

# Landmark Graph-Based Indoor Localization

Fuqiang Gu<sup>1</sup>, *Member, IEEE*, Shahrokh Valaee<sup>2</sup>, *Fellow, IEEE*,  
Kourosh Khoshelham<sup>1</sup>, Jianga Shang<sup>1</sup>, *Member, IEEE*, and Rui Zhang<sup>1</sup>, *Senior Member, IEEE*

**Abstract**—Indoor localization is important for a variety of applications, such as location-based services, mobile social networks, and emergency response. Fusing spatial information is an effective way to achieve accurate indoor localization with little or with no need for extra hardware. However, the existing indoor localization methods that make use of spatial information are either computationally expensive or sensitive to the completeness of landmarks. In this article, we propose a novel, low-cost, high-accuracy indoor localization method based on a landmark graph. The experimental results show that the proposed method outperforms the state-of-the-art methods.

**Index Terms**—Indoor positioning, landmarks, location-based services, pedestrian dead reckoning (PDR), smartphones, spatial information.

## I. INTRODUCTION

WHILE the global navigation satellite system (GNSS) has been successfully applied in a variety of fields, such as car navigation, geofencing, and target tracking, it cannot be used in indoor or urban environments where the GNSS signal is blocked by buildings, trees, or other obstacles. Compared with outdoor positioning, indoor localization is more challenging since indoor spaces are more complicated than outdoor environments in terms of layout, topology and space constraint [1], and indoor applications need higher accuracy [2].

A number of indoor localization systems have been proposed in recent years [3]–[5], which use different techniques, such as WiFi, ZigBee, Bluetooth, ultrawideband (UWB), radio-frequency identification (RFID), and inertial sensors. However, each of these techniques has its own drawbacks when considering accuracy, cost, coverage, complexity, and applicability. In order to achieve higher accuracy with

relatively low cost, hybrid methods combining multiple positioning techniques have been used. Common hybrid methods include multimodal fingerprinting [6], dead reckoning (DR) with WiFi fingerprinting calibration [7], and cooperative localization [8]. The problem of combining several positioning techniques is that the required infrastructure may not be available in many environments or it may be available at a high cost.

On the other hand, spatial knowledge, such as a floor plan, is available in many scenarios and can be used to assist localization without additional cost. While complex indoor spaces block many positioning signals such as WiFi, which makes localization challenging and difficult, they supply spatial constraints that are helpful for calibrating the localization error. Landmarks are one of these spatial constraints useful for indoor localization. A landmark is generally defined in the field of linguistics and cognitive science as everything that stands out of the background [9], [10]. In the context of indoor localization, a landmark refers to a location point where at least one type of sensors presents a distinctive, stable, and identifiable pattern in the readings [2], [11]. Since these location points are naturally distributed in indoor environments, one can combine them easily to bound the localization error with no extra cost.

The indoor localization systems that use landmarks have been proposed in the literature, but they are usually applied for tracking robots by using laser scanners or/and cameras [12], [13]. The systems using these devices are economically or/and computationally expensive and hence are not suitable for indoor pedestrian localization. Although landmarks based on smartphone built-in sensors are also used in some indoor localization systems [2], [14], [15], the performance of these systems relies highly on the completeness of landmarks. A mismatch of landmarks may cause a large localization error and even lead to the failure of localization.

In this article, we present *LG-Loc*, a novel and accurate landmark graph-based indoor localization method for smartphones. Compared to the existing landmark-based localization methods, which are either computation intensive or sensitive to the completeness of the landmarks, our method is computationally efficient and can handle incomplete landmarks. A landmark graph is defined as a directed graph where nodes are landmarks and edges are accessible paths with heading information. The proposed method consists of two phases: 1) offline and 2) online. In the offline phase, data from several smartphone sensors (accelerometer, gyroscope, magnetometer, barometer, light sensor, and WiFi) are collected to detect visited landmarks. Based on these landmarks and the floor maps,

Manuscript received September 23, 2019; revised March 18, 2020 and April 7, 2020; accepted April 18, 2020. Date of publication April 22, 2020; date of current version September 15, 2020. This work was supported by the National Key Research and Development Program of China under Grant 2016YFB0502200. (*Corresponding author: Jianga Shang.*)

Fuqiang Gu and Kourosh Khoshelham are with the Department of Infrastructure Engineering, University of Melbourne, Parkville, VIC 3000, Australia (e-mail: gufq87@gmail.com; k.khoshelham@unimelb.edu.au).

Shahrokh Valaee is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada (e-mail: valaee@ece.utoronto.ca).

Jianga Shang is with the School of Geography and Information Engineering and National Engineering Research Center for Geographic Information System, China University of Geosciences, Wuhan 430074, China (e-mail: jgshang@cug.edu.cn).

Rui Zhang is with the School of Computing and Information Systems, University of Melbourne, Parkville, VIC 3000, Australia (e-mail: rui.zhang@unimelb.edu.au).

Digital Object Identifier 10.1109/JIOT.2020.2989501

we are able to construct an initial landmark graph that consists of landmarks that correspond to stairs, elevators, corners, and turns. The locations of landmarks in the initial landmark graph can be obtained from the floor maps. Then, we update the landmark graph by adding more landmarks (e.g., light landmarks and WiFi landmarks) that cannot be directly obtained from the floor maps in a crowdsourcing way. In the online phase, the newly collected data are used for location initialization, location estimating, and location calibration. First, the user's initial location is inferred by a hidden Markov model (HMM)-based method. Then, her subsequent location is calculated in real time by using the pedestrian DR (PDR) method. To eliminate the accumulated error of PDR, the estimated location is regularly calibrated by matching the detected landmark with those in the landmark graph.

The main challenges of using a landmark graph for accurate indoor localization are as follows: 1) how to infer the initial location without manual input or using WiFi fingerprinting; 2) how to accurately recognize landmarks satisfying the defined three features (distinctiveness, stability, and identifiability); and 3) how to deal with a landmark association issue and missing landmarks. By solving the above challenges, we have made the following contributions to this article.

- 1) We propose a novel landmark graph-based method for indoor localization, which achieves a higher localization accuracy and is computationally efficient. The experimental results show that the proposed method performs the best with a mean error of 0.80 m compared with the map filtering methods (about 1.7 m), the WiFi fingerprinting (about 3.5 m), and the PDR (about 8). Meanwhile, our method performs about five times faster than the map filtering method with 200 particles that can achieve a relatively high accuracy compared to the PDR and the WiFi fingerprinting.
- 2) We propose different rules for detecting varying landmarks and design a metric called belief to deal with the problems of multiple nearby landmarks and missed landmarks. The existing research considers only an isolated landmark for calibrating the accumulated error of PDR, which does not work well when some landmarks are missed or there are multiple landmarks nearby. These problems are addressed by a landmark matching method based on a landmark graph.
- 3) We propose a novel initial location inference method by using a landmark graph-based HMM. The existing systems either use WiFi fingerprinting to determine the initial location of a user or ask the user to input an initial location manually, which are labor intensive or not user friendly. The proposed HMM-based location inference method can determine the initial location of a user without requiring the user's active participation or a WiFi fingerprint database.

## II. RELATED WORK

In this section, we provide a brief overview of related work on indoor localization and spatial information-aided localization.

### A. Indoor Localization

WiFi-based indoor localization is a popular method since it can make use of the existing WiFi infrastructures and thereby reduce the cost of deploying hardware required for localization. Early WiFi-based systems usually convert the received signal strength (RSS) to a distance using a certain signal propagation model such as a log-distance path-loss model, which is then used to calculate the location of a user through triangulation. However, these systems usually suffer from the low-accuracy problem due mainly to the multipath effect of signals in complex indoor environments. To improve the localization accuracy, some recent works have utilized the channel state information (CSI) [16], [17] to drive angle-of-arrival (AOA) or time-of-flight (TOF) information, which is used for locating a user via triangulation. The triangulation methods based on CSI can achieve centimeter-level accuracy. Nevertheless, triangulation methods, regardless of using RSS or CSI, require to know the locations of APs, which are difficult to obtain in many public spaces, limiting their applicability. To relieve the requirement for APs' location knowledge, fingerprinting methods have been proposed [18]–[20]. The fingerprinting methods are also based on RSS or CSI. The main challenge of fingerprinting is that it requires a time-consuming and labor-intensive site survey process. Some solutions have been proposed to reduce the fingerprint collection and calibration effort, such as crowdsourcing with active user participation [21], building a signal propagation model [22], or certain calibration fingerprints [23]. Compared to RSS-based localization for both triangulation and fingerprinting, CSI-based localization achieves much higher localization accuracy, but it has poorer coverage and the extraction of CSI is not supported by the current smartphones, limiting the applicability of CSI-based localization.

DR is another commonly used indoor localization method, which uses inertial sensor readings to estimate the real-time location of a target, given an initial location [24]. The advent of sensor-rich smart devices has enabled DR to be widely used, which is especially useful for localization and tracking in the wireless signal-denied areas. The advantage of DR is that it does not require extra infrastructure and has no coverage limitation. However, DR suffers from the accumulated error problem that the localization accuracy decreases over time, leading to the long-term DR practically being useless. A commonly used solution is to integrate DR with other positioning techniques, such as WiFi [25], UWB [26], and vision [27], which can eliminate both the accumulated location error of DR and the jumping estimations by absolute positioning techniques for a short time. However, these absolute localization techniques are not always available and often need to spend extra cost on deployment and maintenance.

