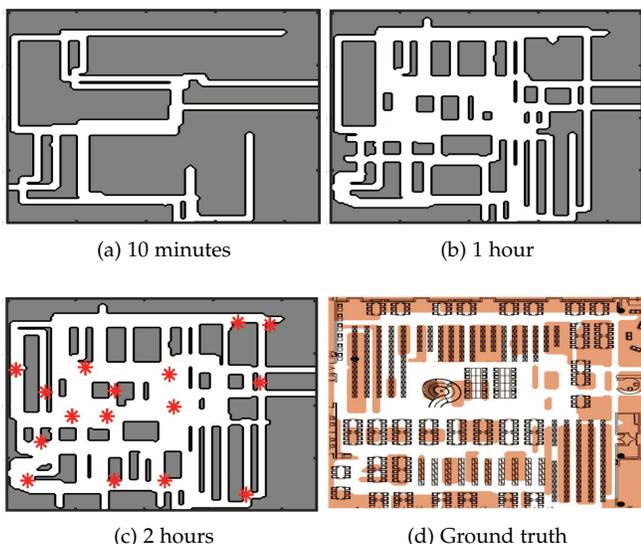


Fig. 14: Floor plan construction of university library

Fig. 14 shows the layout of the testing area in the library, where there are 68 obstacles and 16 landmarks denoted by red stars. Compared with existing scheme that can only sketch the outline of the whole building, the FineLoc can sketch obstacles in details, which is in the color of grey. The map shown in Fig. 14(a) is constructed after 10 mins of volunteers' free activities, and those shown in Fig. 14(b)(c) are constructed after 1 hour and 2 hours of volunteers' free activities. Then we have placed the floor plan into a ground truth and show the performance with a 16.2% obstacles missing and 20.6% extra obstacles in Fig. 14 (d).

6.4 System Comparison

In this section, we compare the FineLoc system with existing methods to verify that FineLoc can realize finer-grained indoor map construction. During experiments, we collect Wi-Fi/BLE signals and construct floor plan based on PiLoc and FineLoc. In particular, existing Wi-Fi floor plan construction methods can only sketch the skeleton of floor plan as shown in Fig. 15(a). Meanwhile, there is also false trace merging due to failed correlation detection. In contrast,

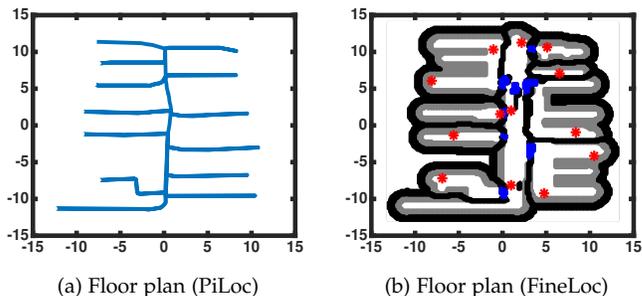


Fig. 15: Map construction comparison

FineLoc can recognize different subrooms, obstacles, walls and doors. Thus we can realize finer-grained floor plan construction as shown in Fig. 15(b).

To verify the performance of subroom recognition and obstacles recognition, we then compare our system with SensorWit. As shown in TABLE. 1, one important advantage of FineLoc is unique calibration. It means that FineLoc can simultaneously construct multiple indoor maps without interrupting each other. In addition, FineLoc can realize a higher accuracy as shown in TABLE. 3.

TABLE 3: System Comparison

System	FineLoc	PiLoc	SensorWit
\mathcal{F}_{room}	82.5%	Not work	76.0%
$\mathcal{F}_{obstacle}$	57.1% ~ 66.7%	Not work	Not work

6.5 Localization Performance

There are 45 landmarks employed in our system. To measure the performance of landmark's position, we show the results in Fig. 16(a), where 80% errors are less than 1.0m, 0.5m and 1.2m respectively. Meanwhile, We collect 84 traces from the three buildings, and select 2459 points in buildings to perform location estimation, where there are 475 points from the classroom building, 726 points from the office building and 1258 points from the library. We utilize BLE beacon nodes and the smartphone to do localization experiments. The resulted CDFs are illustrated in Fig. 16(b), where 80% errors are less than 1.4m, 1.1m and 1.6m respectively. We can see that our system performs stably in different indoor environments.

Fig. 16(c) shows the localization results of dynamic environment, where BLE landmarks can be moved to other places. The experiments are conducted in the office building. We move 5 out of 13 BLE nodes to other randomly selected places and then perform the map updating and location estimation simultaneously. We can see from Fig. 16(c) that the localization error increases dramatically when BLE nodes are moved, because the smartphone may consider the BLE's current position as the position in the original constructed map. However, with our map updating scheme, the new map accommodating the current locations of the BLE nodes can be generated after around 20mins of the volunteers' free activities.

