

Received August 18, 2021, accepted September 28, 2021, date of publication October 13, 2021, date of current version October 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3119577

Smartphone-Based Indoor Tracking in Multiple-Floor Scenarios

THU L. N. NGUYEN¹, **TUAN D. VY**¹, (Student Member, IEEE),
KWAN-SOO KIM¹, (Student Member, IEEE), **CHENXIANG LIN**¹,
AND YOAN SHIN¹, (Senior Member, IEEE)

School of Electronic Engineering, Soongsil University, Seoul 06978, South Korea

Corresponding author: Yoan Shin (yashin@ssu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2020R1A2C2010006).

ABSTRACT With the rapid development of location-based services (LBSs), efficient and mobile-friendly localization algorithms should be designed for users to deliver a reliable LBS. In this paper, we present an algorithm with the corresponding smartphone app that enables users to calculate their locations based on representative infrastructures, such as nearby Wi-Fi access points and Bluetooth low energy (BLE) beacons subject to low-cost, rapid system deployment, and competitive location accuracy. Working under indoor multiple-floor scenarios, our app has three prominent features for estimating user locations. First, we establish a feature identifier to detect the current floor and the feasible area in which the user may walk. Second, owing to the structures of the indoor environment and the presence of different obstacles, the unpredictable variation of the received signal strength (RSS) in indoor environments is considered in the RSS-distance relationship to provide accurate location estimates. Third, with the prevalence of smartphones, we extract smartphone-inertial measurement units to learn users' behavior preferences, while collecting reference signals (e.g., Wi-Fi/BLE readings) along the pathway and input to the tracking algorithm. Then, the user's current location is displayed on the app. With this solution, we can provide an accurate location estimate with relatively low computational complexity regarding mobile device capability, while reducing labor costs from traditional fingerprint deployments. Finally, we test our tracking app in real-time multiple floor scenarios and evaluate the collected tracking data. Experimental results show that our proposed scheme achieves an average localization accuracy of more than 80% within a 2-m error bound in multiple-floor scenarios, while all areas (i.e., corridors, rooms, and stairs) were successfully identified.

INDEX TERMS Smartphone, indoor tracking, Bluetooth low energy, inertial measurement unit, multiple-floor localization.

I. INTRODUCTION

Recently, the design of efficient and effortless localization algorithms is essential for mobile users when tracking their locations in indoor environments. Although the global positioning system (GPS) works very well in outdoor environments, it has high localization errors in indoor environments because GPS signals do not pass through walls and obstacles in buildings. To provide a reliable solution in indoor environments, many indoor technologies have been developed, including Wi-Fi, Bluetooth, and radio frequency identification (RFID). Hence, the task of the localization problem is

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaodong Xu¹.

to extract location information based on the relevant signals according to the given infrastructure installed in indoor buildings [1]–[4]. With the wide range of mobile wireless systems, the aim of indoor localization tasks is to perform real-time and accurate object location using the current infrastructure. In indoor environments, Wi-Fi access points (APs) have become more prevalent because they provide Internet access and other location-based services (LBSs) on mobile devices (e.g., laptops and smartphones). Thus, Wi-Fi has become more popular for indoor tracking, while modern smartphones are able to track the direction and distance traveled [1]–[4]. As a result, the smartphone-based Wi-Fi and pedestrian dead reckoning (PDR) hybrid scheme is often integrated for self-tracking purposes. However, according to [5],

Wi-Fi-based localization on the iPhones is currently very difficult because it is not possible to extract Wi-Fi network information on iOS. Moreover, Wi-Fi-based ranging is inaccurate and produces large localization errors when tracking. Consequently, to upgrade the Wi-Fi-based localization on smartphones, a new dimension that considers wireless signals is required.

