Landmark Graph-based Indoor Localization

Fuqiang Gu, *Member, IEEE,* Shahrokh Valaee, *Fellow, IEEE*,
Kourosh Khoshelham, Jianga Shang *, *Member, IEEE*, and Rui Zhang

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, existing indoor localization methods that make use of spatial information are either computationally expensive or sensitive to the completeness of landmarks. In this paper, we propose a novel, low-cost, high-accuracy indoor localization method based on a landmark graph. Experimental results show that the proposed method outperforms the state-of-the-art methods.

Index Terms—Location-based Services, Indoor Positioning, Landmarks, PDR, Smartphones, Spatial Information.

1 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, 4, 5], which use different techniques such as WiFi, Zigbee, Bluetooth, Ultra-wideband (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 a 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.

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 paper, we present *LG-Loc*, a novel and accurate landmark graph-based indoor localization method for smartphones. Compared to 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: offline and 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, 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 ob-

* *Corresponding Author*
Manuscript received XXX, 2019; revised XXX, 2019. This work is supported by the National Key Research and Development Program of China (No. 2016YFB0502200).
F. Gu and K. Khoshelham are with the Department of Infrastructure Engineering, University of Melbourne, Parkville, Victoria, 3000 Australia (e-mails: gufq87@gmail.com, k.khoshelham@unimelb.edu.au).
S. Valaee is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada (email: valaee@ece.utoronto.ca)
J. Shang is with School of Geography and Information Engineering and with National Engineering Research Center for Geographic Information System, China University of Geosciences, Wuhan, 430074 China (e-mail: jgshang@cug.edu.cn).
R. Zhang is with the Schoool of Computing and Information Systems, University of Melbourne, Parkville, Victoria, 3000 Australia (e-mail:rui.zhang@unimelb.edu.au)

tained from the floor maps. Then, we update the landmark graph by adding more landmarks (e.g., light landmarks, and WiFi landmarks) that can not 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 dead reckoning (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); 3) how to deal with landmark association issue and missing landmarks. By solving the above challenges, we have made the following contributions in this study:

- We propose a novel landmark graph-based method for indoor localization, which achieves a higher localization accuracy and is computationally efficient. Experimental results show that the proposed method performs the best with a mean error of 0.80 meters compared with the Map Filtering methods (about 1.7 meters), the WiFi Fingerprinting (about 3.5 meters), and the PDR (about 8 meters). 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.
- 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. Existing research considers only 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 problem are addressed by a landmark matching method based on a landmark graph.
- We propose a novel initial location inference method by using a landmark graph-based hidden Markov model (HMM). 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 user's active participation or a WiFi fingerprint database.

2 Related Work

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

2.1 Indoor Localization

WiFi-based indoor localization is a popular method since it can make use of 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 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, 19, 20]. 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 crowd-sourcing 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 current smartphones, limiting the applicability of CSI-based localization.

Dead Reckoning (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 the 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 have attracted the attention of many researchers [28, 29, 30]. It can achieve decimeter-level and even centimeter-level accuracy, and has potential to be applied in robot navigation, UAV (unmanned aerial vehicle) navigation, etc. However, unlike WiFi-based localization, which can make use of existing wireless access points, visible light-based localization requires modifying existing lighting infrastructure for the purpose of localization. Also, there is no a uniform international standard regarding visible lights for the purpose of indoor localization.

2.2 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 like trajectory matching [31] and 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 like 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, 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].

3 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 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 WiFi fingerprinting method (if the corresponding WiFi fingerprint database is available) or inferred via utilizing a hidden Markov model-based algorithm according to the landmark graph and sensor readings (if the WiFi fingerprint database is not available).

In the following, we will elaborate each of the three key modules: landmark graph, landmark matching, and location estimation with landmark graph constraints.

4 Landmark Graph

An initial concept of landmark graph has been presented in the our previous work [37]. In this work, we extend the landmark graph by introducing more types of landmarks, and automated update of 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.