7 DISCUSSIONS

Device heterogeneity: When conducting experiments, we can observe notable device heterogeneity. This might

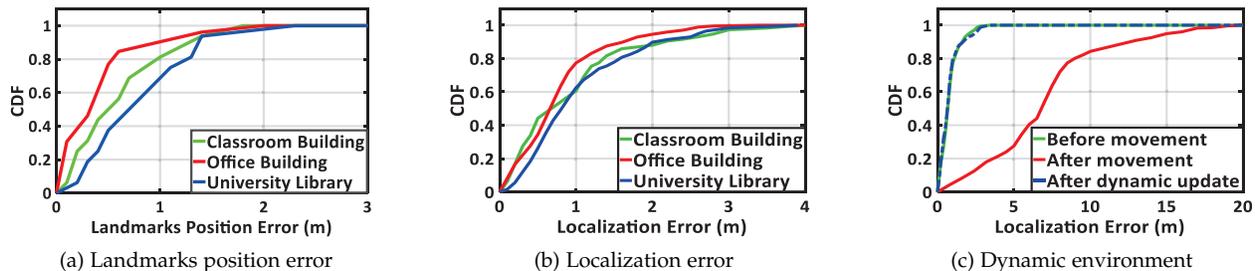


Fig. 16: Localization performance.

result from several reasons such as hardware implementations and placement of beacons. To measure 45 BLE nodes employed in our system (16 in classroom building, 13 in office building and 16 in university library), we plot historical maximum RSSIs in Fig. 17(a). This indicates that it is necessary to perform our trace labeling algorithm instead of a fixed RSSI threshold method.

Deployment of BLE Beacons: As shown in Fig 12, 13 and 14, the densities of BLE nodes are $0.18/100m^2$, $2.6/100m^2$ and $1.3/100m^2$ respectively. This means that the proposed mechanism can work with sparsely populated BLE nodes. In real situation, the sub-room structure such as the office building area shown in Fig. 10(b) requires the most iBeacons to construct the floor plan. It is because that at least one iBeacon is necessary to label each room (§Section 5.2.2). Even so, FineLoc can construct fine-grained floor plan shown in Fig. 13 with limited BLE nodes. Moreover, due to proposed trace labeling and merging mechanism, FineLoc prefers the BLE Beacons that are easier to be in reach of people. It is because that the trace labeled with more landmarks can be merged and revised with less error.

System cost: During our experiments, the cost of each iBeacon is about \$6 [37], which leads to $\$0.01/m^2$, $\$0.16/m^2$ and $\$0.08/m^2$ cost under the three scenarios shown in Fig. 10. We can thus construct fine-grained floor plan with the cost less than $\$0.16/m^2$. Such cost can be easily accepted for localization applications.

Scanner of BLE RSSI: Compared with Wi-Fi scanner, the activation rate of general BLE scanning on smartphone might be low. Through our experiment we find that BLE scanner on smartphone just works in Android 4.3 or higher version and obtains RSSI each 3-5 seconds sometimes. Fortunately, current Android version is normally newer than Android 4.3 and we can improve the scanning rate to 1 time/second. This can be realized by turning on and off BLE adapter periodically using `adapter.stopLeScan()` and `adapter.startLeScan()`. This would increase scanning rate without incurring much energy consumption (Fig. 17(b)).

Energy consumption: The energy consumption of BLE nodes is low; however, scanning BLE beacon signals will consume energy in users' mobile devices. As we mentioned, we could improve the scanning frequency of the mobile device to detect the BLE beacon signal, but will this incur higher energy consumption in the mobile device? We find the answer to the issue through experiments, for which we set the scanning interval to be 1s for 3 kinds of Android phones (HUAWEI Mate 7, InFocus M512, Nexus 5). We maintain the scanning process for 4 hours and show the energy consumption results in Fig. 17(b). We can observe that the resulted energy consumption by improving the scanning

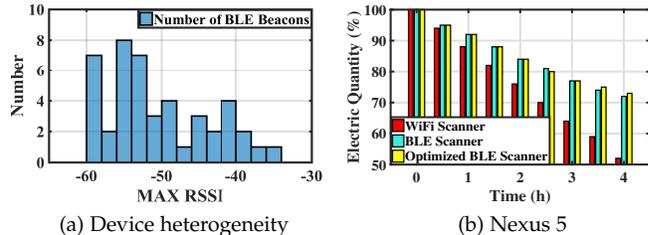


Fig. 17: Device heterogeneity and energy consumption.

frequency is slightly higher than the default configuration, but still lower than half of that for scanning Wi-Fi. For the other two kinds of Android phones, we can observe similar results during the experiments.