Over short distances, Bluetooth low energy (BLE) enables numerous emerging applications that are associated with the Internet of things (IoT), such as asset tracking, home automation, and smart factories. BLE beacons are typically small, inexpensive, and power-efficient. Each BLE beacon can be deployed either as a stand-alone or in combination with Wi-Fi as an AP to cover the area of interest (AoI) [5]. Objects can be tracked indoors using BLE beacons, mobile devices (e.g., smartphones), or BLE tags. Depending on the target application and budget, object tracking can be automatic (i.e., no human participation is needed) or semi-automatic (i.e., people participating with smartphones). In our study, we focus on semi-automatic systems. Here, a smartphone carried by a user is treated as a Wi-Fi/BLE receiver and exploits the reference signals to determine its location, which is the current user's location. Typical BLE broadcast packages contain a universally unique identifier (UUID) and a received signal strength (RSS), which can be detected by smartphones. Owing to its limited accuracy, BLE-based tracking methods are often combined with other technologies such as Wi-Fi and PDR to support tracking.

Regarding the two-dimensional (2D) case, recent works [5]–[12] have adopted fusion or hybrid algorithms that combine Wi-Fi/BLE and PDR to achieve consistent tracking. Most experimental activities take place on a single floor, such as a research lab [10], [11], complex areas consisting of office rooms and long corridors [5], or large-scale areas [12]. Conventional BLE scanners often measure either the presence status (i.e., the presence or absence of an object at a given location corresponding to a given proximity threshold) or report the distance to an object in centimeters. The observed received signal quality is inversely dependent on the distance between the smartphone and nearby reference nodes (RNs). Assuming that the RSS is known when the peripheral and central signals are 1m apart as A , a commonly used relationship between the distance and Bluetooth signal strength is given as in [17].

$$d \approx 10^{(A-R)/10n}, \quad (1)$$

where R is the RSS at the current distance and n is the radio propagation constant. In terms of location estimation, three major techniques were used to obtain the location from the RSS readings. These include triangulation, proximity, and fingerprints [6]–[9]. The triangulation and proximity techniques are more low-cost and energy-efficient, and they enable real-time localization because they do not require a site survey and database maintenance for radio maps. Meanwhile, the fingerprinting techniques in [18]–[20] require a site survey over the AoI to build the finest fingerprint database,

which may lead to increased costs, effort, and inflexibility to the indoor environment dynamics. Recently, a new fingerprinting-based localization was developed using channel state information (CSI) to construct fingerprint maps through CSI and deep learning [21]–[23], which frequently adapts to time and environmental changes. Nevertheless, the manual collection of the corresponding database is costly, not only for dataset construction but also for interpolated classification and analysis.

There are few practical approaches for three-dimensional (3D) space, such as [24]–[28]. With respect to multiple-floor scenarios, a typical 3D localization problem consists of the following three basic steps: (i) current floor detection, (ii) localization in a single floor, and (iii) detecting the areas in which the user may climb or descend stairs. With problem (i), the RSS profile-based floor-detection method is commonly used. In particular, it has been shown that the RSS from nearby Wi-Fi APs on multiple floors can be used for floor detection [26]–[28]. A lightweight fingerprint database can be utilized from the collected RSS readings during the training phase. Problem (ii) can be efficiently solved using the aforementioned approaches in the 2D cases [5]–[12].

In our research, we focused on localizing a mobile user in multiple-floor scenarios. In particular, we aim to develop a localization system for estimating the smartphone location using existing Wi-Fi/BLE infrastructure and inertial sensors with less human effort. Our proposed localization app is specifically built on an iPhone and is fully functional along with information got in advance of products such as Wi-Fi APs and BLE beacons. This is an extended version of our work [5], in which we want to address the following three main questions:

- (i) How do we detect the current floor and the feasible area in which the user may belong?
- (ii) How do we localize and track the user's location at a certain floor area?
- (iii) How can a floor that currently contains a user be detected between two adjacent floors?

To answer the above questions, a user with a smartphone collects RSS readings when the device is stationary at a specified location. Then, we learn about the predictive relationship between the RSSs of the smartphone and some special points near reference positions to build a regression model with minimum least squares error. When a user enters the AoI, the floor level is determined based on the currently connected Wi-Fi AP/BLE beacon list. Once that floor has been determined, the user's location as he/she moves is updated according to the received RSS signal, a prior light radio map, and AP/BLE knowledge. Smartphones with various inertial measurement unit (IMU) readings are exploited to reduce site survey efforts and record users' trace information along the walking path. Moreover, we establish some simple boundary check rules to adjust the estimated location in order to enable it to be smoother to avoid jumping too far, thus improving the user experience.