Apart from WiFi-based methods and DR, there are also other indoor localization methods based on light, RFID, magnetism, Bluetooth, Infrared, sound, ZigBee, etc. In recent years, visible light-based localization has attracted the attention of many researchers [28]–[30]. It can achieve decimeter-level and even a centimeter-level accuracy and has the potential to be applied in robot navigation, unmanned aerial vehicle (UAV) navigation, etc. However, unlike WiFi-based

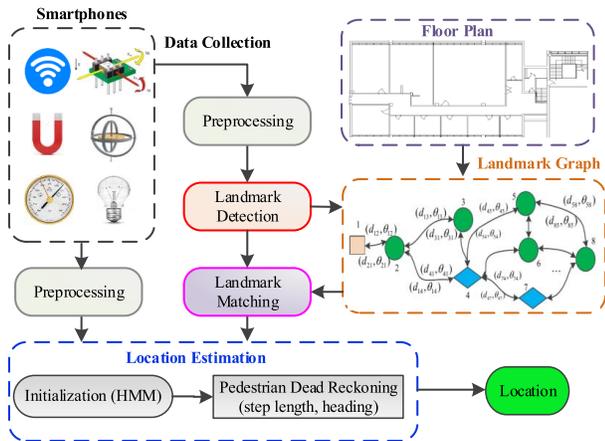


Fig. 1. System architecture of *LG-Loc*.

localization, which can make use of the existing wireless access points, visible light-based localization requires modifying the existing lighting infrastructure for the purpose of localization. Also, there is no uniform international standard regarding visible lights for the purpose of indoor localization.

### B. Spatial Information-Aided Localization

Spatial Knowledge can be used for assisting indoor localization at no extra cost for deploying hardware. There are two main methods of fusing spatial information to improve location accuracy, namely, map matching and spatial model-aided method [3]. Landmark matching [2] is a popular map matching method because of its simplicity and high operation efficiency. However, it is sensitive to the completeness of landmark detection, and inaccurate matching may lead to a larger localization error. Other map matching methods, such as trajectory matching [31] and the Bayesian approaches [32], [33] can usually achieve higher accuracy, but they are computationally expensive, making them impractical for applications running on resource-limited mobile devices such as smartphones.

Another method of fusing spatial information to enhance localization accuracy is using spatial models. Compared with basic indoor maps, indoor spatial models include richer information, not only static objects (e.g., rooms, doors, sensors, and furniture) and dynamic objects (e.g., people), but also their spatial relationships. With more geometric, topological, and semantic information, spatial models can be used to significantly improve location accuracy as well as to achieve more reliable location-based services [34], [35]. However, automated methods for reconstructing indoor spatial models are in their infancy and manual methods are labor intensive and slow [36].

## III. SYSTEM OVERVIEW

The system architecture of *LG-Loc* is shown graphically in Fig. 1. When a user enters a building, she can launch the localization application to obtain her location within the building in real time. The application first requests the landmark graph of this building, which is constructed and stored in the server in the training phase. Then, it starts collecting sensor

readings from the WiFi module, accelerometer, magnetometer, gyroscope, barometer, and light sensors together with the corresponding timestamps. These data are first preprocessed through a low-pass filter and subsequently used to detect landmarks and estimate the location. The initial location of the user can be obtained by using the WiFi fingerprinting method (if the corresponding WiFi fingerprint database is available) or inferred via utilizing an HMM-based algorithm according to the landmark graph and sensor readings (if the WiFi fingerprint database is not available).

In the following, we will elaborate on each of the three key modules: 1) landmark graph; 2) landmark matching; and 3) location estimation with landmark graph constraints.

## IV. LANDMARK GRAPH

An initial concept of the landmark graph has been presented in our previous work [37]. In this article, we extend the landmark graph by introducing more types of landmarks, and an automated update of the landmark graph. In the following, we first give the definition of landmarks and their recognition rules. Then, we show the construction of an initial landmark graph from floor maps and the updating of the landmark graph.

### A. Definition and Recognition of Landmarks

Landmarks in this article are sensory landmarks [11], which are defined as location points where sensor readings present a distinct, stable, and identifiable change pattern. For instance, the gyroscope presents a distinct change when the user takes a turn. A landmark has three features: 1) distinctiveness; 2) stability; and 3) identifiability. The distinctiveness means that the change pattern of sensor readings for a landmark must be distinctive, which stipulates that this landmark can be distinguishable from its surroundings. The stability means that a landmark must be stable for a period of time, which means that it does not change dynamically over time. The identifiability means that a landmark must be detectable by one or more types of sensors at the location point. Mathematically, we define a landmark  $v$  as follows:

$$v = \langle (x, y), (\mathcal{R}_1, \dots, \mathcal{R}_M) \rangle \quad (1)$$

where  $(x, y)$  denotes the coordinate of the landmark,  $(\mathcal{R}_1, \dots, \mathcal{R}_M)$  represents the detection rule in different types of sensor readings, and  $M$  is the number of rules that this landmark possesses. A landmark may satisfy one or more types of detection rules. For example, opening a door may impose a change pattern on both the accelerometer and gyroscope readings.

The landmarks can be categorized by the type of sensor used to detect them. Modern smartphones have integrated various sensors, including accelerometer, gyroscope, barometer, WiFi, and light sensor. Accordingly, we can categorize landmarks into different types as follows.

**Accelerometer Landmark:** The motion state of a user changes at certain locations in an indoor environment, which can be sensed by the accelerometer. For instance, when a user opens a door, his or her motion state would change from walking to still, and then to walking. The location of the door can

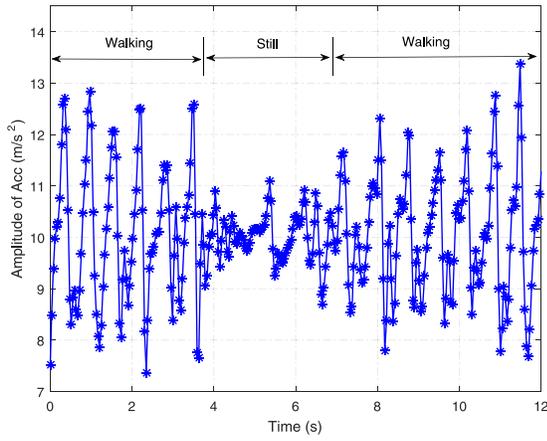


Fig. 2. Change in the amplitude of acceleration when a user passes through a door.

be regarded as an accelerometer landmark if the user has to experience this change every time he or she passes through the door. Fig. 2 shows the change in the amplitude of acceleration when a user passes through a door. The assignment of motion states (e.g., walking and still) to the accelerometer data is based on a supervised classification method developed in our previous work [38]. The change pattern of “Walking  $\rightarrow$  Still (for a few seconds)  $\rightarrow$  Walking” can be regarded as a rule that checks whether a door is a potential accelerometer landmark. This rule can also be used to detect other potential landmarks, such as an indoor water fountain point and other similar activity-related location points.

In our application, the rule  $\mathcal{R}_{\text{acc}}$  of accelerometer landmarks is defined as

$$\begin{aligned} \mathcal{R}_{\text{acc}} = & (\text{loc}_t | m_{t-K_1} : t == \text{walking} \\ & \& m_{t:t+K_2} == \text{still} \\ & \& m_{t+K_2} : t+K_2+K_1 == \text{walking}) \quad (2) \end{aligned}$$

where  $m_t$  represents the user’s motion state (e.g., walking and still) at time  $t$ , and  $K_1$  and  $K_2$  are the two window sizes of accelerometer readings that determine the period of detecting the corresponding motion state. When the user’s motion state follows the pattern of “Walking  $\rightarrow$  Still (for a few seconds)  $\rightarrow$  Walking,” the rule  $\mathcal{R}_{\text{acc}}$  is satisfied, representing that the location point at time  $t$  is a potential accelerometer landmark.

*Gyroscope Landmark:* The walking direction of a user changes when he/she takes a turn or passes a corner. This change can be detected by using the magnetometer and the gyroscope. However, the magnetometer readings tend to be easily affected by ferromagnetic materials, leading to their unsuitability for the detection of turns and corners. Therefore, we use the gyroscope readings to detect turns and corners since they are not influenced by ferromagnetic materials. Fig. 3 shows how the gyroscope readings change when the user takes a left or right turn. To detect a gyroscope landmark, we define the rule  $\mathcal{R}_{\text{gyro}}$  of a gyroscope landmark as follows:

$$\mathcal{R}_{\text{gyro}} = (\text{loc}_t | |\dot{\theta}_t| > \epsilon_{\text{gyro}}) \quad (3)$$

where  $\dot{\theta}_t$  is the gyroscope readings along the vertical direction. If the absolute value of  $\dot{\theta}_t$  is greater than a certain threshold

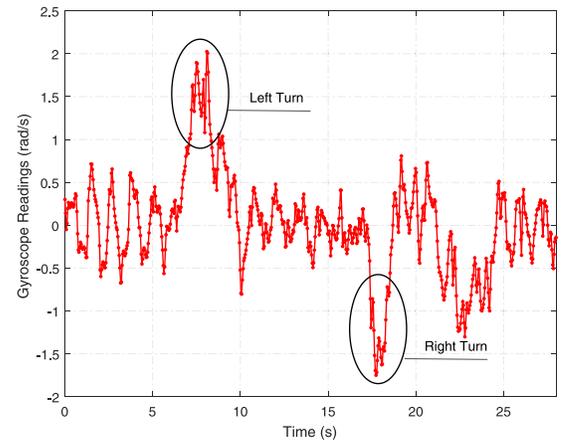


Fig. 3. Change in the gyroscope readings on the z-axis when a user takes a turn. (The user holds the phone in hand.)

