Research Article

Location Fixing and Fingerprint Matching Fingerprint Map Construction for Indoor Localization

Marwan Alakhras (ID),[1] **Mousa Hussein** (ID),[1] **and Mourad Oussalah** (ID)[2]

[1]*EE, United Arab Emirates University, UAE*
[2]*University of Oulu, Finland*

Correspondence should be addressed to Marwan Alakhras; marwan2024@yahoo.com

Building the fingerprint map for indoor localization problems is a labour-intensive and time-consuming process. However, due to its direct influence on the location estimation accuracy, finding a proper mechanism to construct the fingerprint map is essential to enhance the position estimation accuracy. Therefore, in this work, we present a fingerprint map construction technique based on location fix determination and fingerprint matching motivated by the availability of advanced sensing capabilities in smartphones to reduce the time and labour cost required for the site survey. The proposed Location Fixing and Finger Matching (LFFM) method use a landmark graph-based localization approach to automatically estimate the location fixes for the Reference Points and matching the collected fingerprints, without requiring active user participation. Experimental results show that the proposed LFFM is faster than the manual fingerprint map construction method and remarkably improves the positioning accuracy.

1. Introduction

With the increase of ubiquitous deployment of Wi-Fi infrastructures both at indoor and at outdoor environments together with the exponential multiplication of mobile computing devices, intense attention has been paid to various approaches of Wi-Fi network aided location tracking [1]. Such positioning methods are extremely useful in applications such as visitor navigation in museums or large buildings where GPS signal is not available: elderly health care positioning systems, facility management, transportation, and emergency rescue [2, 3]. In this respect, fingerprinting techniques became particularly popular for indoor localization.

More formally, utilizing the existing Wi-Fi infrastructures [4], the fingerprinting approach involves two distinct phases: offline map construction and online location estimation. In the offline phase, a site survey is conducted to collect Received Signal Strength (RSS) at known locations called Reference Points (RPs), which usually contain the Media Access Control (MAC) address of visible Access Points (APs). The RPs' coordinates along with their corresponding RSSs are then stored in a radio map database. The latter is used in the second phase, referred to testing or online phase, which, through some interpolation or machine learning-based technique, enables the system to infer the position of an unknown (mobile) target given its RSS value. Nevertheless, radio map construction often requires a large number of locations—labelled RSS fingerprints—gathered from either wireless APs or RPs, often evenly distributed within the area of interest, in order to ensure a good positioning accuracy during the testing phase. Besides, in order to account for the inherent deficit of wireless signal propagation, several samples, executed with special care, are needed at each measurement.

Therefore, and from our personal experiences as mentioned in [5], a major challenge in constructing efficient radio map is the expensive time requirement and labour-intensive site survey process, hindering a wide set of potential applications, which, in turn, promote RSS fingerprinting localization-based techniques. Besides, the complex and dynamic nature of the indoor wireless environment, which makes radio map maintenance difficult as the signal, is easily

influenced by the structures, layout, and pedestrians around the study site. In order to deal with the radio map construction challenge, many research works have been performed to expedite the site survey process while ensuring high-quality standards. For instance, the Simultaneous Localization and Mapping (SLAM) [6–8] can be used to construct the radio map with lower survey cost, but it may be unsuitable to work on the resource-limited handheld devices such as smartphones due to its heavy computation load [9]. Although many map management methods (e.g., Map Filtering [10] and pedestrian dead-reckoning [11]) can be employed to infer a user's initial location and improve SLAM performance accordingly, their associated high computational cost restricts their implementation in resource-limited ubiquitous devices. Alternatively, the crowdsourcing-based radio map construction has been proposed recently in both active and passive formats [10]. While the active crowdsourcing method requires active user participation, which reduces the need for professional surveyors, it may suffer from intentional frauds due to user participation. On the other hand, the passive crowdsourcing approach reduces user participation by utilizing information from smartphone inertial sensors in order to associate fingerprints to the corresponding RPs. Although the passive crowdsourcing is more practical, its main drawback lies in the low accuracy and the need for GPS readings, which, in turn, may constrain the application scope [12]. Therefore, using the high-end smart devices that are equipped with a variety of sensors (e.g., a barometer that detects floor change in the building) can be vital to improving user experience and crowdsourcing-based radio map construction, which, ultimately, enhances the indoor localization system and expands the applicability of the underlying localization services. However, the accumulated error of many inertial sensors grows over time for instance gyroscope; also, to the potential uncertainty pervading the locations of APs, it may force special care and a regular update that utilizes the environment knowledge.

In this respect, a new passive fingerprint crowdsourcing method has been proposed in this paper. More formally, this work proposes a landmark graph-based localization method for automatically estimating the location of RPs matched with the collected fingerprints in order to design a practical, fast, and reliable fingerprint data collection method using sensor-rich smart devices. Compared to other existing methods, our technique can significantly build a fast fingerprint map with a minimum user participation while ensuring high standards in terms of accuracy and reliability due to the incorporating of a *belief factor* that accounts for the RPs and assesses the quality of location fixes of RPs, so that only those location fixes that are associated with high quality are matched with the corresponding fingerprints. This achieves better location accuracy than Map Filtering and Pedestrian Dead Reckoning (PDR) [13]. Besides, the accuracy of the constructed radio map is compared to state-of-the-art approaches ZEE [10] and manual fingerprint map building techniques [14].

The rest of this paper is organized as follows: Section 2 describes the related work in the field. Section 3 details the proposed system, while Section 4 describes the LFFM technique based on a landmark graph, Section 5 describes the fingerprint map construction, Section 6 explains the experimental findings, and Section 7 draws the conclusion.

2. Related Work

Indoor localization has been an active area of research for the past three decades, initially in the context of mobile robot navigation and more recently in the context of pervasive and mobile computing where the issue of environment mapping is crucial for a variety of applications requiring various levels of accuracy. In this course, one distinguishes approaches that make use of existing infrastructures (e.g., Wi-Fi APs as in RADAR [15]) and those that require special deployment infrastructure (e.g., infrared beacons as in Active Badge [16] and ultrasound devices as in Cricket [17]). In the area of mobile robotics, a significant step was achieved by the emergence of Simultaneous Localization and Mapping (SLAM) which allowed a robot to build a map of the indoor environment, usually, in terms of walls and other obstruction objects, while simultaneously determining its location with respect to the constructed map.

In the SLAM group, special attention has been drawn to the offline map construction techniques that often employ motion and system models, where several approaches have been proposed, typically, such methods differ according to the employed sensors, type of prior knowledge available, and the used mapping techniques [6–8]. For instance, one may distinguish the recently populated GraphSLAM [18], which transforms the SLAM posterior into a graphical network, where a greedy algorithm is usually used for data association. Traditionally, the positions of landmarks (objects) in the environment are estimated using a laser range finder like a sensor (either fixed or scanning through the environment) embedded in some mobile platform, which yields a relatively accurate estimate of the RPs with respect to the local frame. A critical assumption is that these landmarks do not change position during the journey. In the case of nonstatic of either the mobile platform or landmarks, due to the accumulation of the relative positioning error throughout the journey, a correction step is required, yielding an updated map of the environment. For example, the WiFi-SLAM system proposed in [8] uses a Gaussian Process Latent Variable Model to associate Wi-Fi fingerprints in combination with a motion dynamics model without requiring any location labels in the training data. Authors in [19] used GraphSLAM in order to improve the computational efficiency of the WiFi-SLAM system and relieve its dependency on the requirement of fingerprint uniqueness assumption. WiSLAM [7] uses a Bayesian framework to fuse WiFi RSS with data from a foot mounted inertial measurement units for localization and mapping. In [6], a Smart SLAM is proposed to construct a radio map through a fruitful combination of different fusion algorithms, namely, PDR algorithm, Fingerprint Extended Kalman Filter (FEKF), Fingerprint Extended Kalman Filter SLAM (FEKF-SLAM), and Distributed Particle SLAM (DPSLAM). However, the heavy computation load of these SLAM systems prevents them from being implemented on

resource-limited handheld devices such as smartphones. Moreover, another disadvantage of such filtering-like techniques is that the data, once processed, is often discarded. This makes it impossible to revisit all data at the time of a map building task [18].