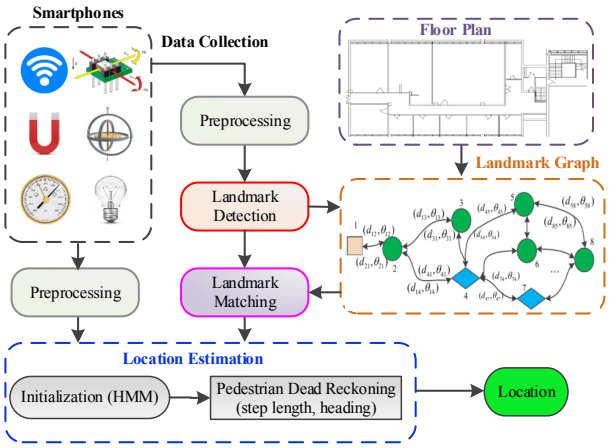

Fig. 1: System architecture of *LG-Loc*

4.1 Definition and Recognition of Landmarks

Landmarks in this research 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: distinctiveness, stability, and 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:

$$v =< (x, y), (\mathcal{R}_1, \cdots, \mathcal{R}_M) > \quad (1)$$

where (x, y) denotes the coordinate of the landmark, $(\mathcal{R}_1, \cdots, \mathcal{R}_M)$ 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 accelerometer readings and gyroscope readings.

Landmarks can be categorized by the type of the 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 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, still) to accelerometer data is based on a supervised classification method developed in our previous work [38]. The change pattern of "Walking $\longrightarrow$ Still (for a few seconds)

$\longrightarrow$ 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 $\mathcal{R}_{acc}$ of

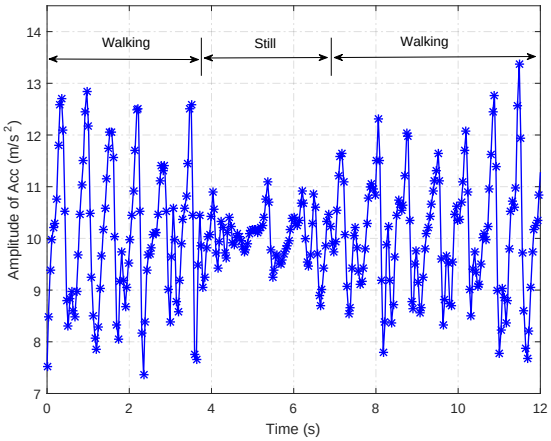

Fig. 2: The change in the amplitude of acceleration when a user passes through a door

accelerometer landmarks is defined as:

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

where m_t represents the user's motion state (e.g, walking, still) at time t, K_1 and K_2 are 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 $\longrightarrow$ Still (for a few seconds) $\longrightarrow$ Walking", the rule $\mathcal{R}_{acc}$ 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 $\mathcal{R}_{gyro}$ of a gyroscope landmark as follows:

$$\mathcal{R}_{gyro} = (loc_t | \left| \dot{\theta}_t \right| > \epsilon_{gyro}) \tag{3}$$

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 ϵ_{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, taking an elevator). Fig. 4 shows the change in the barometer

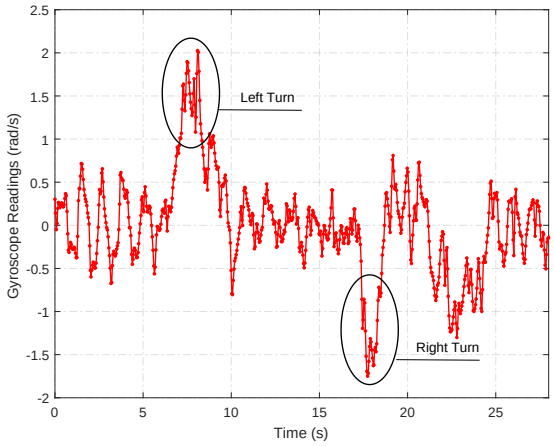

Fig. 3: The change in the gyroscope readings on the Z-axis when a user takes a turn (The user holds the phone in hand)