$\epsilon_{\text{gyro}}$ , we consider this location point as a potential gyroscope landmark.

*Barometer Landmark:* The barometer is able to measure the air pressure, which changes with the altitude or height. This means that it can be used to detect the vertical movement of a user (e.g., going upstairs or downstairs and taking an elevator). Fig. 4 shows the change in the barometer readings when a user walks horizontally, goes upstairs or downstairs, and takes an elevator upward. The entrance and exit of stairs and elevators can be regarded as barometer landmarks since they are identifiable, distinctive, and stable. The entrance detection is done by detecting the change pattern “horizontal movement  $\rightarrow$  vertical movement.” Similarly, the exit is detected by using the pattern “vertical movement  $\rightarrow$  horizontal movement.” Both change patterns are recognized by utilizing the barometer readings. Let  $p_i$  denote the average value of the  $i$ th window of air pressure readings that contains the air pressure value at time  $t$ , and let  $\epsilon_{\text{baro}_1}$  and  $\epsilon_{\text{baro}_2}$  be the thresholds used to detect the user’s horizontal movement and vertical movement, respectively. The rule to detect the entrance to a set of staircases or an elevator is defined as

$$\begin{aligned} \mathcal{R}_{\text{baro}_1} = & \left( \text{loc}_t | (|p_i - p_{i-1}|) < \epsilon_{\text{baro}_1} \right. \\ & \& \left. \left| \sum_{j=i+1}^{i+K_{p_1}} (\text{sgn}(p_j - p_{j-1})) \right| == K_{p_1} \right. \\ & \& \left. |p_{i+K_{p_1}} - p_i| > \epsilon_{\text{baro}_2} \right). \quad (4) \end{aligned}$$

The first term is for detecting the horizontal movement, and the latter two terms are for detecting the vertical movement.  $\text{sgn}$  is the sign function, which is described as

$$\text{sgn}(p_j - p_{j-1}) = \begin{cases} 1, & \text{if } p_j > p_{j-1} \\ 0, & \text{if } p_j = p_{j-1} \\ -1, & \text{if } p_j < p_{j-1}. \end{cases} \quad (5)$$

Similarly, we can define the rule to detect the exit from a set of staircases or an elevator as

$$\mathcal{R}_{\text{baro}_2} = (\text{loc}_t | (|p_i - p_{i+1}|) < \epsilon_{\text{baro}_1})$$

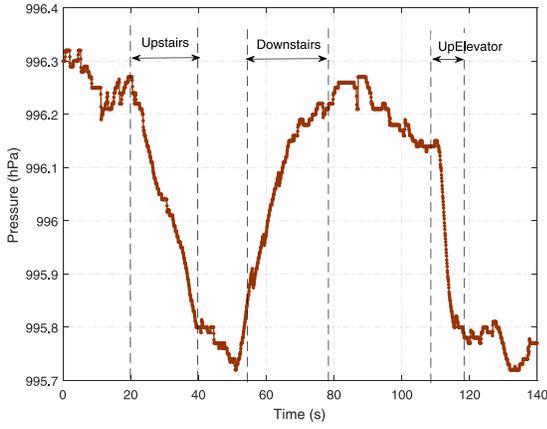


Fig. 4. Change in the pressure when a user takes stairs or an elevator.

$$\begin{aligned} \&\& \left| \sum_{j=i-K_{p_2}+1}^i (\text{sgn}(p_j - p_{j-1})) \right| == K_{p_2} \\ \&\& |p_{i-K_{p_2}} - p_i| > \epsilon_{\text{baro}_2}. \end{aligned} \quad (6)$$

The values of  $K_{p_1}$  and  $K_{p_2}$  are not constant, but are determined dynamically. Their initial values are set to 1, and gradually increase as long as the value of the sign function keeps unchanged.

**WiFi Landmark:** We define a location point that overhears the strongest RSS from an AP within a certain region as a WiFi landmark. There is usually only a small area in which the RSS from a specific AP is the strongest. Such a point is usually stable, distinctive, and identifiable. Fig. 5 denotes the RSS from an AP while the user was walking in a corridor. The location point at the 65th second is a WiFi landmark since it receives the strongest RSS from the corresponding AP compared to other location points.

Let  $\text{RSS}_t$  denote the RSS from an AP at a location point  $\text{loc}_t$ . Then, we can define the rule  $\mathcal{R}_{\text{WiFi}}$  of a WiFi landmark as

$$\begin{aligned} \mathcal{R}_{\text{WiFi}} = (\text{loc}_t | (\text{RSS}_t == \max(\text{RSS}_{t-K_{\text{WiFi}}:t+K_{\text{WiFi}}})) \\ \&\& \text{RSS}_t > \epsilon_{\text{WiFi}}) \end{aligned} \quad (7)$$

where  $\epsilon_{\text{WiFi}}$  is a threshold to exclude the local maxima. We do not consider a location point that witnesses a local maximum of RSS (e.g., the one arising at the 20th second in Fig. 5) as a landmark since it is not stable and cannot be detected accurately.

**Light Landmark:** The light sensor built in a smartphone can be used to measure the light intensity of the environment that the phone’s screen faces. It can be used to detect the projection location of a lamp installed on the ceiling, which can be regarded as a landmark. As shown in Fig. 6, the light sensor in the smartphone presents a peak of illuminance when the user passes below a ceiling lamp. We define the rule  $\mathcal{R}_{\text{light}}$  of a light landmark as

$$\begin{aligned} \mathcal{R}_{\text{light}} = (\text{loc}_t | (LX_t == \max(LX_{t-K_{\text{Light}}:t+K_{\text{Light}}})) \\ \&\& LX_t > \epsilon_{\text{Light}}) \end{aligned} \quad (8)$$

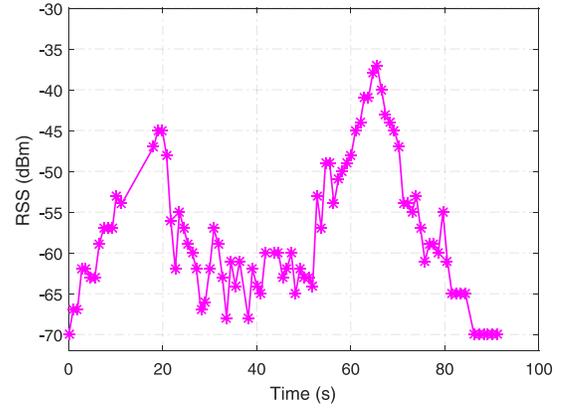


Fig. 5. RSS from an AP at different locations.

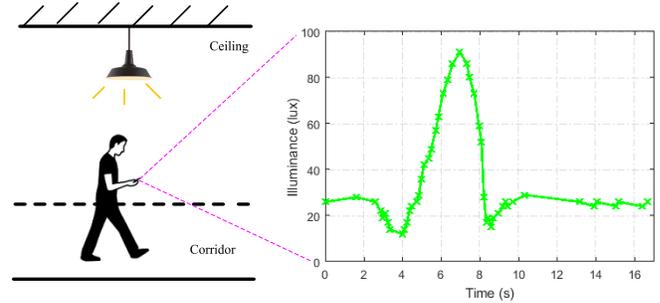


Fig. 6. Change of luminance when a user passes a ceiling lamp.

where  $LX_t$  represents the illuminance sensed by the built-in smartphone light sensor at the location  $\text{loc}_t$ , and  $K_{\text{Light}}$  is a threshold that controls the range of light sensor readings used to extract a peak.  $\epsilon_{\text{Light}}$  is a threshold to exclude weak peaks of illuminance.

### B. Construction of Initial Landmark Graph

The locations of some landmarks correspond to the locations of doors, elevators, stairs, corners, and turns that can be obtained from floor maps, which we assume are available. Other landmarks, such as WiFi landmarks and light landmarks can be gradually learned by combining sensor readings with users’ trajectories, which can be used to assist subsequent localization.

We can construct the corresponding landmark graph using map information and outputs from the landmark detection component. A landmark graph consists of nodes (landmarks) and edges (accessible paths). Let  $G = (V, E)$  denote a landmark graph, where  $V = \{v_1, \dots, v_N\}$  is a set of landmarks and  $E = \{e_1, \dots, e_M\}$  is the set of edges in graph  $G$ . Each edge  $e_i = \langle v_j, v_k, \theta_{jk}, d_i \rangle$  is a tuple consisting of two landmarks, the direction from one landmark to the other, and the distance between the two landmarks.