In contrast to those above computationally expensive map building techniques and costly labour-based site survey methods, crowdsourcing has emerged as a natural participatory model that benefits from daily user activities and widespread of smartphone devices. In this respect, one distinguishes active crowdsourcing [20–22] and passive crowdsourcing [10, 23, 24]. The active crowdsourcing methods construct a radio map by utilizing the user feedback where the surveyors are volunteers who decided to share RSS signal at specific locations of the environment. Especially, authors in [9–11] highlighted several clear advantages of such techniques. First, the underlined radio map can provide robust and accurate fingerprint data even when it is built solely on short-duration RSS measurements. Second, there is no constraint on the type and the number of devices employed. As a result, the active crowdsourcing eliminates the need for costly professional surveyors. Nevertheless, this should not hide some implicit limitations as well. For instance, since users carry heterogeneous devices, this can result in a radio map built where RSS values originated from diverse devices with distinct chipsets and antenna designs, which affect the calibration or consistency of employed radio model, even when the devices were placed at exactly the same positions. It can also result in holes in terms of data sparsity with no designated fingerprint collection points. Indeed, since the radio map is updated by untrained voluntary users without centralized controls, different users can upload their fingerprint data that is collected at slightly different locations but with the same location label. Multiple fingerprint data indicating one particular location not only cause slow location estimation but also store space wastage in a radio map server. Finally, since it requires active user participation, this technique may suffer from intentional frauds [20]. A particular example of an active crowdsource system the Organic Indoor Location (OIL) [20] that periodically asks the users to bind their measurements along with their locations on the floor plan, providing information about the nearby wireless resources, then the determined position is reflected into a global map. Similarly, FreeLoc [21] and Mobile Organic Localization Engine (Mol'e) [22] use semantic labels rather than exact floor maps to annotate fingerprints with locations such as rooms, hallways, and corridors.

On the other hand, passive crowdsourcing methods associate fingerprints with the corresponding RPs with the aid of smartphone inertial sensors. For example, EZ [23] uses occasional GPS fixes together with reported RSS measurements of specific APs arising from users' participation to build a radio map that does not require any predeployment effort. Nevertheless, the reliance on the method on the existence of occasional GPS fixes might be an issue in an indoor environment, which delays the map construction until the availability of GPS signal. Zee [10] utilizes smartphone inertial sensors to track the user while performing Wi-Fi scan simultaneously, which enabled the authors to construct a radio map in a non-

intrusive way. However, the approach uses magnetometers to calculate the direction, which is easily affected by building structure, service amenities, and furniture in indoor environments. Besides, its use of the particle filter to fuse inertial sensors with a floor plan is computationally expensive, which makes it unsuitable for running on resource-limited devices. LiFS [24] builds the radio map by exploiting the built-in sensors of the mobile device with the floor plan of the testing environment, which resulted in a relatively fast deployment process and a less labour effort. Especially, it was shown that LiFS works well in buildings exhibiting some spatial symmetry of the environment layout; e.g., office rooms are connected by a corridor and are equally distributed on both sides of the corridor, but may fail to work on other types of buildings. Although the passive crowdsourcing significantly reduces the labour cost of site surveying task and does not require active user participation, it has several problems too. This includes low accuracy, limited applicability to hand-device implementation constraints, and more importantly the requirement for GPS readings. Table 1 summarizes some of the most common fingerprint map construction techniques.

Our work falls in the class of methods that rely on the existing infrastructure, namely, Wi-Fi network, without recourse to any additional deployment or detailed floorplan and exact AP placements, which distinguishes it from works in [20, 22]. Similarly, our approach also belongs to the class of passive crowdsourcing methods in the sense of enabling random smartphone users to participate in the radio map construction task as in [21], but with the additional step of utilizing inertial device sensory information in order to estimate the RPs positions. In addition, our approach introduces a confidence factor (called belief factor) that accounts for the quality of location fixes of RPs, so that only those location fixes that are associated with high quality are matched with the corresponding fingerprints. Our approach also bears similarity with Gu et al.'s work [13] that makes use of mobile inertial sensors for position estimation but with the additional use of crowdsourcing technology and confidence factor analysis in the pattern-matching task. Likewise, our work bears similarity with EZ's approach [23] with the difference that our proposal does not require any GPS fixe locations, making use of device inertial sensors. Besides, we also restrict to cases where some prior information about the environment through floor plan is provided, which excludes fully unknown environment scenarios as in some other studies.

3. System Overview

The outline of the proposed fingerprint collecting technique based on Location Fixing and Fingerprint Matching (LFFM) is shown in Figure 1. It is composed of two main modules: Location Fixing of RPs and Fingerprint Matching.

The Location Fixing module receives measurements from the built-in smartphone sensors and is fed with a landmark graph [13]. The sensor measurements—basically, *barometer*, *accelerometer*, *gyroscope*, and *magnetometer*—are used to determine the step size and directional heading of the

TABLE 1: Fingerprint map construction techniques.

Methods	System name	Algorithm and requirements	Accuracy of RM	Testbed area
Crowdsourcing	RedPIN [25]	Label position by user, indoor map	Room level (90%)	26 rooms
	Molé [22]	Kernel, accelerometer	Room level (91%)	3-floor building
	FreeLoc [21]	Relative RSS comparison	<3 m	A laboratory, a corridor
SLAM	WiFi-SLAM [8]	GP-LVM, initial ISO map model	3.97 m (ME)	250–500 m (traces)
	SignalSLAM [26]	Least square, PDR, GraphSLAM, landmarks, accelerometer, gyroscope, magnetometer	<16.5 m (MD)	200 m × 160 m
	Graph-SLAM [18, 19]	Sparse graph, constraint optimization, least square, linearization, approximation, EKF SLAM, accelerometer, GPS	<10 m	Urban area 600 m × 800 m
Inertial sensors	Zee [10]	DR, augmented particle filter, indoor map, accelerometer, gyroscope, magnetometer	1.2 m (50%), 1.8 m (80%)	65 m × 35 m
	LiFS [24]	DR, feature extraction, indoor map, accelerometer	5.88 m (ME)	70 m × 23 m
	WILL [27]	PDR, K-means, accelerometer	Room level (86%)	70 m × 23 m
Semisupervised learning	Manifold learning [28]	Manifold alignment, inherent spatial correlation of RSS, path loss model, partial RPs, APs' locations, indoor map	3.8 m (ME); 2.4 m (ME)	40 m × 30 m, 5 APs; 40 m × 20 m, 4 APs
	Coforest [29]	Implicit crowdsourcing sampling, random forest ensemble classifier, partial RPs, RSS	3.65 m (ME)	800 m^2, 30 APs
Unsupervised learning	WRM [30, 31]	HMM, EM, memetic algorithm, path loss model, indoor map, APs' locations	Around 3 m (ME)	80 m × 32 m, 30 APs
Path loss model	ARIADNE [32]	Ray tracing, path loss model, simulated annealing algorithm, APs' location, partial RPs, indoor map	3 m (ME), 2.5 m (STD)	65 m × 48 m, 5 APs
	Multiwall Path Loss Model (MWM) [33]	MWM, APs' location, indoor map, parameters setting for Gaussian distribution, Euclidean distance error, kNN	1.2 m (ME)	480 m^2, 3 APs
Interpolation	Inverse Distance Weighting (IDW) [34]	RSS, interpolation and extrapolation methods, estimation error statistics, uniform grid, IDW, probabilistic positioning	5~20 m (ME)	150 m × 60 m, 316 2.4 GHz APs, 106 5 GHz APs
	Kriging [35]	Kriging algorithm, spatial interpolation, semivariogram model fitting, unbiased estimation, RSS, K-weighted nearest neighbours,	1.12 m (ME)	9.5 m × 2.5 m, 9 APs
	Forward Interpolation [36]	cubic spline, boundary condition (fixed, zero-slope natural, nonnode), RMS, RSS	2.82 m (best)	5 rooms, 4 APs