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 $\longrightarrow$ vertical movement". Similarly, the exit is detected by using the pattern "vertical movement $\longrightarrow$ 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 ϵ_{baro_1} and ϵ_{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:

$$\begin{aligned}\mathcal{R}_{baro_1} = (loc_t | (|p_i - p_{i-1}|) < \epsilon_{baro_1} \\ \&\& \left| \sum_{j=i+1}^{i+K_{p_1}} (sgn(p_j - p_{j-1})) \right| == K_{p_1} \\ \&\& |p_{i+K_{p_1}} - p_i| > \epsilon_{baro_2}).\end{aligned} \tag{4}$$

The first term is for detecting horizontal movement, and the latter two terms are for detecting vertical movement. sgn is the sign function, which is described as:

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

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

$$\begin{aligned}\mathcal{R}_{baro_2} = (loc_t | (|p_i - p_{i+1}|) < \epsilon_{baro_1} \\ \&\& \left| \sum_{j=i-K_{p_2}+1}^{i} (sgn(p_j - p_{j-1})) \right| == K_{p_2} \\ \&\& |p_{i-K_{p_2}} - p_i| > \epsilon_{baro_2}).\end{aligned} \tag{6}$$

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.

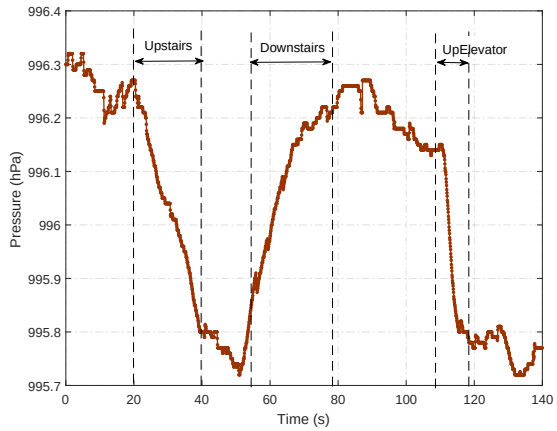

Fig. 4: The change in the pressure when a user takes stairs or an elevator

WiFi Landmark — We define a location point that overhears the strongest received signal strength (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 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 RSS_t denote the RSS from an AP at location point loc_t. Then, we can define the rule $\mathcal{R}_{WiFi}$ of a WiFi landmark as:

$$\mathcal{R}_{WiFi} = (loc_t|(RSS_t == max(RSS_{t-K_{WiFi}:t+K_{WiFi}})) \\ \&\& \; RSS_t > \epsilon_{WiFi})), \quad (7)$$

where ϵ_{WiFi} is a threshold to exclude 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.

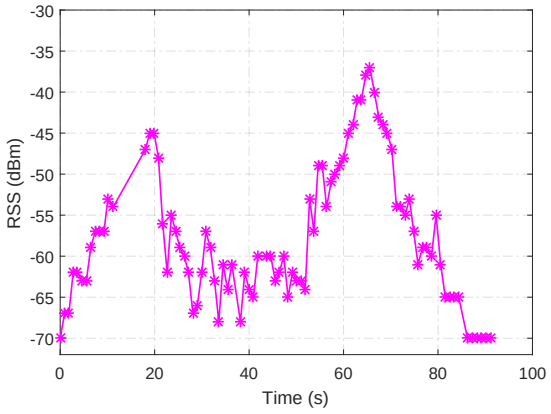

Fig. 5: Received signal strength from an AP at different locations

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 $\mathcal{R}_{light}$ of a light landmark as:

$$\mathcal{R}_{light} = (loc_t|(LX_t == max(LX_{t-K_{Light}:t+K_{Light}})) \\ \&\& \; LX_t > \epsilon_{Light})), \quad (8)$$

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

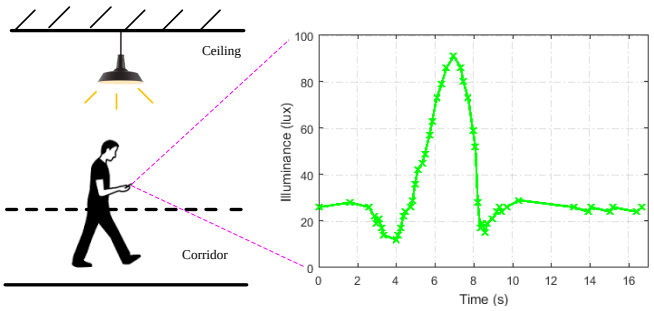

Fig. 6: The change of luminance when a user passes a ceiling lamp

4.2 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 $V = \{v_1, \cdots, v_N\}$ is a set of landmarks and $E = \{e_1, \cdots, e_M\}$ is the set of edges in graph G. Each edge $e_i =< v_j, v_k, \theta_{jk}, d_i >$ is a tuple consisting of two landmarks, the direction from one landmark to the other, and the distance between the two landmarks.