### C. Updating of Landmark Graph

The landmark graph is initially constructed according to a floor plan, which is incomplete since the floor plan does not give the room-level details, such as the layout of furniture or office desks. We also cannot obtain the locations of WiFi

**Algorithm 1: Update the Landmark Graph**


---

**Input** : An initial landmark graph  $G$ ,  $N$  trajectories, each contains  $n_i$  potential landmarks denoted by  $\{v_1^i, v_2^i, \dots, v_{n_i}^i\}$

**Output**: A landmark graph  $G$  with newly added landmarks

- 1 Assign the potential landmarks into  $N_0$  initial landmark clusters denoted by  $\{c_1, \dots, c_{N_0}\}$ , each cluster contains one potential landmark
- 2 // *Distance constraint-based clustering*:
- 3 **repeat**
- 4     $\forall i, j$ , **if**  $\text{dist}(c_i, c_j) \leq d_c \&\& \text{rule}(c_i) == \text{rule}(c_j)$  **then**
- 5        Merge the two clusters into one
- 6    **end**
- 7 **until**  $\text{dist}(c_i, c_j) > d_c \quad \forall i, j$ ;
- 8 // *Delete the clusters with too few elements*:
- 9 **foreach**  $c_i \neq \text{null}$  **do**
- 10    **if**  $\text{size}(c_i) < \epsilon_c$  **then**
- 11        delete  $c_i$
- 12    **end**
- 13 **end**
- 14 Compute the size of remaining clusters, represented by  $N_c$
- 15 // *Update the landmark graph*:
- 16 **for**  $i++ \leq N_c$  **do**
- 17    Add a new node  $v_i$  into  $G$  with the center of  $c_i$  as the coordinate, and the detection rule as a property of  $v_i$
- 18    Compute the heading and distance from node  $v_i$  to each of its neighbouring nodes that are already in  $G$
- 19    Add all relative edges  $e_j = \langle v_i, v_k, \theta_{ik}, d_j \rangle$  into  $G$
- 20 **end**

---

landmarks and light landmarks from a floor plan. However, these landmarks that cannot be derived from a floor plan can be learned by crowdsourcing.

Suppose we have collected  $N$  trajectories of the user, each trajectory contains  $n_i$  potential landmarks denoted by  $\{v_1^i, v_2^i, \dots, v_{n_i}^i\}$ . Let  $N_0$  denote the total number of potential landmarks from the  $N$  trajectories, namely,  $N_0 = \sum_{i=1}^N n_i$ . These potential landmarks are the location points that satisfy one type of landmark detection rules but are not yet included in the current landmark graph. The updating procedure of the landmark graph is described in Algorithm 1, where  $\text{dist}(c_i, c_j)$  is a function used to compute the Euclidean distance between the center's coordinate of two landmark clusters  $c_i$  and  $c_j$ , and  $\text{rule}$  is a function used to get the detection rule of the corresponding landmark.  $d_c$  is a distance threshold that indicates whether or not two clusters can be merged, and  $N_c$  is a threshold reflecting the stability of a landmark.

## V. LANDMARK MATCHING

A key challenge to using landmarks for assisting localization is to solve the data association issue [13], [39], [40]. In other words, when there are multiple landmarks nearby, it is difficult to determine the detected landmark. An additional challenge is to deal with the case that one or more landmarks are missing.

To solve the above problems, we define a belief  $\text{bel}$  to indicate how much we can trust the location point that meets the detection rule is a landmark. The belief that the location point  $(x_t, y_t)$  is matched with the landmark  $v_k$  in the landmark graph is expressed as

$$\text{bel}(v_k) = \delta(R_k, R_t) \cdot r(\theta_k, \theta_t) \cdot g(d_k, d_t) \quad (9)$$

where  $k$  is the index of landmark in the landmark graph.  $R_k$  is the detection rule of the reference landmark  $v_k$ , and  $R_t$  is

the type of the detected landmark at time  $t$ .  $\theta_k$  and  $\theta_t$  are the reference heading and the estimated heading from the time of visiting the last landmark to time  $t$ .  $d_k$  and  $d_t$  are the reference distance and the traveled distance from the last landmark to a location  $(x_t, y_t)$ .  $\delta$  is the Dirac delta function, which is defined as

$$\delta(R_k, R_t) = \begin{cases} 1, & \text{if } R_k == R_t \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

and  $r$  is the rectangle function, described as

$$r(\theta_k, \theta_t) = \begin{cases} 1, & \text{if } |\theta_k - \theta_t| < \epsilon_{\text{theta}} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where  $\epsilon_{\text{theta}}$  is a heading threshold. The function  $g$  is defined as

$$g(d_k, d_t) = e^{-|d_k - d_t|}. \quad (12)$$

If there are multiple possible landmarks nearby, the one with the largest value of  $\text{bel}$  will be chosen as the detected landmark. To reduce the risk of mismatching, we set a belief threshold  $\epsilon_{\text{bel}}$  to exclude fake landmarks. For instance, a user may take a turn in the middle of a corridor, which may be mistakenly detected as a gyroscope landmark. However, since fake landmarks have a lower belief, we are able to exclude them by judging whether their belief is smaller than the threshold  $\epsilon_{\text{bel}}$ . We use only the landmarks with belief larger than  $\epsilon_{\text{bel}}$  to calibrate the accumulative error of PDR.

In some cases, some landmarks may be missing. For example, certain landmarks at the locations of doors will be missed if a door is left open since the user does not stop to open the door (no “Walking  $\rightarrow$  Still  $\rightarrow$  Walking” pattern). The lamps might be on or off at different times of the day, which will lead to failure of the detection of the corresponding light landmark. In these cases, we simply ignore them and do not correct the user's location until the next landmark is detected.

## VI. LOCATION ESTIMATION

In this section, we introduce the proposed location estimation method using a landmark graph. It includes location initialization and PDR-based location estimation with landmark graph constraints.

### A. Location Initialization

The initial location of an indoor localization system can be obtained in a few ways. A popular way is to use the WiFi fingerprinting method [41]. This is done by matching the newly collected WiFi fingerprint with the fingerprints that are collected and stored in a database in the training phase. The drawback of obtaining the initial location by WiFi fingerprinting is that it requires an offline training process, which is time consuming and labor intensive.

Alternatively, the user can input an initial location when launching the localization app. This way is adopted by many systems since it does not require much effort to deploy or train the systems. However, it needs active user participation, which may not be user friendly and is even difficult if the user is not familiar with the environment.

In this article, we present a novel way to infer the user’s initial location which uses a landmark graph-based HMM [42]. Initially, the app has no location information of the user, and thus the user is required to walk for a while, allowing the app to collect enough sensor readings in order to estimate her location. Then, these sensor readings are fed to the landmark recognition module and the PDR module to generate a sequence of observations that consist of detected landmarks (without location knowledge), distances, and directions between these landmarks. Let  $S = \{v_1, \dots, v_N\}$  denote a set of  $N$  hidden states (which are landmarks),  $O = \{o_1, \dots, o_M\}$  represent a sequence of  $M$  observations and  $o_k = (R_k, d_k, \theta_k)$ . Thus, the initial location inference can be modeled as a decoding problem. Namely, given the observation sequence  $O$  and a model  $\lambda = (A, B, \pi)$ , choose a corresponding landmark sequence  $Q = \{v_1 v_2 \dots v_M\}$  which best explains the observation sequence  $O$ . In the model  $\lambda$ ,  $A$  and  $B$  are the state transition matrix and emission matrix, respectively, and  $\pi$  is the initial state distribution. Before the model can be exploited to infer location, we need to compute the transition and emission matrices.

*Transition Probabilities:* The values of the transition matrix are obtained based on the landmark graph. In principle, the element  $a_{ij}$  of the transition matrix  $A$  must meet the following properties:

$$a_{ij} = P(s_t = v_j | s_{t-1} = v_i) \geq 0 \quad (13)$$

$$\sum_{j=1}^N a_{ij} = 1 \quad (14)$$

where  $s_t$  represents the landmark visited at time  $t$ . We define the transition probability between two landmarks according to whether there is a direct connection between them in the landmark graph. A higher probability  $p_h$  is assigned to the landmark pair that has a direct connection and a lower probability  $p_l$  is assigned to those without a direct connection. We adopt the approach of probability assignment in [30], and the values of high and low transition probabilities are computed as follows:

$$p_h = K \cdot p_l \quad (15)$$

$$\sum_{j=1}^N a_{ij} = I \cdot p_h + (N - I) \cdot p_l = 1 \quad (16)$$

where  $K$  is the ratio between the higher probability and the lower probability, and  $I$  is the number of state pairs with higher transition probabilities.