particular user using the well-known Pedestrian Dead Reckoning (PDR) method described in [37]. The estimated step size and directional heading are then employed to infer the current user position in real time with respect to the landmarks according to the provided landmark graph. In this respect, a landmark processing task is necessary in order to reduce the error accumulation of the PDR, in addition to position calibration based on landmark detection.

The Fingerprint Matching module uses the fixed locations estimated by the Location Fixing module and the scanned Wi-Fi signal information (MAC address and corresponding RSS) to generate the fingerprint radio map. Location estimates together with the corresponding match fingerprint are then added into the radio map—only if they meet a certain quality requirement, which will be detailed

later on. This constraint is added to guarantee the accuracy and reliability of the generated fingerprint map. More details on the algorithm will be elaborated in the next section. However, initially, Table 2 will help the reader to navigate through the article by providing a list of abbreviations.

4. Location Fixing and Fingerprint Matching (LFFM) Based on Landmark Graph

4.1. Landmark Detection. Our approach utilizes the fact that naturally spread smartphones in indoor environments are rich with powerful sensors and sensing ability. In our work, a landmark will refer to spatial points where sensor measurements indicate an identifiable and distinguishable pattern or change in pattern. For example; corners and turns will force
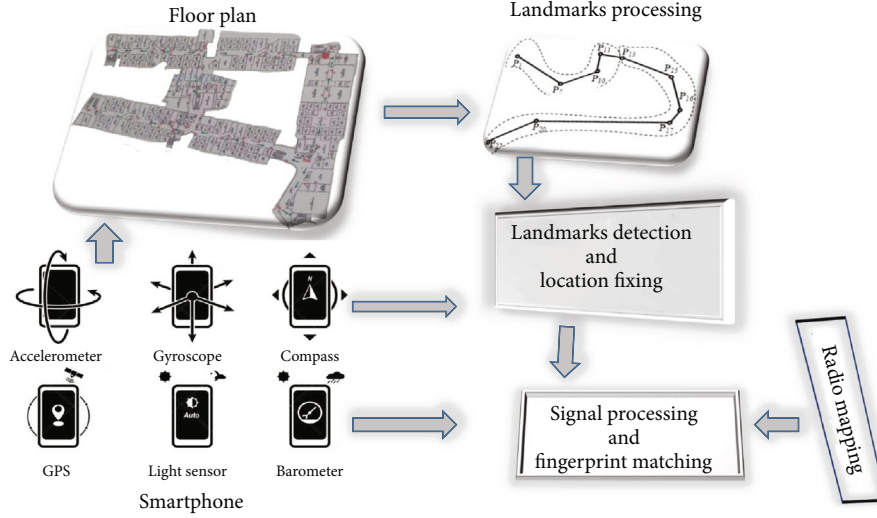
Figure 1: System architecture.

users to change walking direction, while, stairs and elevators are highlighted by elevation change. Similarly, doors are triggered whenever a change in the velocity is identified, and thus, a movement pattern can be predicted.

For this purpose, we utilize information issued from inertial sensors, mainly, barometer, gyroscope, and accelerometer for landmark detection. The detection of landmarks is based on the distinct change of pattern in one or more types of sensor readings. The locations of these landmarks correspond to the locations of elevators, stairs, corners, turns, and doors that can be simply obtained from floor plans, which is usually available for most indoor environments.

More formally, a landmark L can be defined as

$$L : <(x, y, z), (R_1, \cdots, R_N) >, \quad (1)$$

where (x, y, z) denotes the location of the landmark according to the underlined floor plan and $(R_1, \cdots, R_N)$ represents the set of detection rules, according to the different types of sensor measurements, satisfied by the corresponding landmark, where N corresponds to the number of rules triggered by this landmark. Typically, each device inertial sensor would trigger one or more rules associated to the constraints on the measurement values/patterns linked to each landmark type.

4.1.1. Barometer-Based Landmark. The barometer sensor measurements provide information about vertical movement detection, for example, a user going upstairs, going downstairs, or taking an elevator, since the barometric value changes with the altitude or height. Although the barometric pressure is influenced by many factors such as temperature and altitude, due to the relatively instantaneous measurement time, such factors are discarded in this work, and we only account for altitude information. The change in the barometer readings, when a user walks horizontally, upstairs/downstairs, or takes the elevator, can be shown in Figure 2.

Here, the entrance and exit of stairs and elevators are considered barometer landmarks due to their pattern representation in barometer pressure measurements, since it is distinctive, identifiable, and stable. The entrance detection includes a "horizontal movement→vertical movement" pattern. Similarly, the exit detection includes a "vertical movement→horizontal movement" pattern.

More formally, having observed the change in barometer readings as shown in Figure 2, where it is almost a linear change when the user moves vertically, in this case, the linear model can be used to fit those readings; then, let p_i denote the pressure of the ith window of air pressure readings at time t, and let β_{bar_1} and β_{bar_2} be the thresholds used to detect horizontal movement and vertical movement, respectively. Then R_{bar} rule to detect the entrance to a set of staircases or an elevator can be defined by the following rule:

$$R_{\mathrm{bar}_1} : \left(\mathrm{loc}_t \text{ when } (|p_i - p_{i-1}|) < \beta_{\mathrm{bar}_1} \text{ AND } |S_1| \right. \\ = = K_{p_1} \text{ AND } \left| p_i + K_{p_1} - p_i \right| > \beta_{\mathrm{bar}_2} \right). \quad (2)$$

The first term is for detecting horizontal movement, and the latter two terms are for detecting vertical movement. The thresholds β_{bar_1} and β_{bar_2} on horizontal and vertical movements are set manually after an extensive testing phase. Although many algorithms can be used to determine the thresholds experimentally [38], for the time being, it is out of the scope of this work, thus all the thresholds are set manually based on the empirical test as shown in Table 2, for example, and as illustrated in Figure 2, β_{bar_1} can be calculated as $(1013.4 - 1013.2)/4 \text{ sec} = 0.05 \text{ hPa}$ for every second in case of going up or down on the stairs.

The function S_1 can be defined as

$$S_1 = \sum_{j=i+1}^{i+K_{p_1}} \left(\mathrm{sgn} \left(p_j - p_{j-1} \right) \right), \quad (3)$$

TABLE 2: List of abbreviations.