The remainder of this paper is organized as follows. In Section III, we present our tracking app components and the method used to collect information from BLE beacons and IMU information for localization purposes. In Section V, we present our tracking scheme for online applications, which corresponds to an AoI classification scenario. The experimental results are presented in VI, followed by concluding remarks in Section VII.

II. BACKGROUND

A. INDOOR LOCATION SENSING TECHNIQUES

Regarding the aforementioned indoor infrastructures, three main sensing techniques for obtaining related location information are described below.

1) MULTILATERATION

At a specific time, the smartphone first measures the RSS from the Wi-Fi AP within its communication range and translates it into a distance according to a specific signal propagation model. Then, the triangulation technique [1] calculates the location of the smartphone using a set of radial distances. This technique is very simple but can only be accurate when there exists a line-of-sight (LoS) between the smartphone and the Wi-Fi AP. Thus, one major disadvantage of this technique is that it suffers from environmental dynamics, which prevents researchers from developing accurate indoor localization systems based on this technique.

2) BLE PROXIMITY

The proximity of a BLE beacon can be obtained using several tools based on the smartphone operation. For example, in the iOS system, **CoreBluetooth** framework [29] scans nearby BLE beacons and provides the proximity of the smartphone to some nearby BLE beacons in the area. A BLE scanner such as an iPhone can run the BLE scanning function provided by this framework. In general, proximity-based localization selects the strongest RSS from specific BLE beacon and determines the smartphone location to be the region covered by the BLE beacon. Thus, it often provides a rough location estimate but is very easy to implement in a smartphone.

3) FINGERPRINTING

The fingerprinting technique is often used to obtain the spatial relationship between the RSS and the location. It computes the smartphone location by matching the online RSS readings to the fingerprints collected during the offline phase. Then, the matching process picks the RSS fingerprint, which has the smallest standard deviation to the online one owing to the time-varying nature of the indoor propagation channel, even at the same location. To support tracking, RSS fingerprints are collected during user movements. In particular, when a user enters a building, and his/her smartphone records the RSS fingerprints along his pathway.

The finalized fingerprints are then summed along different walking paths.

B. EXISTING LOCALIZATION SYSTEMS

First, in this section, the reference nodes (RNs) refer to nearby Wi-Fi APs, BLE beacons, or a combination of them, where the smartphones can obtain their reference signals within the communication range. With respect to localization within a single floor, because modern smartphones are already equipped with both Wi-Fi and Bluetooth receivers, they can collect the RSS readings from RNs to localize themselves. Several studies have attempted to achieve this goal via Wi-Fi-only [1]–[4], BLE-only [10]–[12], or a combination of them [5], [13]–[16], [24]–[27]. Most of the studies tried to jointly estimate the location of the smartphone based on a radio map of the AoI. The core idea is that using the online RSS reading, the smartphone matches it to the location of sample points whose RSS fingerprints are the closest match to the current smartphone location. The k nearest neighbor-based method [22] can be applied to estimate the location of the smartphone. In this way, it has low complexity and can handle the mobility of the user, but the accuracy of such schemes suffers from environmental dynamics, usually producing a location error of 5~10m. Alternative probabilistic methods such as histograms [30] and kernel-based schemes [32] can be utilized in the RSS database to calculate the user's location.

Recent works [5], [10]–[12], [31] showed that the performance of any hybrid wireless sensing tracking system depends on two factors: the RN deployment strategy and the description of the richness of information extracted from the hybrid signals. Table 1 summarizes some existing hybrid localization systems that have been successfully implemented on smartphones. We note that the localization accuracy of each work depends on several factors, such as the BLE deployment density, a comprehensive survey of the Wi-Fi/BLE fingerprint database, and the characteristics of the AoI. Moreover, there exists a trade-off between accuracy and cost when designing a pervasive indoor tracking system, especially in mobile environments. For example, many tracking algorithms [10]–[12] require that a sufficient number of BLE beacons cover the AoI, and their deployment spacing must remain sufficiently large to avoid interference from nearby Wi-Fi APs and other BLE beacons. Meanwhile, [5] designed a hybrid Wi-Fi/iBeacon indoor tracking system that overcomes the problem of limited public Wi-Fi information on iOS devices.