4.3 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 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 $\{v_1^i, v_2^i, \cdots, v_{n_i}^i\}$. 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 $dist(c_i, c_j)$ is a function used to compute the Euclidean distance between the center's coordinate of two landmark clusters c_i and c_j, and $rule$ 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.

Algorithm 1: Update the landmark graph

Input : An initial landmark graph G, N trajectories, each contains n_i potential landmarks denoted by $\{v_1^i, v_2^i, \cdots, v_{n_i}^i\}$
Output: A landmark graph G with newly-added landmarks

Assign the potential landmarks into N_0 initial landmark clusters denoted by $\{c_1, \cdots, c_{N_0}\}$, each cluster contains one potential landmark
// Distance constraint-based clustering:
repeat
  $\forall i, j$, **if** $dist(c_i, c_j) <= d_c \&\& rule(c_i) == rule(c_j)$ **then**
    Merge the two clusters into one
  end
until $dist(c_i, c_j) > d_c, \forall i, j$;
// Delete the clusters with too few elements:
foreach $c_i != null$ **do**
  if $size(c_i) < \epsilon_c$ **then**
    delete c_i
  end
end
Compute the size of remaining clusters, represented by N_c
// Update the landmark graph:
for $i++ <= N_c$ **do**
  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
  Compute the heading and distance from node v_i to each of its neighbouring nodes that are already in G
  Add all relative edges $e_j =< v_i, v_k, \theta_{ik}, d_j >$ into G
end

5 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 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:

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

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. θ_k and θ_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 location (x_t, y_t). δ is the Dirac delta function, which is defined as

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

and r is the rectangle function, described as

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

where ϵ_{theta} is a heading threshold. The function g is defined as

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

If there are multiple possible landmarks nearby, the one with the largest value of bel will be chosen as the detected landmark. To reduce the risk of mis-matching, we set a belief threshold ϵ_{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 ϵ_{bel}. We use only the landmarks with belief larger than ϵ_{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 $\longrightarrow$ Still $\longrightarrow$ 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 next landmark is detected.

6 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.

6.1 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 study, we present a novel way to infer the user's initial location which uses a landmark graph-based hidden Markov model (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 $S = \{v_1, \cdots, v_N\}$ denote a set of N hidden states (which are landmarks), $O = \{o_1, \cdots, o_M\}$ 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 $Q = \{v_1 v_2 \cdots v_M\}$ which best explains the observation sequence O. In the model λ, A and B are the state transition matrix and emission matrix, respectively, and π is the initial state distribution. Before the model can be exploited to infer location, we need to compute the transition matrix and emission matrix.

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:

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

$$\sum_{j=1}^{N} a_{ij} = 1, \tag{14}$$

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 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:

$$p_h = K \cdot p_l, \tag{15}$$

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

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:

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

where δ, r, and g are the same functions as in (9), and $deg^+(s_i)$ is the outdegree of s_i. R_k^* is the detection rule of the collected observation o_k, and R_j is the detection rule of s_j connected from s_i by an edge in the landmark graph. θ_k^* is the observed heading, and θ_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 $Q = \{v_1 v_2 \cdots v_M\}$ that best explains the given observation sequence $O = \{o_1, \cdots, o_M\}$. We define the Viterbi variable as:

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

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

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

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.