R_{bar}	Barometer landmark rule
R_{acc}	Accelerometer landmark rule
R_{gyr}	Gyroscope landmark rule
β_{bar_1}	Barometer horizontal movement threshold
β_{bar_2}	Barometer vertical movement threshold
p_i	Pressure of the ith window of air pressure readings at time t
m_t	Represents the motion state (e.g., walking and stationary) at time t
β_1	Accelerometer threshold for walking state
β_2	Accelerometer threshold for steady state
$\dot{\theta}_t$	Gyroscope measurement along the vertical direction
β_{gyr}	Gyroscope threshold
S_t	Step length
θ_t	Heading direction
P_t	Pressure values at time t
$P_{\mathrm{diff}_{(a,b)}}$	Pressure difference between floor a and floor b
$\mathrm{con}(l_k)$	Confidence level of land mark
$\delta(R_k, R_t^*)$	Validity of rule detection
$r(\theta_k, \theta_t^*)$	Validity of path detection
$g(d_k, d_t^*)$	Ratio of moved distance
β_θ	Heading direction threshold
β_{con}	Confidence threshold
P_S	Path segment connecting two neighbour landmarks
$\mathrm{bel}(P_S)$	Quality belief of path segment P_S
T	Step periodicity
$\beta_{\mathrm{bel}(P_S)}$	Threshold of belief value $\mathrm{bel}(P_S)$ of the path segment P_S
T_r	Trajectory

where sgn is the signum function, which can be described as

$$\mathrm{sgn}\left(p_j - p_{j-1}\right) = \begin{cases} 1, & \text{if } p_j > p_{j-1}, \\ 0, & \text{if } p_j = p_{j-1}, \\ -1, & \text{if } p_j < p_{j-1}. \end{cases} \quad (4)$$

Similarly, the rule to detect the exit from a set of staircases or an elevator can be defined by the following rule:

$$R_{\mathrm{bar}_2} : \left(\mathrm{loc}_t \text{ when } (|p_i - p_{i+1}|) < \beta_{\mathrm{bar}_1} \text{ AND } |S_2|\right.$$
$$\left. = = K_{p_2} \text{ AND } \left|p_i + K_{p_2} - p_i\right| > \beta_{\mathrm{bar}_2}\right). \quad (5)$$

Then the function S_2 can be defined in the same spirit of (2):

$$S_2 = \sum_{j=i-K_{p_2}+1}^{i+K_{p_1}} \left(\mathrm{sgn}\left(p_j - p_{j-1}\right)\right). \quad (6)$$

The two values K_{p1} and K_{p2} are not constant, but are determined dynamically. Their initial values can be set to 1 and gradually increase as long as the value of the signum function was kept unchanged.

4.1.2. Accelerometer-Based Landmark. The motion that presents a distinct change pattern can also be measured by the accelerometer. It can be considered, where a point that witnesses the changing pattern of "walking→steady→walking" (of course for an acceptable short period, i.e., one second or few seconds) can be regarded as a potential accelerometer landmark (the term *still* or *steady* refers to motionless action). This pattern may happen when passing doors as shown in Figure 3 or passing water fountain, which can be detected by comparing the magnitude of the accelerometer measurement against a predefined threshold (which is again set based on empirical results). A location point is regarded as an accelerometer landmark if the accelerometer readings present this changing pattern every time the user passes it. Formally, the rule R_{acc} of accelerometer landmarks is defined by the following rule:

$$R_{\mathrm{acc}} : \left(\mathrm{loc}_t \text{ when } m_{t-\beta_1:t} \text{ is walking AND } m_{(t:t+\beta_2)}\right.$$
$$\left. \text{is steady AND } m_{t+\beta_1} \text{ is walking}\right), \quad (7)$$

where m_t represents the motion state (e.g, walking, steady) at time t. β_1 and β_2 are two thresholds that are set empirically to determine the period of the corresponding motion state.

4.1.3. Gyroscope-Based Landmark. A location point where the gyroscope measurements present a distinct and stable pattern is considered a gyroscope landmark. The magnetometer can also be used to detect a change in direction, but its measurements tend to be affected by ferromagnetic materials. Therefore, gyroscope measurements seem to be more suitable for detecting the change in direction in our experiment. The target is to determine the right and left turn patterns as shown in Figure 4. This pattern can usually be witnessed at the location of a turn, corner, or door. The rule R_{gyr} to detect a gyroscope landmark can be set as

$$R_{\mathrm{gyr}} : \left(\mathrm{loc}_t \text{ when } \left|\dot{\theta}_t\right| > \beta_{\mathrm{gyr}}\right), \quad (8)$$

where $\dot{\theta}_t$ is the gyroscope measurement along the vertical direction. When the absolute value of $\dot{\theta}_t$ is greater than a certain threshold β_{gyr}, this location point can be considered a potential gyroscope landmark. Trivially, the initial direction heading can be set to north or south depending on the underlying environment orientation.
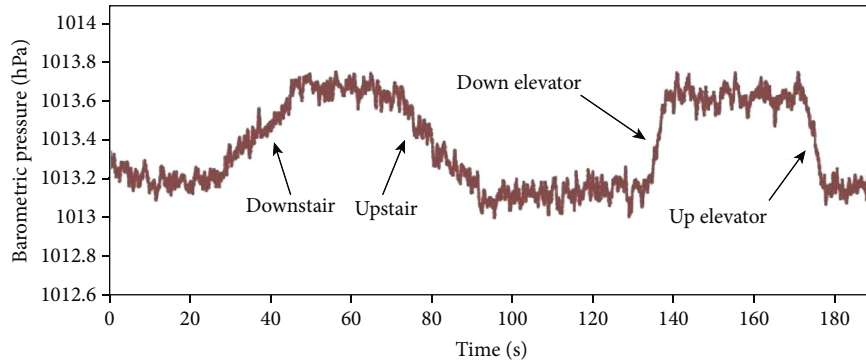
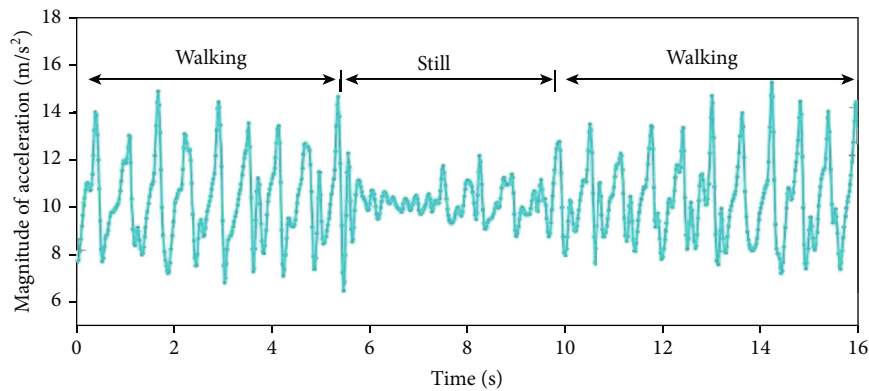FIGURE 2: Barometer readings on changing elevations.