*Emission Probabilities:* The emission probability reflects the concept that each state emits an observation with a particular conditional probability distribution. Given the state  $s_i$ , the probability  $b_{ik}$  can be computed using the following formula:

$$\begin{aligned} b_{ik} &= P(o_t = o_k | s_t = s_i) \\ &= \max_{1 \leq j \leq \text{deg}^+(s_i)} [\delta(R_j, R_k^*) \cdot r((\theta_j, \theta_k^*)) \cdot g(\text{dis}(d_j, d_k^*))] \end{aligned} \quad (17)$$

where  $\delta$ ,  $r$ , and  $g$  are the same functions as in (9), and  $\text{deg}^+(s_i)$  is the outdegree of  $s_i$ .  $R_k^*$  is the detection rule of the collected

---

### Algorithm 2: HMM-Based Location Inference

---

**Input** : A landmark graph  $G$ , sensor readings from accelerometer, gyroscope, magnetometer, barometer, WiFi, light sensor  
**Output**: Initial location and current location

- 1 Extract the sequence of observations  $o_1, o_2, \dots, o_M$  based on collected sensor readings
- 2 Calculate the transition matrix  $A = a_{ij}$  and the emission matrix  $B = b_{ik}$  according to the landmark graph and the observation sequence
- 3 // Initialization:
- 4 **for**  $i = 1 \rightarrow N$  **do**
- 5      $\phi_1(i) = \pi_i \cdot b_i(o_1)$
- 6      $\Psi_1(i) = 0$
- 7 **end**
- 8 // Recursion:
- 9 **for**  $k = 2 \rightarrow M$  **do**
- 10     **for**  $j = 1 \rightarrow N$  **do**
- 11          $\phi_{k+1}(j) = \max_{1 \leq i \leq N} [\phi_k(i) \cdot a_{ij}] \cdot b_j(o_{k+1})$
- 12          $\Psi_{k+1}(j) = \text{argmax}_{1 \leq i \leq N} [\phi_k(i) \cdot a_{ij}]$
- 13     **end**
- 14      $P^* = \max_{1 \leq i \leq N} [\phi_k(i)]$
- 15     **if**  $P^* > \epsilon_{\text{termination}}$  **then**
- 16         // location determined
- 17          $v_M^* = \text{argmax}_{1 \leq i \leq N} [\phi_k(i)]$
- 18         **break**;
- 19     **end**
- 20 **end**
- 21 // Path backtracking:
- 22 **for**  $t = k - 1 \rightarrow 1$  **do**
- 23      $v_t^* = \Psi_{t+1}(v_{t+1}^*)$
- 24 **end**
- 25 // Location Inference:
- 26 Compute the initial location  $(x_0, y_0)$  and current location  $(x_k, y_k)$  according to the first visited landmark  $v_1^*$  and the last visited landmark  $v_k^*$

---

observation  $o_k$ , and  $R_j$  is the detection rule of  $s_j$  connected from  $s_i$  by an edge in the landmark graph.  $\theta_k^*$  is the observed heading and  $\theta_j$  is the heading of the edge connecting  $s_i$  to  $s_j$ .  $d_k^*$  is the observed distance and  $d_j$  is the traveled distance from last  $s_i$  to  $s_j$ . Then, the emission matrix  $B$  is normalized.

We use the Viterbi algorithm to initialize the localization system. The objective is to find a sequence of landmarks  $Q = \{v_1 v_2 \dots v_M\}$  that best explains the given observation sequence  $O = \{o_1, \dots, o_M\}$ . We define the Viterbi variable as

$$\phi_k(i) = \max_{v_1, v_2, \dots, v_{k-1}} P[v_1 v_2 \dots v_k = i, o_1 o_2 \dots o_k | \lambda]. \quad (18)$$

It represents the highest probability along a single landmark path at time  $k$ , which best explains the given observations. By induction, we can further obtain

$$\phi_{k+1}(j) = \left[ \max_i \phi_k(i) a_{ij} \right] \cdot b_j(o_{k+1}). \quad (19)$$

To infer the initial location and current location of a user, we require to keep track of the argument that maximizes the Viterbi variable for each  $t$  and  $j$ . The complete algorithm of determining the initial location and current location is described in Algorithm 2.

In the above algorithm, the initial state distribution  $\pi$  is set according to the type of the first detected landmark. Namely, we set  $\pi_i = 1/N_0$ ,  $1 \leq i \leq N_0$ , where  $N_0$  is the number of landmarks having the same detection rule as the first observation. Then, according to subsequent observations, we can use

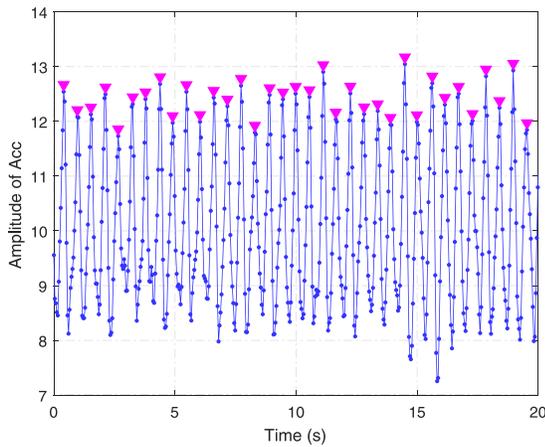


Fig. 7. Peaks of the acceleration when a user walks.

the above algorithm to initialize the localization system. The initial location is determined when the maximum of the Viterbi variable  $\phi_k$  is larger than a threshold  $\epsilon_{\text{termination}}$ . This means that the path becomes unique given  $k$  observation sequences.

### B. PDR-Based Location Estimation With Landmark Graph Constraints

The PDR method consists of step event detection, step length estimation, and heading estimation. We first introduce the three steps of PDR and then present how a landmark graph can be used to calibrate the accumulative error of PDR.

*Step Event Detection:* The step event can be captured by using the smartphone built-in accelerometer because its readings present a periodical and repetitive pattern when the user is walking [43], [44]. The number of steps that the user takes corresponds to the number of peaks in the amplitude of accelerometer readings. For instance, Fig. 7 shows the change in the amplitude of acceleration when a user walks normally, in which magenta triangles represent the peaks of acceleration and each peak corresponds to one step. By detecting and counting these peaks, we are able to compute the number of steps and to further calculate the step length. To make the peak-based detection robust to phone poses, we use the amplitude of the acceleration

$$\text{acc}_t = \sqrt{\text{acc}_{x_t}^2 + \text{acc}_{y_t}^2 + \text{acc}_{z_t}^2} \quad (20)$$

where  $\text{acc}_{x_t}$ ,  $\text{acc}_{y_t}$ , and  $\text{acc}_{z_t}$  are the accelerometer readings along the  $x$ -axis,  $y$ -axis, and  $z$ -axis of the phone at time  $t$ , respectively. To avoid the effect of false walking and noise caused by changing phone carrying ways, we adopt the periodicity and similarity constraints strategy in [43]. The periodicity is the difference in the timestamp of two neighboring accelerometer readings. The periodicity constraint is used to limit the range of walking periodicity to a certain interval (e.g., [0.4, 1]) so as to overcome the overcounting problem to some extent. Another useful constraint is the similarity constraint, which is based on the observation that the acceleration peaks for two steps are close when users walk naturally. Peaks that exceed a similarity threshold are identified as false peaks.

Only an acceleration peak that satisfies both the periodicity and similarity constraints is counted as one step.

*Step Length Estimation:* Different models, such as the Weinberg model [45], the Kim model [46], the linear model [47], and the deep learning model [48], can be used to estimate a user's step length, but these models either are sensitive to the amplitude of acceleration, require the knowledge of the user's stature or a large amount of data to train. In this article, we combine the step counting-based method with the landmark graph to compute the step length of the user. Initially, the step length is set to a constant  $l_0$ , which will be updated once the user passes two neighboring landmarks. Let  $v_1$  and  $v_2$  indicate the two neighboring landmarks that a user passes subsequently, and  $N_s$  be the number of steps that a user takes to travel from  $v_1$  to  $v_2$ . Then, we can calculate  $l$  as follows:

$$l = \frac{\text{dist}(v_1, v_2)}{N_s} \quad (21)$$

where  $\text{dist}(v_1, v_2)$  is the Euclidean distance between  $v_1$  and  $v_2$ . Note that the trajectory of the user traveling from  $v_1$  to  $v_2$  should be a straight line, otherwise we keep the last estimated step length. The advantage of our step length estimation method is that it does not require the user's stature information and can adapt to varying walking speeds since it keeps updated as the user passes two neighboring landmarks that are on a straight line.

*Heading Estimation Considering Constraints:* Both the compass and the gyroscope built in a smartphone can be used to compute the heading of the smartphone user. However, the compass is vulnerable to ferromagnetic materials while the gyroscope suffers from the drift problem. A solution to this is combining the compass readings and gyroscope readings by using the Kalman filter or other similar techniques [14]. However, the readings in the two sensors still fluctuate even when the user walks along a straight line.

In fact, we observe that it is not necessary to get the exact movement direction since the movement of a user in an indoor environment is constrained by the geometry of the environment. For example, a user can only move in two directions in a corridor. Constraining users' motion direction can help achieve higher location accuracy since the constrained direction is more robust than specific heading readings provided by the compass or gyroscope. In this article, we constrain the user's walking direction when she walks in a corridor or corridor-like environment by using the landmark graph where the direction between two landmarks is constrained and pre-set. The compass and gyroscope are only used to provide a coarse estimation to compute the heading sector. When the user walks in a room where there are no landmarks, we use the gyroscope readings to estimate the heading.

The complete proposed method is described in Algorithm 3. It starts from the location initialization module, which uses an HMM-based method to initialize the location. Then, the acceleration-based peak detection is used to partition the sensor readings into segments. When there is not a salient peak, a fixed window size (e.g., 50 Acc samples) is used to partition sensor data. A segment starts at a time moment witnessing an

acceleration peak and ends at the next time moment appearing another acceleration peak. Each segment corresponds to the data collected during the period of the user taking a step unless the acceleration variance of this segment is smaller than an acceleration threshold (which is 0.5 in this article) that is used to distinguish walking state from still state. After detecting a step event, we conduct the standard PDR method under the landmark graph constraints on the step length and heading. Let  $(x_k, y_k)$  denote the location of a user at step  $k$ ,  $l_k$  be the corresponding step length, and  $\theta_k$  represent the heading, then the location estimation using the PDR method can be written as

$$x_k = x_{k-1} + l_k \sin(\theta_k) \quad (22)$$

$$y_k = y_{k-1} + l_k \cos(\theta_k) \quad (23)$$

$$\theta_k = f_\theta(\theta_k^*, \theta_k^r) \quad (24)$$

where  $f_\theta$  is the function that determines the heading at step  $k$  according to the compass reading  $\theta_k^*$ , and the heading information  $\theta_k^r$  in the landmark graph. The step length  $l_k$  can be obtained by using (21). After the location estimation using the PDR for each step, sensor readings of each type starting from the time that witnesses a landmark to the current time that the user takes a step are fed to the landmark detection module to detect whether a potential landmark is encountered. A potential landmark is confirmed by the landmark matching module by comparing its belief with the belief threshold. If a landmark is detected and confirmed, its location will be used to correct the user's location and to update her step length.