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
Extract the sequence of observations $o_1, o_2, \cdots, o_M$ based on collected sensor readings
Calculate the transition matrix $A = a_{ij}$ and the emission matrix $B = b_{ik}$ according to the landmark graph and the observation sequence
// Initialization:
for $i = 1 \rightarrow N$ **do**
  $\phi_1(i) = \pi_i \cdot b_i(o_1)$
  $\Psi_1(i) = 0$
end
// Recursion:
for $k = 2 \rightarrow M$ **do**
  for $j = 1 \rightarrow N$ **do**
    $\phi_{k+1}(j) = \max_{1 \leq i \leq N} [\phi_k(i) \cdot a_{ij}] \cdot b_j(o_{k+1})$
    $\Psi_{k+1}(j) = \underset{1 \leq i \leq N}{argmax} [\phi_k(i) \cdot a_{ij}]$
  end
    $P^* = \max_{1 \leq i \leq N} [\phi_k(i)]$
  if $P^* > \epsilon_{termination}$ **then**
    // location determined
      $v_M^* = \underset{1 \leq i \leq N}{argmax} [\phi_k(i)]$
    break;
  end
end
// Path backtracking:
for $t = k - 1 \rightarrow 1$ **do**
  $v_t^* = \Psi_{t+1}(v_{t+1}^*)$
end
// Location Inference:
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^*

In the above algorithm, the initial state distribution π 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 the subsequent observations, we can use the above algorithm to initialize the localization system. The initial location is determined when the maximum of the Viterbi variable ϕ_k is larger than a threshold $\epsilon_{termination}$. This means that the path becomes unique given k observation sequences.

6.2 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 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:

$$acc_t = \sqrt{acc_{x_t}^2 + acc_{y_t}^2 + acc_{z_t}^2}, \tag{20}$$

where acc_{x_t}, acc_{y_t}, and 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 over-counting 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 satisfy both the periodicity and similarity constraints is counted as one step.

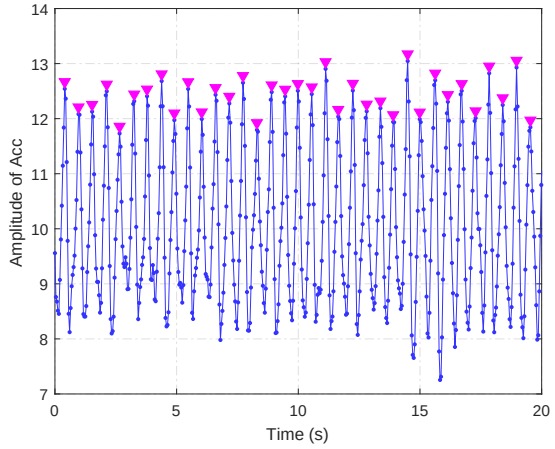

Fig. 7: Peaks of the acceleration when a user walks

Step length estimation — Different models, such as the Weinberg model [45], the Kim model [46], the linear model [47], and 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 research, 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:

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

where $dist(v_1, v_2)$ 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 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 research, 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 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 a 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 study) 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 θ_k represent the heading, then the location estimation using the PDR method can be written as:

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

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

$$\theta_k = f_\theta(\theta_k^*, \theta_k^r), \tag{24}$$

where f_θ is the function that determines the heading at step k according to the compass reading θ_k^*, and the heading information θ_k^r in the landmark graph. The step length l_k can be obtained by using equation (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.

Algorithm 3: Landmark graph-based location estimation

Input : A landmark graph G, sensor readings from accelerometer (acc), gyroscope ($gyro$), compass (θ^*), barometer ($baro$), WiFi ($wifi$), light sensor ($light$)
Output: Real-time location estimation