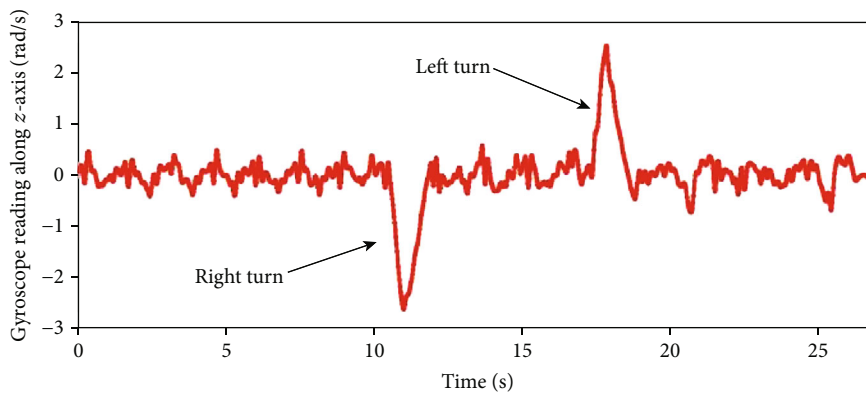FIGURE 3: Accelerometer reading example when passing door landmark.

FIGURE 4: Gyroscope readings on right and left turns.

4.2. Landmark Graph for Location Fixing of RPs. A landmark graph can be defined as a directed graph where nodes are landmarks and edges are accessible paths with heading information.

Let $G = (L, E)$ denote a landmark graph where $L = \{l_1, l_2, \cdots, l_N\}$ is a set of landmarks and $E = \{e_1, e_2, \cdots, e_M\}$ is the set of edges in graph G. Each edge e_i consists of the two landmarks, direction from one landmark to another, and the corresponding distance between them, $e_i = <l_j, l_k, \theta_{jk}, d_i>$. Note that the direction from landmark l_j to landmark l_k is different from that from l_k to landmark l_j. In other words, there are two edges between any two neighbour land-

marks. The general algorithm for Location Fixing and Fingerprint Matching (LFFM) using a landmark graph in indoor localization is shown in Figure 5.

The details of each task in Figure 5 is commented as follows:

(1) The construction of the landmark graph requires location landmarks, which can be extracted from a floor plan. Most buildings are symmetric. For instance, floors of the same building have a similar or almost identical layout, so that by changing floor information on its landmark graph, we can easily obtain the landmark graph for another floor
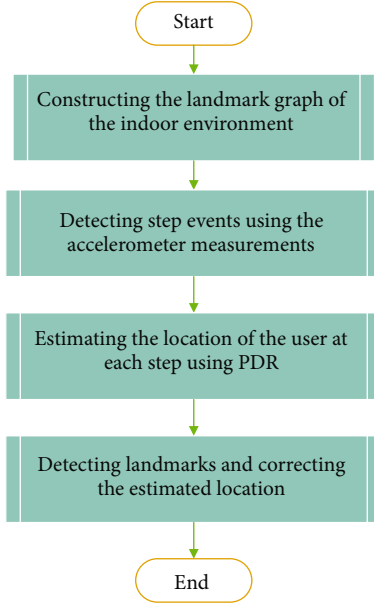
FIGURE 5: Landmark detection algorithm.

(2) The step detection is done via detecting acceleration peaks, where each acceleration peak corresponds to one step. In order to distinguish the walking state from the steady state, we check if the change of the acceleration within the detected time window is greater than the set threshold. If a step is detected, then the corresponding heading direction and step size are computed

(3) Using the inertial sensor measurements and the PDR technique, the user's position at each step can be calculated given his/her initial location as follows:

$$x_t = x_{t-1} + S_t \sin(\theta_t), \tag{9}$$

$$y_t = y_{t-1} + S_t \sin(\theta_t), \tag{10}$$

$$f_t = f_{t-1} - \frac{P_t - P_{t-1}}{P_{\text{diff}_{(a,b)}}}, \tag{11}$$

where x_t and y_t are the user coordinates at floor f_t at time t and S_t is the corresponding step length and θ_t the heading direction. P_t and P_{t-1} are the air pressure values at time t and $t-1$, respectively. And $P_{\text{diff}_{(a,b)}}$ is the air pressure difference between floor a and floor b. Adding the threshold values for both step detection and direction heading will ensure values and of course a robust position when these landmarks are added later to the localization map

(4) While the PDR is conducted to estimate the location of the user, the measurements from the barometer, gyroscope, and accelerometer are simultaneously used to detect landmarks. Although both the gyroscope readings and compass readings (inferred from accelerometer readings and magnetometer readings)

can be used to estimate the heading, they cannot provide a robust heading estimation since the gyroscope has the drift problem and the compass is vulnerable to ferromagnetic materials [39]. Therefore, the landmark graph is used to assist with the heading direction estimation. If a user is detected to walk on the path connecting two landmarks in the landmark graph, the heading from this landmark graph will be used. Otherwise, the compass readings will be used as the heading. Also, the step size is updated when a user passes two neighbouring landmarks in the landmark graph. Let l_1 and l_2 indicate the two neighbour landmarks that a user passes subsequently. Given the (x, y) coordinates of each of these landmarks, and the number N_S of detected steps between them; then the step size S can be calculated as follows:

$$S = \frac{\sqrt{\left(X_{l_1} - X_{l_2}\right)^2 + \left(Y_{l_1} - Y_{l_2}\right)^2}}{N_S} \tag{12}$$

The step size estimation method does not require the user's stature information and can adapt to varying walking speeds, and it is updated as the user passes two neighbour landmarks on the landmark graph.

Using landmarks for assisting localization may result in a need to solve the data association issue [40], in other words, when there are multiple landmarks nearby, or when there is a case that one or more landmarks are missed from sensor data, or mainly it can be used to avoid false detections.

We need to have a measure of how much confidence a location points meeting the landmark detection rule. The confidence that a location point is matching should be constrained by the following facts; first, the detected landmark should be a valid detection for one of the landmark types; second, it has to fall within the same moving direction; in other words, there should be somehow a path between the estimated landmark at a specific time and the landmark at an earlier time; third, it has to have an acceptable distance from the previous detection. Formally, let l_k be a landmark then the confidence $\text{con}(l_k)$ in the landmark graph can be calculated as

$$\text{con}(l_k) = \delta(R_k, R_t^*) \cdot r(\theta_k, \theta_t^*) \cdot g(d_k, d_t^*), \tag{13}$$

where k is the index of a landmark in the landmark graph, R_k is the detection rule of the reference landmark l_k, and R_t^* is the type of the detected landmark at time t; then $\delta(R_k, R_t^*)$ is the validity function of the detected landmark, with δ as the Dirac delta function that is denoted as

$$\delta(R_k, R_t^*) = \begin{cases} 1, & \text{if } R_k = R_t^*, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

From (13), where θ_k and θ_t^* are the reference heading and the estimated heading from the time visiting the last

landmark to time t. Then, the path validity r is the rectangle function adopted from our earlier work [14] and can be described as

$$r(\theta_k, \theta_t^*) = \begin{cases} 1, & \text{if } |\theta_k - \theta_t^*| < \beta_\theta, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where β_θ is a heading direction threshold, when d_k and d_t^* are the reference distance and the traveled distance from the last landmark to the location point. Then the distance weight function g is defined as

$$g(d_k, d_t^*) = \frac{1}{|d_k - d_t^*|}. \tag{16}$$

Once every landmark is being associated with a confidence level, then, when multiple landmarks are nearby, the one with the highest confidence factor will be selected. Moreover, fake landmarks may occur due to user unpredictable actions, for instance making a turn in the middle of the corridor, which may be resulting in detection of a gyroscope landmark. To solve this issue, a confidence threshold β_{con} value may be used as such as

$$l_k : \text{con}(l_k) \geq \beta_{\text{con}}, \quad \forall l_k : (x, y)_t. \tag{17}$$

In some other cases, when landmarks are missed from sensor data, for instance, certain landmarks at the locations of doors can be missed if a door is left open since "walking→steady→walking" behaviour pattern will not be detected. In such a case, we have no option but to ignore the underlined landmark, which may result in increasing the accumulation error. But this error will be reduced back when the next landmark is detected since after the addition of every landmark the graph will be denser, having more nodes and, definitely, more connective paths, which will result in reducing the chance of error for the upcoming detections.