## VII. EXPERIMENTS AND RESULTS

The proposed indoor localization solution was evaluated by experiments conducted in an office building. The office building consists of eight floors, which is a typical office environment, including elevators, staircases, corridors, common rooms, and office rooms. The testing path goes through two floors of this building, and its length is about 362 m.

The device we used is a Google Nexus 6 smartphone equipped with WiFi, accelerometer, magnetometer, gyroscope, barometer, and light sensor. Six volunteers took part in the experiments. The participants walked along the preset path with the phone in hand and reported the preset markers they encountered to evaluate the location accuracy. Note that the effect of the different phone carrying ways on the localization performance has been analyzed in our previous work [14] and many other works [49], [50]. The data recorded include the media access control (MAC) address of visible wireless access points and corresponding RSS, and readings from the accelerometer, gyroscope, compass, barometer, and light sensor. All these data were recorded with their corresponding timestamps so that we could align data from different sensors to jointly infer ground-truth location points.

To obtain the ground truth, we set 126 markers along the testing path with an interval of 3 m. When the participants passed a marker, they were required to click this marker on an Android app that we developed for collecting sensor data and recording ground-truth locations.

### Algorithm 3: Landmark Graph-Based Location Estimation

```

Input : A landmark graph  $G$ , sensor readings from accelerometer ( $acc$ ), gyroscope ( $gyro$ ), compass ( $\theta^*$ ), barometer ( $baro$ ), WiFi ( $wifi$ ), light sensor ( $light$ )
Output: Real-time location estimation
1 Location initialization using the proposed HMM-based method to obtain  $(x_0, y_0)$ 
2 while new sensor readings are available do
3    $[acc_{t-1:t}] = \text{peakDetect}(acc)$ ; //Conduct the acceleration -based peak detection, and return the accelerometer readings between the last peak  $pk_{t-1}$  and current peak  $pk_t$ 
4   if  $\text{var}([acc_{t-1:t}]) > 0.5$  then
5     /* A step happens */
6      $l_k = l_{k-1}$ ; //The initial step length  $l_0$  is known
7      $\theta_k = \text{estimateHeading}(G, \theta_k^*, v^*)$ ; //Estimate the heading according to the landmark graph, compass readings, and the last visited landmark  $v^*$ 
8      $[x_k, y_k] = \text{computeNextLocation}(l_k, \theta_k)$ ; //Compute the next location
9   end
10   $[x_r, y_r, R, r^*] = \text{landmarkDetect}(acc, gyro, baro, wifi, light)$ ; //Detect different types of landmarks
11   $[v\_id] = \text{landmarkMatch}(G, x_r, y_r, R, r^*)$ ; //Compute the belief of the detected potential landmark, if  $\text{bel}(v\_id) > \epsilon_{bel}$ , then return the id of this landmark in the landmark graph
12  if  $v\_id > 0$  then
13    /* If the detected landmark is confirmed as a landmark in the landmark graph */
14     $[x_k, y_k] = \text{correctError}(G, v\_id, loc, r^*)$ ; //Use the location of the detected landmark to correct the user's location, and return the current corrected location
15     $l_k = \text{updateStepLength}(G, l_{k-1}, v^*, v\_id, n)$ ; //Compute the step length according to the step length model, the last reference step length  $l_{k-1}$ , the last visited landmark  $v^*$  and the detected landmark  $v\_id$ , and the number of steps that the user takes to travel from  $v^*$  to  $v\_id$ 
16  end
17   $\text{print}(x_k, y_k)$ ; // Output the location estimation
18 end

```

TABLE I  
PARAMETER SETTING

Function	Parameter	Value
Acc landmark	Window size	50
	Walking state threshold	2 s
	Still state threshold	1 - 8 s
Gyroscope landmark	Window size	10
	Gyro threshold $\epsilon_{gyro}$	1.1 rad/s
Baro landmark	Pressure threshold $\epsilon_{baro1}$	0.05 hPa
	Pressure threshold $\epsilon_{baro2}$	0.3 hPa
WiFi landmark	Window size	3
	RSS threshold $\epsilon_{WiFi}$	- 42 dBm
Light landmark	Window size	10
	Light threshold $\epsilon_{light}$	100 lux
Landmark matching	Heading threshold $\epsilon_{theta}$	30°
	Belief threshold $\epsilon_{bel}$	0.25
WiFi Fingerprinting	Number of neighbors $K$	{1, 3}
	Number of particles	{200, 500}
Map Filtering	Variance of step length error	0.1 m
	Variance of heading error	10 degree
PDR	Initial step length $l_0$	0.65 m

Table I gives the values of parameters of our landmark detection methods, and those of the methods we used for comparison with our method.

#### A. Distribution of Landmarks

In total, there are 83 landmarks in our testing environment, including 5 accelerometer landmarks, 8 barometer landmarks,

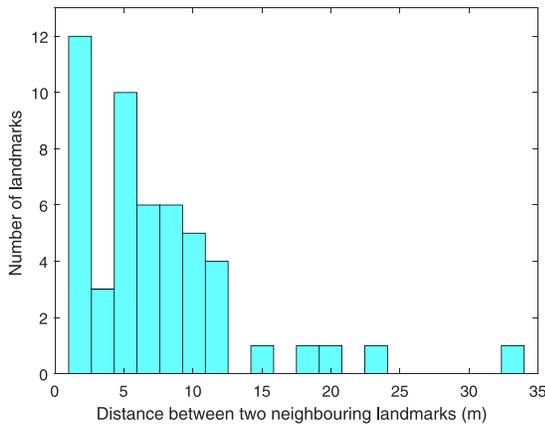


Fig. 8. Landmark distribution density.

24 gyroscope landmarks, 12 light landmarks, and 34 WiFi landmarks. It should be noted that some landmarks of different types are distributed at the same location point or share a small area (e.g., 1 m<sup>2</sup>). This is because different sensors would present a landmark pattern when the user performs some activities. For instance, both a user's motion state and walking direction change when she opens a door, which will impose a landmark pattern on the accelerometer readings and the gyroscope readings. These landmarks within a small area can be merged as a landmark with multiple detection rules. When one of these detection rules is met, this landmark would be verified and used to correct the user's location if it meets the belief requirement. After merging these landmarks within a small area into one landmark, we obtain 51 landmarks for assisting localization.

Fig. 8 shows the landmark distribution density. The biggest distance between two neighboring landmarks is about 34 m, while the median distance is about 6 m. This relatively densely distributed landmarks can ensure that our indoor localization method achieves an ideal accuracy, which we will show in the following section.

### B. Accuracy of Step Counting and Step Length Estimation

Both the step detection and step length estimation have an impact on the accuracy of localization. To evaluate the performance of the peak detection-based step counting method, we conducted experiments with six participants. Each participant was asked to walk 300 steps in two phone carrying ways: 1) normal, the participant carried the phone in a fixed pose (in a trouser's pocket) during walking and 2) free, the participant carried the phone in arbitrary poses during walking. Table II shows the accuracy of step counting using peak detection and peak detection with the periodicity and similarity constraints in the two phone poses. It depicts that the peak detection with constraints can achieve a higher accuracy of step counting (more than 97% on average) than the general peak detection method (about 94%). We argue that the achieved accuracy is high enough for the landmark graph-based localization as the accumulated error will be regularly calibrated by using landmarks information.

Then, we analyze the performance of step length estimation. The step length estimation used in this article is based on the

TABLE II  
ACCURACY OF STEP COUNTING

	Peak detection		Peak detection with constraints	
	Normal (%)	Free (%)	Normal (%)	Free (%)
User 1	98.3	99.7	99.7	97.7
User 2	93.3	92.7	98.7	97.3
User 3	94.3	94.7	97.3	99.7
User 4	93.7	94.3	97.0	99.7
User 5	91.0	90.0	96.0	96.3
User 6	93.7	97.0	97.3	98.7
Average	94.1	94.7	97.7	98.2

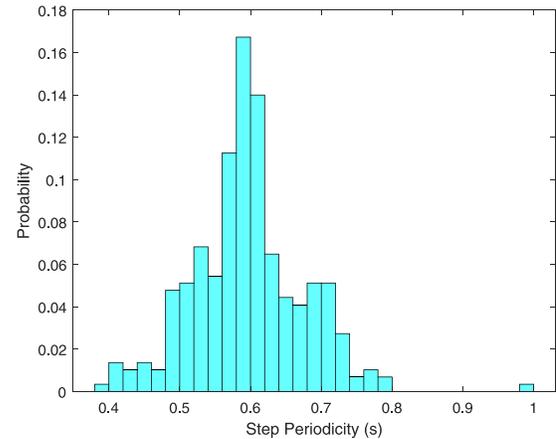


Fig. 9. Stability of step periodicity.

argument that the step length is relatively stable when a user walks naturally for a short period of time. For the same user, her step length is mainly affected by her walking frequency that corresponds to step periodicity. To justify the above argument, we analyze the stability of step periodicity. As shown in Fig. 9, the periodicity of most steps falls in the interval of [0.5, 0.7], which is relatively stable. It approximately follows a normal distribution. Considering that the distribution of landmarks is dense, the accumulated error caused by the step length estimation can be regularly corrected. More details about the step length estimation can be found in [48].