Location initialization using the proposed HMM-based method to obtain (x_0, y_0)
while *new sensor readings are available* **do**
  $[acc_{t-1:t}]$ = 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*
  if *var*$([acc_{t-1:t}]) > 0.5$ **then**
    / A step happens */*
    $l_k = l_{k-1}$;*//The initial step length l_0 is known*
    θ_k = estimateHeading(G, θ_k^*, v^*); *//Estimate the heading according to the landmark graph, compass readings, and the last visited landmark v^**
    $[x_k, y_k]$=computeNextLocation(l_k, θ_k); *//Compute the next location*
  end
  $[xr, yr, R, t^*]$= landmarkDetect (acc, $gyro$, $baro$, $wifi$, $light$); *//Detect different types of landmarks*
  $[v\_id]$ = landmarkMatch(G,xr, yr, R, t^*); *//Compute the belief of the detected potential landmark, if $bel(v\_id) > \epsilon_{bel}$, then return the id of this landmark in the landmark graph*
  if $v\_id > 0$ **then**
    / If the detected landmark is confirmed as a landmark in the landmark graph */*
    $[x_k, y_k]$ = correctError(G, v_id, loc, t^*); *//Use the location of the detected landmark to correct the user's location, and return the current corrected location*
    l_k = 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*
  end
  print(x_k, y_k); *// Output the location estimation*
end

7 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 meters.

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 pre-set path with the phone in hand, and reported the pre-set markers they encountered to evaluate the location accuracy. Note that the effect of 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 three meters. 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.

Table 1 gives the values of parameters of our landmark detection methods, and those of the methods we used for comparison with our method.

TABLE 1: 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 ϵ_{gyro}	1.1 rad/s
Baro landmark	Pressure threshold ϵ_{baro_1}	0.05 hPa
	Pressure threshold ϵ_{baro_2}	0.3 hPa
WiFi landmark	Window size	3
	RSS threshold ϵ_{WiFi}	- 42 dBm
Light landmark	Window size	10
	Light threshold ϵ_{light}	100 lux
Landmark matching	Heading threshold ϵ_{theta}	30^o
	Belief threshold ϵ_{bel}	0.25
WiFi Fingerprinting	Number of neighbors K	{1, 3}
Map Filtering	Number of particles	{200, 500}
	Variance of step length error	0.1 m
	Variance of heading error	10 degree
PDR	Initial step length l_0	0.65 m

7.1 Distribution of landmarks

In total, there are 83 landmarks in our testing environment, including 5 accelerometer landmarks, 8 barometer landmarks, 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 square meters). 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 meters, while the median distance is about 6 meters. This relatively densely distributed landmarks can ensure that our indoor localization method achieves an ideal accuracy, which we will show in the following section.

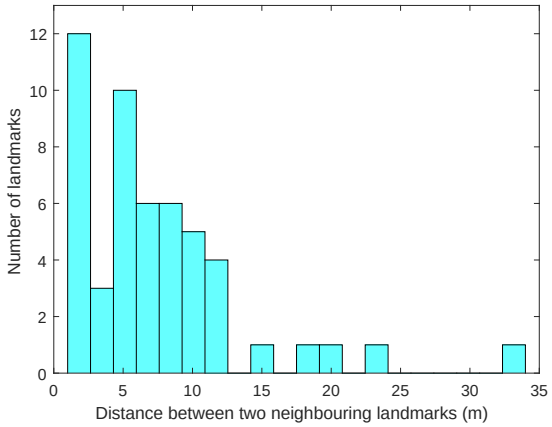

Fig. 8: Landmark distribution density

7.2 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; 2) free, the participant carried the phone in arbitrary poses during walking. Table 2 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 paper is based on the 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

TABLE 2: 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**

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].

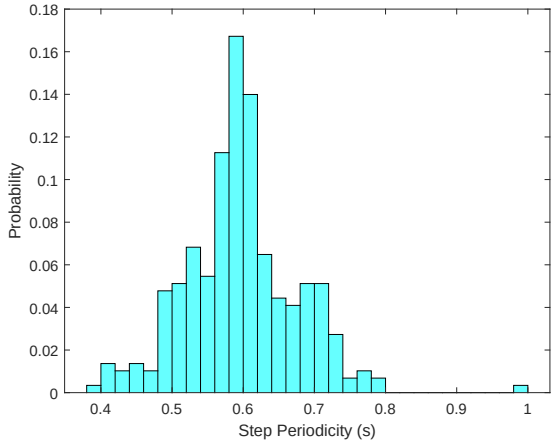

Fig. 9: Stability of step periodicity

7.3 Initial location determination

We randomly selected 10 location points along the testing path as the respective starting location for 10 tests (the system did not have the coordinate knowledge). Then we tested the proposed HMM-based method to initialize the localization system. Table 3 shows the testing results of the 10 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 meters, which is relatively short for most applications.