C. TECHNICAL CHALLENGES AND MOTIVATION

Although there are many systems and solutions for indoor environments that rely on Wi-Fi/BLE/IMU readings, the optimal tracking strategy is still an open problem. In contrast to the aforementioned works [10]–[12], [24]–[28], we identify current technical problems for 3D tracking systems on smartphones, especially in multiple-floor scenarios.

TABLE 1. Existing hybrid-based tracking systems on smartphones.

| System | Method | Range | Accuracy |
|--------|--|---|---|
| [5] | Smartphone IMU, lightweight Wi-Fi/iBeacon fingerprinting and location correction | Floorplan with two parts: one part of 39 m×17m and another one of 86m×24m | 1.5~5m with PDR/Wi-Fi fingerprint-only and 1~2m with hybrid PDR+Wi-Fi+iBeacon |
| [10] | Smartphone IMU-based system with iBeacon corrections | Research lab (19.0m×16.2m) or lecture hall (25.0m×17.0m) | 1~3m in research lab and 2~5m in empty hall |
| [11] | Map-based pinpointing and triangulation | Laboratory and office corridor | ≤ 3m |
| [12] | Hybrid IMU and Wi-Fi/iBeacon fingerprinting with particle filter | large-scale area (600m ²) | ≤ 2m |
| [24] | Integrated Wi-Fi RSS and IMU readings | Modern office building consists of 4 floors of 120m×40m | 3.9~7.4m |
| [25] | BLE/Micro electro-mechanical systems | Two floors: 64.6m×18.5m and 81.2m×18.5m | < 5.6m with only BLE or MEMS and < 1.7m with hybrid system |
| [26] | 3D Wi-Fi fingerprinting and PDR with particle filter | Three office floors in Warsaw Spire Office | 1.5~12m in 2D space and 1.5~6.5m in 3D space |
| [27] | A mixed Wi-Fi/BLE/IMU reading with beacons distribution and map-matching | Ground floor of a university building where each lap is equal to 242m | 2.48~5.2m |
| [28] | Sensor-assisted particle filter model for positioning in a multi-floor building by fusing Wi-Fi or Bluetooth signals and IMU | multiple floors where each floor is about 74.4m×37.2m | 2~4m and a floor detection accuracy of 83~94% |

1) APPLICATION PROGRAMMING INTERFACE (API) CAPABILITIES AND INTEGRATION

The Wi-Fi AP information can be extracted easily on Android smartphones via public API, but it is very difficult on iPhones owing to the secure Apple policy [33], [34]. An iPhone can only extract information from a single currently-connected Wi-Fi AP instead of a list of APs in the area. With this limited information, Wi-Fi usage has become optional for localization purposes. For example, the authors in [24] proposed that RSSs from nearby Wi-Fi APs can be utilized for floor detection by evaluating the principle of fingerprinting with a prior knowledge of certain areas. Such a method suffers from heavy database and performance degradation when the Wi-Fi AP density is low. And the most common beacon protocols used in iOS are Apple's iBeacon and Google's URIBeacon. By default, both of them make use of proximity sensing capabilities to estimate a user range as (1) from RSS measurements. However, BLE beacons have a variety of discriminating capabilities to recognize a specific location, subject to the inherent limitations of radio signal propagation. Converting a relevant distance from the RSS requires a practical model that reflects the effects of the deployed BLE locations in the AoI.

2) BLE LOCATION DEPLOYMENT

The challenge is to balance the trade-off between deployment cost and power consumption during the offline phase (e.g., AoI survey) and online phase (e.g., power consumed in each BLE scanning, system calibration). The density of BLE beacons must be sufficient for the tracking app to accurately localize a user within a short processing time, while the distance between them must be greater than a safe threshold to avoid interference between them and nearby Wi-Fi APs. Under the assumption that only existing Wi-Fi APs, BLE, and smartphones are used without additional devices, we would like to design a precise tracking app using a certain number of pre-deployed BLE beacons and a smartphone.