### C. Initial Location Determination

We randomly selected ten location points along the testing path as the respective starting location for ten tests (the system did not have the coordinate knowledge). Then, we tested the proposed HMM-based method to initialize the localization system. Table III shows the testing results of the ten location points. It shows that the average distance that a user has to travel in order to determine her initial location is about 9.95 m, which is relatively short for most applications.

### D. Localization Accuracy

We compare the localization accuracy of the proposed method with that of popular indoor localization methods, including PDR I (Acc + Compass), PDR II (Acc + Gyro) [51], WiFi fingerprinting [52], and map filtering [53]. Both PDR I and PDR II methods use the step counting-based method to calculate the step length, but PDR I uses compass readings (inferred from magnetometer readings and accelerometer

TABLE III  
DISTANCE REQUIRED TO DETERMINE THE INITIAL LOCATION

Point No.	Traveled Distance (m)
1	2.99
2	1.95
3	3.25
4	30.55
5	2.99
6	30.55
7	20.36
8	2.99
9	3.25
10	0.65
Average distance	9.95

readings) to estimate the heading while PDR II uses the gyroscope readings to compute the heading (with the compass readings to provide initial heading). For the WiFi fingerprinting method, a fingerprint database was constructed by using data of five trajectories in the training phase. The ground-truth locations of fingerprints in the training phase were obtained according to the marker’s data. The ground truth location between the two markers was computed by using the time interpolation method. We compared two fingerprinting methods with different parameters in the fingerprint matching process, namely, fingerprinting I (which chooses the location of the nearest neighbor in the fingerprint database as the localization result), and fingerprinting II (which uses the average of  $K$ -nearest neighbors’ location as the localization result,  $K = 3$ ). In the map filtering method, a particle filter was used to fuse the PDR II method with the floor plan. The particles that violated space constraints, such as passing through a wall or obstacle, were removed from the candidate samples by setting the weights of these particles to zero. We considered two map filtering methods with different particle number: 1) map filtering I (200 particles) and 2) map filtering II (500 particles).

Figs. 10 and 11 show the trajectories calculated by different methods. The reason why the trajectories seem to be discontinuous is that the testing path goes through two floors and we do not show the staircases in the figures for simplicity. The ground truth locations, which were obtained from the markers’ data, are marked with black circles. The trajectory (marked in red) computed by our method can completely match the ground-truth path. On the contrary, there is a big deviation in the trajectory by the PDR I (in green) and that by the PDR II (in blue) from the testing path. The trajectories obtained from the WiFi fingerprinting methods are usually matched with the testing path, but they might have some jumps because of the fluctuation characteristics of the WiFi signal. The map filtering methods can mostly be matched with the testing path, but sometimes cannot converge to the testing path when the error of the PDR becomes too large.

Fig. 12 shows the cumulative distribution function of the localization error achieved by different methods, from which we can see that the proposed method significantly outperforms other methods. Specifically, the proposed method reaches an accuracy of 88% with the error less than 1.5 m, which is much higher than that of the map filtering (about 63%) and that of the WiFi fingerprinting (about 33%). The PDR methods

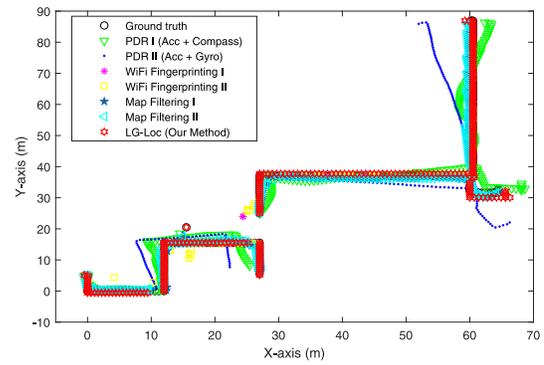


Fig. 10. Trajectories computed by different methods (on the fourth floor).

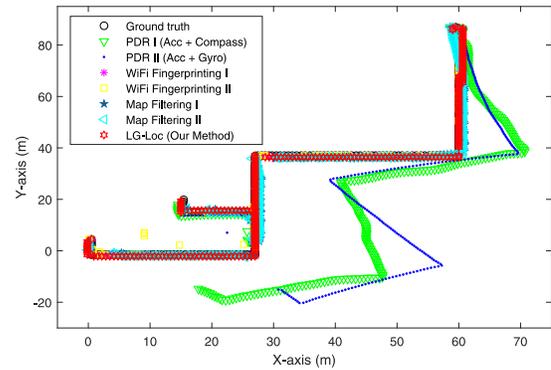


Fig. 11. Trajectories computed by different methods (on the fifth floor).

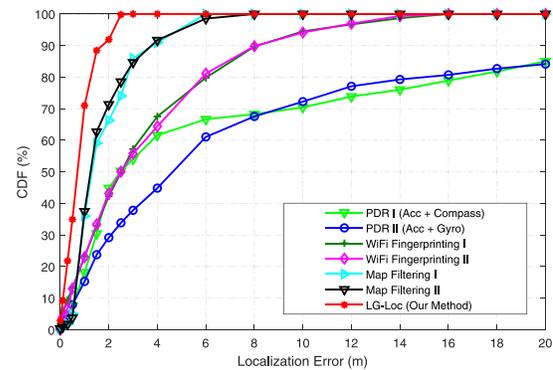


Fig. 12. Accuracy comparison.

perform worst since they do not consider spatial constraints or any prior knowledge (e.g., WiFi fingerprints). Table IV demonstrates the mean error of different methods. Our method performs the best with a mean error of 0.80 m, followed by the map filtering methods (about 1.7 m). The mean error of the WiFi fingerprinting method using NN for matching is similar to that of fingerprinting using KNN for matching (3.5 m). The PDR methods can only achieve a mean error of greater than 7-m due to the intrinsic error of inertial sensors.

E. Computational Efficiency

We compare the computational efficiency of our method with the map filtering method that is widely used to fuse a floor plan with PDR and can achieve a relatively high-localization accuracy. Table V shows the computation time for the

TABLE IV  
LOCALIZATION ERROR ACHIEVED BY DIFFERENT METHODS

Method	Mean Error (m)
PDR I	7.23
PDR II	8.90
WiFi Fingerprinting I	3.52
WiFi Fingerprinting II	3.52
Map Filtering I	1.77
Map Filtering II	1.69
<b>LG-Loc (Proposed method)</b>	<b>0.80</b>

TABLE V  
TIME EFFICIENCY COMPARISON (FOR A TRAJECTORY WITH A LENGTH OF ABOUT 360 m)

Method	Time Consumed (s)
Map Filtering I (200 particles)	10.66
Map Filtering II (500 particles)	24.26
<b>LG-Loc (Our method)</b>	<b>2.17</b>

estimation of a user's location on a trajectory with an approximate length of 360 m by our method and the map filtering methods. Our method performs about five times faster than the map filtering method with 200 particles. Increasing the number of particles would lead to a significant increase in the computation load of the map filtering method. This is because the map filtering methods require to detect whether the particles cross a wall or other obstacles, which is a frequent and time-consuming operation. In contrast, our method corrects the accumulated error of the PDR by using a landmark graph, which does not need to detect whether the estimated location crosses a wall or other obstacles. Therefore, our method is more suitable to run on the resource-limited computing platforms, such as smartphones and smartwatches.

### VIII. DISCUSSION

We have demonstrated a landmark graph-based indoor localization method, which is low cost, robust, and achieves high accuracy. By organizing sensory landmarks in a graph, we are able to achieve an accuracy under 1 m, which is much higher than the existing methods. To reduce the human effort to manually input her initial location or collect WiFi fingerprints, we have shown how the landmark graph can be effectively used to infer the initial location via an HMM. To deal with the issues of landmark association and missed landmarks, we designed a belief metric for accurate landmark matching.

While the proposed method has shown excellent performance, there are still some potential issues worth exploring in the future. First, the proposed method uses a variety of sensors in a smartphone, which may drain the smartphone's power quickly. Fortunately, recent advances in the battery capacity and quick charging technology are promising to help relieve the issue. Nevertheless, it will be worthwhile to investigate energy-saving strategies to reduce the sensors' sampling frequency or to turn off some sensors that are not used for a period of time. Second, the proposed method may not work well in open areas (e.g., large halls) where there are not enough landmarks. This issue should

be addressed by integrating more sensors or information (e.g., camera), which can further increase the scalability and robustness. Third, most landmarks are detected by using certain defined rules, which require to define certain thresholds empirically. Recent advances in machine learning, especially in deep learning can be useful for detecting landmarks in an unsupervised or semisupervised way [54], [55]. Fourth, we have not considered the cooperation between multiple devices, which is a promising direction for future research. Different devices may lie at different locations, exchanging the information (e.g., locations, distances to landmarks) between these devices will certainly improve the accuracy.