8 CONCLUSIONS

In this paper, we propose FineLoc, a finer-grained self-calibrating indoor localization system based on RSSI of wireless signals. Existing wireless indoor localization system can only sketch coarse-grained trace floor plan. However, FineLoc can adapt itself to various environments and sketch room outline, doors and obstacles. Through experiments, we show a finer-grained floor plan with 80% error less than $2.5m$ and 80% localization error less than $1.6m$. Besides, FineLoc works normally under heterogeneous equipment and dynamic environment. It can update landmarks' position and the floor plan automatically.

9 ACKNOWLEDGEMENT

The work in this paper is supported by the National Key Research and Development Program of China 2017YFB1003000, and National Natural Science Foundation of China (No. 61822206, 61829201, 61872233, 61602303, 61572319, 61532012, 61428205).

REFERENCES

- [1] Z. Yang, Z. Zhou and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, pp.1-32, 2013.
- [2] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, 2000, pp. 775-784.
- [3] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proc. ACM MobiSys*, 2005, pp. 205-218.
- [4] K. Chintalapudi, A. Padmanabha Iyer and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. ACM MobiCom*, 2010, pp. 173-184.
- [5] S. Yang, P. Dessai, M. Verma and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in *Proc. IEEE INFOCOM*, 2013, pp. 2481-2489.
- [6] S. He and S. H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," in *IEEE Communications Surveys & Tutorials*, 2016, pp. 466-490.

- [7] D. Halperin, W. Hu, A. Sheth and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," in *ACM SIGCOMM*, 2011.
- [8] M. Kotaru, K. Joshi, D. Bharadia and S. Katti, "SpotFi: Decimeter Level Localization Using WiFi," in *Proc. ACM SIGCOMM*, 2015, pp. 269-282.
- [9] S. Kumar, S. Kumar S and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. USENIX NSDI*, 2016, pp. 165-178.
- [10] L. Yang, Y. Chen, X. Li, C. Xiao, M. Li and Y. Liu, "Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using COTS Devices," in *Proc. ACM MobiCom*, 2014.
- [11] K. Liu, X. Liu and X. Li, "Guoguo: Enabling Fine-Grained Smartphone Localization via Acoustic Anchors," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp.1144-1156, May 2016.
- [12] R. Faragher and R. Harle, "Location fingerprinting with Bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418-2428, Nov. 2015.
- [13] B. Zhou, M. Elbadry, R. Gao and F. Ye, "BatMapper: Acoustic Sensing Based Indoor Floor Plan Construction Using Smartphones," in *Proc. ACM MobiSys*, 2017, pp. 42-55.
- [14] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proc. ACM MobiCom*, 2012, pp. 293-304.
- [15] C. Wu, Z. Yang, Y. Liu and W. Xi, "Will: wireless indoor localization without site survey," in *IEEE Transactions on Parallel & Distributed Systems*, 2013, pp. 839-848.
- [16] Z. Yang, C. Wu and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proc. ACM MobiCom*, 2012, pp. 269-280.
- [17] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef and R. R. Choudhury, "No need to war-drive:unsupervised indoor localization," in *Proc. ACM MobiSys*, 2012, pp. 197-210.
- [18] G. Shen, Z. Chen, P. Zhang, T. Moscibroda and Y. Zhang, "Walkie-Markie: indoor pathway mapping made easy," in *Proc. USENIX NSDI*, 2013, pp. 85-98.
- [19] C. Luo, H. Hong and M. C. Chan, "PiLoc: A self-calibrating participatory indoor localization system," in *International Symposium on Information Processing in Sensor Networks*, 2014, pp. 143-154.
- [20] C. Luo, H. Hong, M. C. Chan, J. Li, X. Zhang and Z. Ming, "MPi-Loc: Self-calibrating multi-floor indoor localization exploiting participatory sensing," in *IEEE Transactions on Mobile Computing*, 2018, pp. 141-154.
- [21] C. Birkner "MLB's iBeacon Technology is A Marketing Home Run," <https://www.ama.org/publications/MarketingNews/Pages/mlb-ibeacons.aspx>.
- [22] T. Reddy, "15 Companies From Airports to Retail Already Using Beacon Technology," <https://www.umbel.com/blog/mobile/15-companies-using-beacon-technology/>.
- [23] iBeacon.io, <http://ibeacon.com.hk/>
- [24] WeChat mini program, <https://mp.weixin.qq.com/debug/wxadoc/dev/api/iBeacon.html#wxstopbeacondiscoverobject>.
- [25] X. Wu, R. Shen, X. Tian, P. Liu and X. Wang, "iBILL: Using iBeacon and Inertial Sensors for Accurate Indoor Localization in Large Open Areas," in *IEEE Access*, 2017, pp. 14589-14599.
- [26] B. Ferris, D. Fox and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *International Joint Conference on Artificial Intelligence*, 2007, pp. 2480-2485.
- [27] KAILOS, <https://kailos.io/>.
- [28] Y. Shu, K. G. Shin, T. He and J. Chen, "Last-Mile Navigation Using Smartphones," in *Proc. ACM MobiCom*, 2015, pp. 512-524.
- [29] C. Zhang, and X. Zhang, "LiTell: Robust indoor localization using unmodified light fixtures," in *Proc. ACM MobiCom*, 2016, pp. 230-242.
- [30] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. ACM MobiCom*, 2014, pp. 249-260.
- [31] Y. He, J. Liang J and Y. Liu, "Pervasive Floorplan Generation Based on Only Inertial Sensing: Feasibility, Design, and Implementation," in *IEEE Journal on Selected Areas in Communications*, 2017, pp. 1132-1140.
- [32] N. Nakajima, S. Tang, T. Ogishi and S. Obana, "Improving Precision of BLE-based Indoor Positioning by Using Multiple Wearable Devices," in *Ajunct Proc. ACM MOBIQUITUS* 2016, pp. 118-123.
- [33] G. G. Anagnostopoulos and M. Deriaz, "Accuracy Enhancements in Indoor Localization with the Weighted Average Technique," in *Proc. SENSORCOMM* 2014.
- [34] D. K. Cho, M. Mun, U. Lee, W. J. Kaiser, and M. Gerla, "Autogait: A mobile platform that accurately estimates the distance walked," in *PerCom* 2010, pp. 116-124.
- [35] F. Li, C. Zhao, G. Ding, C. Liu and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM Ubicomp*, 2012, pp. 421-430.
- [36] S. Chen, M. Li, K. Ren and C. Qiao, "Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos," in *IEEE ICDCS*, 2015, pp. 1-10.
- [37] Minewtech E5, <https://en.minewtech.com/ibeacon/E5.html>.