TABLE 3: 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

7.4 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 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 5 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 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 neighbour in the fingerprint database as the localization result), and Fingerprinting II (which uses the average of K nearest neighbours's 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: Map Filtering I (200 particles), and Map Filtering II (500 particles).

Fig. 10 and Fig. 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 meters, which is much higher than that of the Map Filtering (about 63%) and that of the WiFi Fingerprinting (about 33%). The PDR methods perform worst since they do not consider spatial constraints or any prior knowledge (e.g., WiFi fingerprints). Table 4 demonstrates the mean error of different methods. Our method performs the best with a mean error of 0.80 meters, followed by the Map Filtering methods (about 1.7 meters). The mean error of

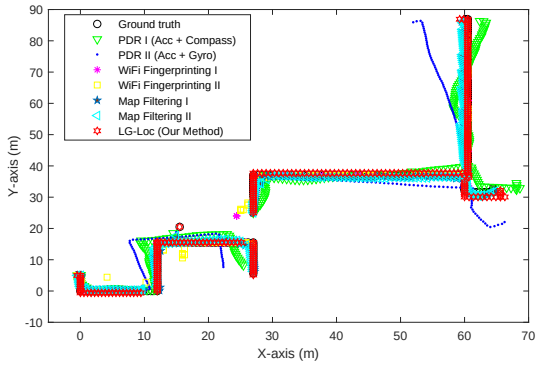

Fig. 10: Trajectories computed by different methods (on the fourth floor)

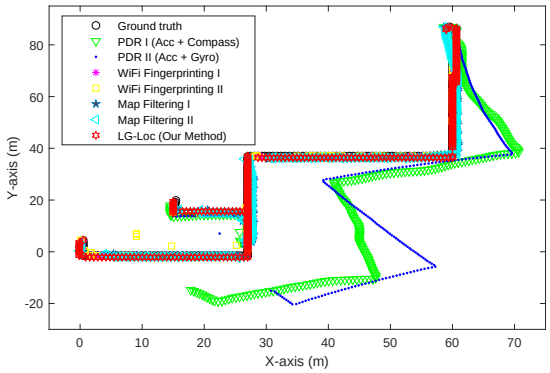

Fig. 11: Trajectories computed by different methods (on the fifth floor)

the WiFi Fingerprinting method using NN for matching is similar to that of Fingerprinting using KNN for matching (3.5 meters). The PDR methods can only achieve a mean error of greater than 7 meters due to the intrinsic error of inertial sensors.

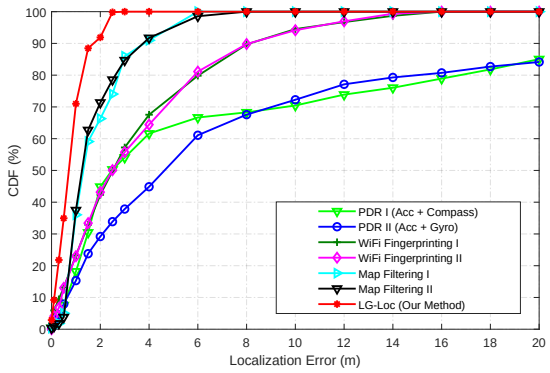

Fig. 12: Accuracy comparison

7.5 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 5 shows the computation time for the estimation of a user's location on a trajectory with

TABLE 4: 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
***LG-Loc* (Proposed method)**	**0.80**

an approximate length of 360 meters 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. By 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 smart watches.

TABLE 5: Time efficiency comparison (for a trajectory with a length of about 360 meters)

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

8 Discussion

We have demonstrated a landmark graph-based indoor localization method, which is low cost, robust, and achieves a high accuracy. By organizing sensory landmarks in a graph, we are able to achieve an accuracy under 1 meter, which is much higher than 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 a 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 that 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 is 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 semi-supervised 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.

9 Conclusion

In this paper, we present 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 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 mobile platform.