5. Construction of Fingerprint Map

5.1. Quality of Location Fixes of RPs. The construction of a fingerprint map is one essential step towards fine position estimation. Therefore, any location fix is subject to quality check before being matched to any fingerprint and added to the fingerprint map. As explained in the earlier section, the PDR method consists of two components: *step size estimation* and *heading direction estimation*. So, a robust algorithm will tend to bind the accumulative error of these two components. In [41], it has been proved that the step periodicity for the same motion state (e.g., walking and jogging) suffers from small variation when a user moves at a relatively constant speed. However, this will significantly vary when a user remains stationary (steady) while using the smartphone arbitrarily, for texting, playing phone games, etc. Therefore, limiting the step periodicity to a certain interval will reduce the

location estimation error. The step periodicity can be defined as the period of one step, which is equal to the time difference between two neighbour peaks of the accelerometer measurements as shown in Figure 3. Let P_S denote a path segment connecting two neighbour landmarks with N steps in between and $N + 1$ RPs.

$$P_S = \{(t_i, x_i, y_i, f_i), \quad \forall i = 1, 2, \cdots, N + 1\}, \tag{18}$$

where x_i and y_i are the coordinates of RP_i at time t_i with corresponding floor information f_i. And let $T = \{T_1, \cdots, T_N\}$ indicate the step periodicity set measured by the accelerometer for these N steps. Then the quality belief of path segment P_S going through those RP can be evaluated as

$$\text{bel}(P_S) = \frac{\sum_{i=1}^N \mathcal{X}(T_i) \cdot T_i}{\sum_{i=1}^N T_i} \cdot \frac{1}{\sigma_{T^*}}, T^* \subseteq T, \quad \forall T_{\min} \leq T \leq T_{\max}, \tag{19}$$

where the first term indicates the ratio of valid steps with respect to the total number of steps after outliering false walks or frequent stops. The outliering step is carried out using the identifier function $\mathcal{X}(T_i)$, which is defined as

$$X(T_i) = \begin{cases} 1, & \text{if } T_i \in [T_{\min}, T_{\max}], \\ 0, & \text{otherwise}. \end{cases} \tag{20}$$

The second term is reciprocal of the standard deviation of the valid time set T^*, which includes the valid walks only after outliering the fake walks. In this case, if the user walks with constant speed, then the deviation will be too small. Finally, the belief value $\text{bel}(P_S)$ of the path segment P_S is compared to a certain threshold $\beta_{\text{bel}(P_S)}$, and the location fix is considered accurate and reliable to be matched with and then added to the fingerprint map.

5.2. Fingerprint Matching and Construction of Fingerprint Map. It is necessary to evaluate the estimated location fix quality before using fingerprint matching. Because the accuracy of the RPs' location fix estimation has a direct impact on the accuracy of the construction of the fingerprint map.

Now, let T_r denote the trajectory that the user has traveled, including K path segments, which are divided by landmarks, namely, $T_r = \{T_1, \cdots, T_K\}$. And the set of fingerprints collected along this trajectory $FP = \{fp_1, \cdots, fp_N\}$, where N is the number of Wi-Fi scans, each fingerprint contains the RSS and the MAC address of its corresponding RP, since the time when a walk state event happens may be different from the time when the WiFi scan is conducted. Synchronizing the time to conduct the Wi-Fi scan with the time the walk event happens is necessary in order to match the fingerprint with the estimated location fixes. Suppose that the Wi-Fi scan at time t_j happens during the time period the user walks from the location (x_{k-1}, y_{k-1}) to (x_k, y_k), namely, $t_{k-1} \leq t_j \leq t_k$; then

```
LFFM algorithm
Input:
        initila location (X_0, Y_0, f_0)
        Trajectory T_r = {T_1, ⋯, T_k}
        vector of
        FP = {(RSS_1, MAC_{AP_1})_1, (RSS_2, MAC_{AP_2})_2, ⋯, (RSS_i, MAC_{AP_i})_N}
Output:
        Fingerprint Map
for j = 1 : N do
    Compute the location (x_j, y_j, f_j) of the j-th potential Location fix;
    Search for the P_{S_i} that includes location (x_j, y_j, f_j)
    Compute the belief bel(P_{S_i}) of P_{S_i}
    if bel(P_{S_i}) > β_{bel} then
        Match the fingerprint PF_j and the corresponding location (x_j, y_j, f_j)
        Add the tuple (x_j, y_j, f_j FP_j) to the fingerprint map;
    end if
end for
```

ALGORITHM 1: Location Fixing and Fingerprint Matching.

FIGURE 6: Experiment test bed layout.

we can estimate the location of the jth potential location fix RPs by using a linear interpolation:

$$x_j = x_{(k-1)} + \frac{(x_k - x_{k-1}) \cdot (t_j - t_{k-1})}{t_k - t_{k-1}}, \qquad (21)$$

$$y_j = y_{(k-1)} + \frac{(y_k - y_{k-1}) \cdot (t_j - t_{k-1})}{t_k - t_{k-1}}. \qquad (22)$$

According to the calculated location (x_j, y_j), then the path segment P_{S_i} can be found, such that it includes the location (x_j, y_j, f_j). After this, we evaluate the quality of P_{S_i}, where only when the $\text{bel}(P_{S_i})$ meets the threshold requirement—it is essential to evaluate the quality since this will affect the map construction, and therefore, will affect the localization process—later on, the fingerprint is matched

with the corresponding location fix and then is added to the fingerprint map in term of RSS_j, MAC_{AP_j}, and its corresponding location (x_j, y_j, f_j). This process is recursively done until all the elements in the FP set are used. A summary of the process is shown in Algorithm 1.

6. Experiments and Results

The proposed method was evaluated by experiments conducted in a three-story office building. The area of each floor is about 6,750 square meters. The testing path goes through two floors of this building, and its length is about 420 meters as shown in Figure 6. The building is made of precast concrete, blockwork, steel structure, aluminium, ceramic tiles, and zinc-coated materials and also offices and classroom furniture, lab equipment (electrical, electronic and mechanical), and it has many other home

TABLE 3: Parameters of experimental results.