Xinyu Tong received the B.E degree in Department of Electronic Information and Electrical Engineering from Shanghai Jiao Tong University, Shanghai, China in 2015. He is currently a Ph.D student in Shanghai Jiao Tong University. His research interests include wireless sensor network and wireless localization.



Ke Liu received the B.E degree in Department of Electronic Information and Electrical Engineering from Shanghai Jiao Tong University, Shanghai, China in 2018. She is going to pursuing the M.S degree in University of California, San Diego. Her research interest is crowdsourcing based indoor localization.



Xiaohua Tian received his B.E. and M.E. degrees in communication engineering from Northwestern Polytechnical University, Xian, China, in 2003 and 2006, respectively. He received the Ph.D. degree in the Department of Electrical and Computer Engineering (ECE), Illinois Institute of Technology (IIT), Chicago, in Dec. 2010. Since Mar. 2011, he has been with the School of Electronic Information and Electrical Engineering in Shanghai Jiao Tong University, and now is an Associate Professor with the title of SMC-

B scholar. He serves as the column editor of *IEEE Network Magazine* and the guest editor of *International Journal of Sensor Networks* (2012). He also serves as the TPC member for *IEEE INFOCOM* 2014-2017, best demo/poster award committee member of *IEEE INFOCOM* 2014, TPC co-chair for *IEEE ICC* 2014-2016, TPC Co-chair for the 9th International Conference on Wireless Algorithms, Systems and Applications (WASA 2014), TPC member for *IEEE GLOBECOM* 2011-2016, TPC member for *IEEE ICC* 2013-2016, respectively.



Luoyi Fu received her B. E. degree in Electronic Engineering from Shanghai Jiao Tong University, China, in 2009 and Ph.D. degree in Computer Science and Engineering in the same university in 2015. She is currently an Assistant Professor in Department of Computer Science and Engineering in Shanghai Jiao Tong University. Her research of interests are in the area of social networking and big data, scaling laws analysis in wireless networks, connectivity analysis and random graphs.



Xinbing Wang received the B.S. degree (with honors) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001. He received the Ph.D. degree, major in the Department of electrical and Computer Engineering, minor in the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is a professor in the Department of

Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. Dr. Wang has been an associate editor for *IEEE/ACM Transactions on Networking* and *IEEE Transactions on Mobile Computing*, and the member of the Technical Program Committees of several conferences including *ACM MobiCom* 2012, *ACM MobiHoc* 2012-2014, *IEEE INFOCOM* 2009-2017.