References

[1] S. Winter, "Indoor spatial information," *International Journal of 3-D Information Modeling (IJ3DIM)*, 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 *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 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," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[4] L. Pei, M. Zhang, D. Zou, R. Chen, and Y. Chen, "A survey of crowd sensing opportunistic signals for indoor localization," *Mobile Information Systems*, vol. 2016, 2016.

[5] F. Gu, X. Hu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, and J. Shang, "Indoor localization improved by spatial context-a survey," *ACM Computing Surveys*, vol. 52, no. 3, pp. 64:1–64:35, Jul. 2019.

[6] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "Fm-based indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 169–182.

[7] F. Evennou and F. Marx, "Advanced integration of wifi and inertial navigation systems for indoor mobile positioning," *Eurasip journal on applied signal processing*, vol. 2006, pp. 164–164, 2006.

[8] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the 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," *British Journal of Developmental Psychology*, 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 International Journal of Geo-Information*, vol. 6, no. 6, p. 183, 2017.

[11] F. Gu, K. Khoshelham, J. Shang, and F. Yu, "Sensory landmarks for indoor localization," in *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*. IEEE, 2016, pp. 201–206.

[12] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[13] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," *Aaai/iaai*, vol. 593598, 2002.

[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. 27 251–27 272, 2015.

[15] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, "Alimc: Activity landmark-based indoor mapping via crowdsourcing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2774–2785, 2015.

[16] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, 2013, pp. 71–84.

[17] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 269–282.

[18] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 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," *International Journal of Distributed Sensor Networks*, vol. 11, no. 1, p. 109642, 2015.

[20] 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.

[21] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie, "Growing an organic indoor location system," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 271–284.

[22] H. Aly and M. Youssef, "New insights into wifi-based device-free localization," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 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 Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1031–1043, 2015.

[24] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 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, no. 6, p. 1272, 2017.

[26] A. De Angelis, J. Nilsson, I. Skog, P. Händel, and P. Carbone, "Indoor positioning by ultrawide band radio aided inertial navigation," *Metrology and Measurement Systems*, 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 Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.

[28] B. Xie, K. Chen, G. Tan, M. Lu, Y. Liu, J. Wu, and T. He, "Lips: A light intensity–based positioning system for indoor environments," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 4, p. 28, 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 *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 136–147.

[30] S. Liu and T. He, "Smartlight: Light-weight 3d indoor localization using a single led lamp," in *Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2017.

[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 *2017 IEEE International Conference on Communications (ICC)*. IEEE, 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 Sensors Journal*, vol. 19, no. 8, pp. 3100–3110, 2019.

[33] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Indoor tracking using undirected graphical models," *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2286–2301, 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, no. 12, p. 2137, 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 Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 157–171, 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," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 41, 2016.

[38] ——, "User-independent motion state recognition using smartphone sensors," *Sensors*, vol. 15, no. 12, pp. 30 636–30 652, 2015.

[39] H. Abdelnasser, R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef, "Semanticslam: Using environment landmarks for unsupervised indoor localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1770–1782, 2016.

[40] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *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 Transactions on mobile computing*, vol. 11, no. 12, pp. 1983–1993, 2012.

[42] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[43] 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.

[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 *Proceedings of the 2012 ACM conference on ubiquitous computing*. ACM, 2012, pp. 421–430.

[45] H. Weinberg, "Using the adxl202 in pedometer and personal navigation applications," *Analog Devices AN-602 application note*, 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," *Positioning*, vol. 1, no. 08, p. 0, 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 Transactions on Instrumentation and Measurement*, 2018.

[49] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 2014, pp. 329–342.

[50] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Robust indoor positioning with lifelong learning," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2287–2301, 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 *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2012, pp. 1–9.

[52] A. W. S. Au, C. Feng, S. Valaee, S. Reyes, S. Sorour, S. N. Markowitz, D. Gold, K. Gordon, and M. Eizenman, "Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device," *IEEE Transactions on Mobile Computing*, 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 *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 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 *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, 2014, pp. 280–294.

[55] M. Ali, T. ElBatt, and M. Youssef, "Senseio: Realistic ubiquitous indoor outdoor detection system using smartphones," *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3684–3693, 2018.