### IX. CONCLUSION

In this article, we presented a novel, low-cost, high-accuracy indoor localization method based on the proposed landmark graph. Landmarks can be detected by using sensor activities and building a correspondence between the sensor activity and location information. Compared to the existing indoor localization methods, our method can achieve a better localization accuracy and has a higher computation efficiency, which means it is more suitable to run on the mobile platform.

### REFERENCES

- [1] S. Winter, "Indoor spatial information," *Int. J. 3D Inf. Model.*, vol. 1, no. 1, pp. 25–42, 2012.
- [2] 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 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 197–210.
- [3] J. Shang, X. Hu, F. Gu, D. Wang, and S. Yu, "Improvement schemes for indoor mobile location estimation: A survey," *Math. Probl. Eng.*, vol. 2015, pp. 1–32, May 2015.
- [4] L. Pei, M. Zhang, D. Zou, R. Chen, and Y. Chen, "A survey of crowd sensing opportunistic signals for indoor localization," *Mobile Inf. Syst.*, vol. 2016, pp. 1–16, Jun. 2016.
- [5] F. Gu *et al.*, "Indoor localization improved by spatial context—A survey," *ACM Comput. Surveys*, vol. 52, no. 3, pp. 1–35, Jul. 2019.
- [6] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. ACM 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 169–182.
- [7] F. Evrennou and F. Marx, "Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1–11, Dec. 2006.
- [8] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, 2009.
- [9] C. C. Presson and D. R. Montello, "Points of reference in spatial cognition: Stalking the elusive landmark," *Brit. J. Develop. Psychol.*, vol. 6, no. 4, pp. 378–381, 1988.
- [10] I. Fellner, H. Huang, and G. Gartner, "Turn left after the WC, and use the lift to go to the 2nd floor—Generation of landmark-based route instructions for indoor navigation," *ISPRS Int. J. Geo Inf.*, vol. 6, no. 183, pp. 1–23, 2017.
- [11] F. Gu, K. Khoshelham, J. Shang, and F. Yu, "Sensory landmarks for indoor localization," in *Proc. IEEE 4th Int. Conf. Ubiquitous Position. Indoor Navig. Location Services (UPINLBS)*, 2016, pp. 201–206.
- [12] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.
- [13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI/IAAI*, 2002, Art. no. 593–598.
- [14] J. Shang, F. Gu, X. Hu, and A. Kealy, "APFiLoc: An infrastructure-free indoor localization method fusing smartphone inertial sensors, landmarks and map information," *Sensors*, vol. 15, no. 10, pp. 27251–27272, 2015.

- [15] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, "ALIMC: Activity landmark-based indoor mapping via crowdsourcing," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2774–2785, Oct. 2015.
- [16] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," presented at the 10th USENIX Symp. Netw. Syst. Design Implement. (NSDI), 2013, pp. 71–84.
- [17] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFI: Decimeter level localization using WiFi," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 269–282, 2015.
- [18] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.
- [19] X. Hu, J. Shang, F. Gu, and Q. Han, "Improving Wi-Fi indoor positioning via ap sets similarity and semi-supervised affinity propagation clustering," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 1, 2015, Art. no. 109642.
- [20] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 1st Quart., 2016.
- [21] J.-G. Park *et al.*, "Growing an organic indoor location system," in *Proc. ACM 8th Int. Conf. Mobile Syst. Appl. Services*, 2010, pp. 271–284.
- [22] H. Aly and M. Youssef, "New insights into WiFi-based device-free localization," in *Proc. ACM Conf. Pervasive Ubiquitous Comput. Adjunct Publication*, 2013, pp. 541–548.
- [23] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed, "Joint indoor localization and radio map construction with limited deployment load," *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 1031–1043, May 2015.
- [24] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, 3rd Quart., 2013.
- [25] C. Yu, H. Lan, F. Gu, F. Yu, and N. El-Sheimy, "A MAP/INS/Wi-Fi integrated system for indoor location-based service applications," *Sensors*, vol. 17, pp. 1–18, 2017.
- [26] A. De Angelis, J. Nilsson, I. Skog, P. Händel, and P. Carbone, "Indoor positioning by ultrawide band radio aided inertial navigation," *Metrol. Meas. Syst.*, vol. 17, no. 3, pp. 447–460, 2010.
- [27] M. Ramezani, D. Acharya, F. Gu, and K. Khoshelham, "Indoor positioning by visual-inertial odometry," *ISPRS Ann. Photogrammetry Remote Sens. Spat. Inf. Sci.*, vol. 4, pp. 371–376, Sep. 2017.
- [28] B. Xie *et al.*, "LIPS: A light intensity-based positioning system for indoor environments," *ACM Trans. Sensor Netw.*, vol. 12, no. 4, pp. 1–27, 2016.
- [29] Y.-L. Wei, C.-J. Huang, H.-M. Tsai, and K. C.-J. Lin, "CELLI: Indoor positioning using polarized sweeping light beams," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2017, pp. 136–147.
- [30] S. Liu and T. He, "SmartLight: Light-weight 3D indoor localization using a single led lamp," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst.*, 2017, pp. 1–14.
- [31] L. Ma, Y. Fan, Y. Xu, and Y. Cui, "Pedestrian dead reckoning trajectory matching method for radio map crowdsourcing building in WiFi indoor positioning system," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–6.
- [32] Y. Wu, P. Chen, F. Gu, X. Zheng, and J. Shang, "HTRACK: An efficient heading-aided map matching for indoor localization and tracking," *IEEE Sens. J.*, vol. 19, no. 8, pp. 3100–3110, Apr. 2019.
- [33] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Indoor tracking using undirected graphical models," *IEEE Trans. Mobile Comput.*, vol. 14, no. 11, pp. 2286–2301, Nov. 2015.
- [34] J. Shang, X. Hu, W. Cheng, and H. Fan, "GridiLoc: A backtracking grid filter for fusing the grid model with PDR using smartphone sensors," *Sensors*, vol. 16, pp. 1–22, 2016.
- [35] D. Acharya, M. Ramezani, K. Khoshelham, and S. Winter, "BIM-Tracker: A model-based visual tracking approach for indoor localisation using a 3d building model," *ISPRS J. Photogrammetry Remote Sens.*, vol. 150, pp. 157–171, Apr. 2019.
- [36] L. Díaz-Vilariño, K. Khoshelham, J. Martínez-Sánchez, and P. Arias, "3D modeling of building indoor spaces and closed doors from imagery and point clouds," *Sensors*, vol. 15, no. 2, pp. 3491–3512, 2015.
- [37] F. Gu, A. Kealy, K. Khoshelham, and J. Shang, "Efficient and accurate indoor localization using landmark graphs," *Int. Archives Photogrammetry Remote Sens. Spat. Inf. Sci.*, vol. 41, pp. 509–514, Jun. 2016.
- [38] F. Gu, A. Kealy, K. Khoshelham, and J. Shang, "User-independent motion state recognition using smartphone sensors," *Sensors*, vol. 15, no. 12, pp. 30636–30652, 2015.
- [39] H. Abdelnasser *et al.*, "SemanticSLAM: Using environment landmarks for unsupervised indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1770–1782, Jul. 2016.
- [40] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. IJCAI*, 2003, pp. 1151–1156.
- [41] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 1983–1993, Dec. 2012.
- [42] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [43] F. Gu, K. Khoshelham, J. Shang, F. Yu, and Z. Wei, "Robust and accurate smartphone-based step counting for indoor localization," *IEEE Sens. J.*, vol. 17, no. 11, pp. 3453–3460, Jun. 2017.
- [44] 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 *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 421–430.
- [45] H. Weinberg, "Using the adxl202 in pedometer and personal navigation applications," *Anal. Devices*, vol. 2, no. 2, pp. 1–6, 2002.
- [46] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *J. Global Position. Syst.*, vol. 3, no. 1–2, pp. 273–279, 2004.
- [47] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.
- [48] F. Gu, K. Khoshelham, C. Yu, and J. Shang, "Accurate step length estimation for pedestrian dead reckoning localization using stacked autoencoders," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2705–2713, Aug. 2019.
- [49] N. Roy, H. Wang, and R. R. Choudhury, "I am a smartphone and I can tell my user's walking direction," in *Proc. 12th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2014, pp. 329–342.
- [50] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Robust indoor positioning with lifelong learning," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2015.
- [51] T. Gallagher, E. Wise, B. Li, A. G. Dempster, C. Rizos, and E. Ramsey-Stewart, "Indoor positioning system based on sensor fusion for the blind and visually impaired," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, 2012, pp. 1–9.
- [52] A. W. S. Au *et al.*, "Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 2050–2062, 2013.
- [53] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "ZEE: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.
- [54] V. Radu, P. Katsikouli, R. Sarkar, and M. K. Marina, "A semi-supervised learning approach for robust indoor-outdoor detection with smartphones," in *Proc. 12th ACM Conf. Embedded Netw. Sensor Syst.*, 2014, pp. 280–294.
- [55] M. Ali, T. ElBatt, and M. Youssef, "SenseIO: Realistic ubiquitous indoor outdoor detection system using smartphones," *IEEE Sens. J.*, vol. 18, no. 9, pp. 3684–3693, 2018.