| Function	Parameter	Value
Barometer landmark	Pressure threshold β_{bar1}	0.05 hPa
	Pressure threshold β_{bar2}	0.3 hPa
Accelerometer landmark	Window size	50 samples
	Walking-state threshold	2 s
	Steady-state threshold	1-5 s
Gyroscope landmark	Window size	10 samples
	Gyro threshold β_{gyr}	1.1 rad/s
PDR	Pressure difference PR_{diff}	0.45 hPa
	Heading threshold β_θ	30°
	Confidence threshold β_{con}	0.25
	Initial step size	0.63 m
Quality evaluation	Step periodicity threshold T_{max}	1 s
	Step periodicity threshold T_{min}	0.4 s
	Belief threshold β_S	15
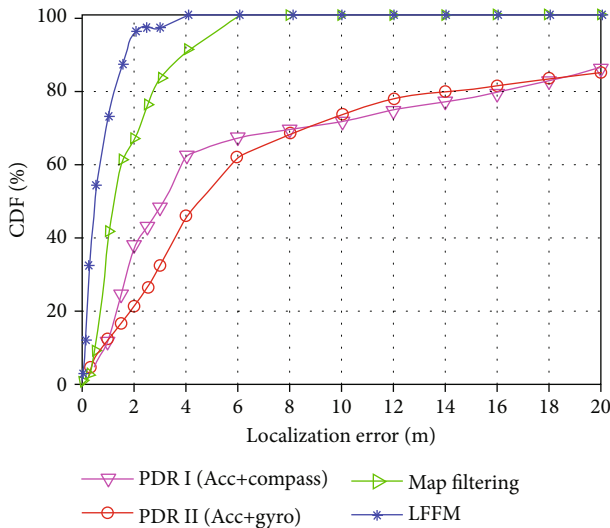
FIGURE 7: Localization accuracy comparison.

appliances. Using this environment and the test bed in Figure 6, a set of experiments were carried out to evaluate the proposed method.

The device used in the experiments is a Samsung Galaxy Note 3 smartphone equipped with Wi-Fi, accelerometer, magnetometer, gyroscope, and barometer. An android app was developed to collect the sensor data. A test user walked along the preset path with the phone in hand and clicked on the app to record the measurements of sensors and collect information about location fixes of RPs to evaluate the location accuracy. The data recorded include the MAC address of visible APs and corresponding RSS and readings from the accelerometer, gyroscope, and barometer. The values of parameters used in this work were empirically determined as shown in Table 3.
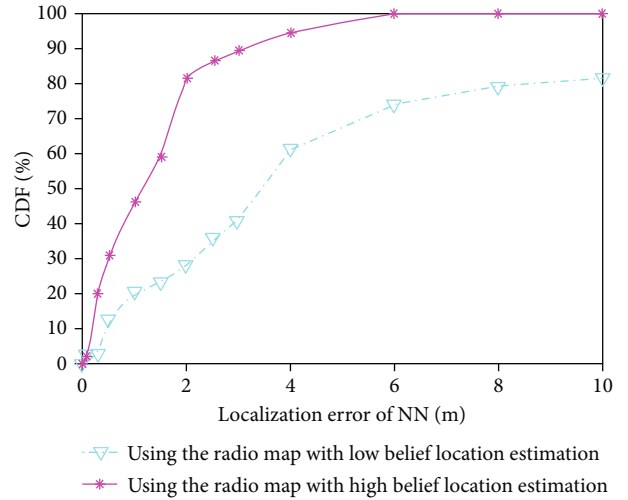

FIGURE 8: Localization error for kNN with different belief thresholds.

The results of LFFM based on the landmark graph for indoor localization method are compared with those of commonly used PDR-I and PDR-II methods in [13] and Map Filtering method in [42]. Both PDR methods used the step-counting approach to estimate the step size; however, to estimate the heading direction, PDR-I used the compass measurements, while PDR-II used the gyroscope measurements, which is almost the same as the sensors used in our work. The cumulative distribution of localization errors shown in Figure 7 shows that our method significantly outperforms the other methods, achieving a mean error of 0.71 meters.

Another experiment was carried out on a long straight path so that the heading direction could be ignored. This was necessary to evaluate the accuracy of location fixes, where in the first part the user walked along the straight distance with constant speed and in the second part the user walked the same distance with varying speeds and stopped at a few locations to imitate a fake walk status. The localization average error was less than 1 meter in the first part and about 5 meters in the second part.

Two different fingerprint maps were created with different belief thresholds introduced; later those maps were fused into the well-known algorithm k-nearest neighbour (kNN)—in the same spirit as in [5]—the localization error of kNN is shown in Figure 8. It can be seen that using the location estimation with high belief ($bel(P_S) > 18$) to match fingerprints for constructing a radio map achieves much higher accuracy than the one with low belief ($bel(P_S) < 10$). Therefore, it is necessary to control the quality of location estimation that is used to match fingerprints with location fixes, which has a direct effect on the accuracy of the fingerprinting localization method.

The best way to evaluate the constructed maps can be done by comparing the localization results using a map constructed using LFFM and another map. In this sense, another comparison was done for fingerprint map construction using the Zee system proposed in [10], and the manual fingerprint map construction technique, despite the huge time cost for
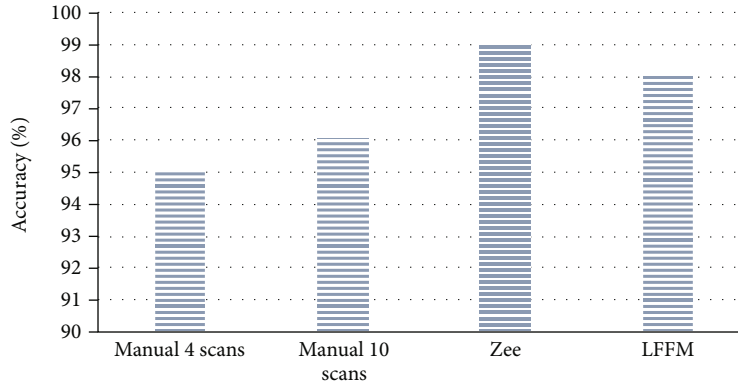
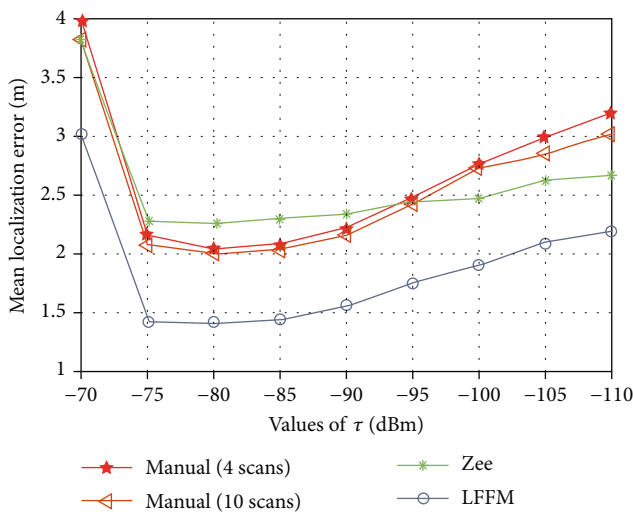FIGURE 9: System recognition accuracy at -80 dBm.

FIGURE 10: Effect of RSS threshold on the mean localization error.

constructing the fingerprint map manually, where RSS data has to be collected from a grid of points on the floor map. In this work two various Wi-Fi scans were made, the first with 4 Wi-Fi scans at each grid point, and the second with 10 Wi-Fi scans. These various scans seemed to be necessary to compensate for the orientation of the user, which is a very well-known problem in the fingerprint-based techniques. A total of 300 grid points were placed on the floor with a 2-meter distance interval. The localization errors were compared after fusing the resulting fingerprint maps into the kNN. The positive value fingerprint representation was used [43], namely, $\text{positive}_i(fp) = (RSS_i - \tau)$ if the ith AP is present in the fingerprint fp and $RSS_i > \tau$, where RSS_i is the Received Signal Strength from the ith AP and τ is a threshold value (APs whose RSS were lower than the threshold are considered not detected); otherwise, $\text{positive}_i(fp) = 0$. The accuracy of applied method is shown in Figure 9.

The effect of RSS threshold usually has a direct effect on the localization accuracy. Figure 10 demonstrates that the best performance of all the methods is achieved when the RSS threshold τ was set within [-75, -85] dBm; in our experiments, we have set $\tau = -80$ dBm. A larger or smaller value of τ will lead to an increase in the mean localization error. This

is because increasing the RSS threshold may introduce Access Points with very weak signals, which are vulnerable to human movements, and decreasing the RSS threshold would exclude some useful Access Points that can help improve the localization accuracy.

7. Conclusion

This paper presented a fast and reliable Location Fixing and Fingerprint Matching to build a fingerprint map based on a landmark graph. Compared to the existing methods, namely ZEE and manual fingerprint map constructions performed in our earlier works, the new method outperforms the manual method and is almost equally as fast as the ZEE method. Moreover, it does not require active user participation; it takes less time and effort and can construct an accurate fingerprint map with the addition of the belief factor.

Data Availability

The data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communication Surveys and Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[2] T. Cura, "A parallel local search approach to solving the uncapacitated warehouse location problem," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 1000–1009, 2010.

[3] Y. S. Lee, J. W. Park, and L. Barolli, "A localization algorithm based on AOA for ad-hoc sensor networks," *Mobile Information Systems*, vol. 8, no. 1, 72 pages, 2012.

[4] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[5] M. Alakhras, M. Hussein, and M. Oussalah, "Multivariable fuzzy inference with multi nearest neighbour for indoor WLAN localization based on RSS fingerprint," in *2013 UKSim 15th International Conference on Computer Modelling and Simulation*, pp. 656–662, Cambridge, UK, 2013.

[6] R. M. Faragher and R. K. Harle, "SmartSLAM-an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing," in *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013)*, pp. 1006–1019, Nashville Convention Center, Nashville, Tennessee, September 2013.

[7] L. Bruno and P. Robertson, "WiSLAM: improving FootSLAM with WiFi," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–10, Guimaraes, Portugal, 2011.

[8] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proceedings of the 20th international joint conference on Artifical intelligence*, pp. 2480–2485, Hyderabad, India, January 2007.

[9] D. Wagner and D. Schmalstieg, "History and future of tracking for mobile phone augmented reality," in *2009 International Symposium on Ubiquitous Virtual Reality*, pp. 7–10, Gwangju, South Korea, 2009.

[10] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization Anshul," in *Proceedings of the 18th annual international conference on Mobile computing and networking - Mobicom '12*, pp. 293–304, Istanbul, Turkey, 2012.

[11] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, pp. 114–123, Seoul, Korea, 2008.

[12] S. Jung, S. Lee, and D. Han, "A crowdsourcing-based global indoor positioning and navigation system," *Pervasive and Mobile Computing*, vol. 31, pp. 94–106, 2016.

[13] F. Gu, A. Kealy, K. Khoshelham, and J. Shang, "Efficient and accurate indoor localization using landmark graphs," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B2, pp. 509–514, 2016.

[14] M. Oussalah, M. Alakhras, and M. I. Hussein, "Multivariable fuzzy inference system for fingerprinting indoor localization," *Fuzzy Sets and Systems*, vol. 269, pp. 65–89, 2015.

[15] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, pp. 775–784, Tel Aviv, Israel, 2000.

[16] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.

[17] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, pp. 32–43, Boston, MA, USA, 2000.

[18] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5–6, pp. 403–429, 2006.

[19] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor WiFi Graph-SLAM," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1038–1043, Shanghai, China, 2011.

[20] J.-g. Park, B. Charrow, D. Curtis et al., "Growing an organic indoor location system," in *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, pp. 271–284, San Francisco, CA, USA, 2010.

[21] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: calibration-free crowdsourced indoor localization," in *2013 Proceedings IEEE INFOCOM*, pp. 2481–2489, Turin, Italy, 2013.

[22] J. Ledlie, J. Park, D. Curtis, A. Cavalcante, and L. Camara, "Mole: a scalable, user-generated WiFi positioning engine," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–10, Guimaraes, Portugal, 2011.

[23] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking - MobiCom '10*, pp. 173–184, Chicago, IL, USA, 2010.

[24] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th annual international conference on Mobile computing and networking - Mobicom '12*, pp. 269–280, ACM Press, Istanbul, Turkey, 2012.

[25] P. Bolliger, "Redpin - adaptive, zero-configuration indoor localization through user collaboration," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments - MELT '08*, pp. 55–60, San Francisco, CA, USA, 2008.

[26] P. Mirowski, T. K. Ho, S. Yi, and M. MacDonald, "Signal-SLAM: simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals," in *International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–10, Montbeliard-Belfort, France, 2013.

[27] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[28] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed, "Joint indoor localization and radio map construction with limited deployment load," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1031–1043, 2015.

[29] C. Song and J. Wang, "WLAN fingerprint indoor positioning strategy based on implicit crowdsourcing and semi-supervised learning," *ISPRS International Journal of Geo-Information*, vol. 6, no. 11, p. 356, 2017.

[30] S. Jung, B. Moon, and D. Han, "Unsupervised learning for crowdsourced indoor localization in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2892–2906, 2016.

[31] S.-H. Jung and D. Han, "Automated construction and maintenance of Wi-Fi radio maps for crowdsourcing-based indoor positioning systems," *IEEE Access*, vol. 6, pp. 1764–1777, 2018.

[32] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "ARIADNE: a dynamic indoor signal map construction and localization system," in *Proceedings of the 4th international conference on Mobile systems, applications and services - MobiSys 2006*, pp. 151–164, Uppsala, Sweden, 2006.

[33] I. H. Alshami, A. Ahmad, and S. Sahibuddin, "Automatic WLAN fingerprint radio map generation for accurate indoor positioning based on signal path loss model," *ARPN Journal of*

Engineering and Applied Sciences, vol. 10, no. 23, pp. 17930–17936, 2015.

[34] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-based interpolation and extrapolation methods for RSS-based localization with indoor wireless signals," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1340–1353, 2015.

[35] S.-S. Jan, S.-J. Yeh, and Y.-W. Liu, "Received signal strength database interpolation by kriging for a Wi-Fi indoor positioning system," *Sensors*, vol. 15, no. 9, pp. 21377–21393, 2015.

[36] L. Peng, J. Han, W. Meng, and J. Liu, "Research on radio-map construction in indoor WLAN positioning system," in *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*, pp. 1073–1077, Harbin, China, 2010.

[37] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12*, pp. 197–210, Low Wood Bay, Lake District, UK, 2012.

[38] C. Dinh, D. Tantinger, and M. Struck, "Automatic emergency detection using commercial accelerometers and knowledge-based methods," in *2009 36th Annual Computers in Cardiology Conference (CinC)*, pp. 485–488, Park City, UT, USA, 2009.

[39] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[40] X. Qu, *Landmark Based Localization : Detection and Update of Landmarks with Uncertainty Analysis*, University Paris-Est, 2016.

[41] F. Gu, K. Khoshelham, J. Shang, F. Yu, and Z. Wei, "Robust and accurate smartphone-based step counting for indoor localization," *IEEE Sensors Journal*, vol. 17, no. 11, pp. 3453–3460, 2017.

[42] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, pp. 421–430, Pittsburgh, Pennsylvania, 2012.

[43] J. Torres-Sospedra, R. Montoliu, S. Trilles, Ó. Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.