3) DATABASE COLLECTION AND ENVIRONMENT RECOGNITION

Any indoor tracking application relies heavily on the map of the AoI, which can illustrate different features such as rooms, corridors, and staircases. The map database generation should extract these features and engage with smartphone-based sensor readings during the user's moving trajectory. Collecting more Wi-Fi/BLE samples at multiple locations improves localization accuracy, but also introduces large errors and affects the user's experience. Intuitively, recording more RSS samples at the same location at different times and averaging RSSs from the same Wi-Fi AP/BLE location as a fingerprint can mitigate random noises and contribute to a robust solution. Owing to the presence of obstacles indoors, it is necessary to involve relative measurement errors from an unpredictable variation of reference RSS readings and motion sensors. From the users' perspective, many fingerprints in the database cause a long query time to determine the area where the user is currently located, thus affecting the overall localization system.

4) HYBRID LOCALIZATION ALGORITHM

With the above observations, a hybrid location estimation is required in order to accurately estimate user location with a short processing time. Because various measurement errors may lead to inaccurate localization results, online calibration and drift compensation should be adopted according to the geometric constraints.

In our design, we do not aim to provide the most accurate solution for the given indoor infrastructure, but try to uncover the true potential of these tools. Our tracking app takes advantage of the currently-existing Wi-Fi and only uses some BLE beacons to provide extra information, while combining with inertial sensing to monitor mobility levels. Thus, the proposed solution includes a lightweight fingerprint assembly collection module, a robust regression model for RSS-distance conversion, and

a hybrid localization algorithm. We demonstrate how to solve the previously mentioned problems in the following section.

- We observe that a good tracking app should not follow any complex smartphone operations. To overcome the Wi-Fi difficulties on iPhones, we applied the proximity method in [5], where a Wi-Fi status bar is converted to a related range. Using this approach, the smartphone can easily determine the proximity of nearby Wi-Fi APs within the network perimeter. Furthermore, instead of using the default settings for the BLE measurement model, we consider several factors, such as the path loss component and attenuation. Considering the randomness, the RSS diminishes with distance according to a certain propagation model.
- One of the advantages of the BLE beacon is that it can be placed freely within the AoI and provides good signal geometry. BLE beacons can be installed along with the available Wi-Fi APs in the AoI to improve the localization accuracy. Inspired by [5], we deploy BLE beacons over the AoI in an optimal strategy to reach the currently-existing Wi-Fi coverage and remain clear from the interference of other nearby BLE beacons. Through the training phase, we classified the AoI into three areas that represent three types of environments according to the RSS readings. For each partition, we calculated the signal propagation parameters, such as the path loss exponent and attenuation factor. Thus, when a user comes online, it can predict and control the RSS fluctuations over both time and space by adjusting these parameters.
- For data collection based on Wi-Fi/BLE, we build a sparse 3D database of the geometry captured by several key points. A key point consists of the spatial location and pre-recorded RSS readings. For example, a typical key point is constructed as $\kappa_{ij} = \langle ID_{ij}, (x_{ij}, y_{ij}), RSS_{ij} \rangle$, where i is the floor level, j is the number of orders, and RSS_{ij} is the corresponding RSS record in the database. In terms of smartphone-based inertial sensing, because the PDR relies on the IMU readings, we studied several relevant factors such as step detection and counting, stride length, and heading direction to select robust and reliable estimation models for the user's mobility characteristics.
- A hybrid indoor localization algorithm collects Wi-Fi BLE readings with smartphone-based IMU to estimate the user location along a walking path. Moreover, RSS readings can be treated as a random process and its path loss needs to be modeled depending on whether the smartphone is in LoS, partial-none-line-of-sight (p-NLoS), or full NLoS condition. Thus, with a good BLE deployment strategy and suitable path loss modeling, we aim to track a smartphone in indoor environments. To achieve this goal, we use a smartphone-assisted IMU as extra information for mobility tracking in our localization app.

III. PROPOSED LOCALIZATION APP COMPONENTS

As illustrated in Fig. 1, we introduce three major components of our tracking app: (i) An RN collection layer consists of a Wi-Fi/BLE scanner. The scanner runs a scanning function provided by the corresponding RN type and an API library with the smartphone operation. For example, in the iOS system, the proximity function in [5] provides a trusted-range for Wi-Fi, whereas the **CoreBluetooth** framework [29] scans nearby BLE beacons and estimates the distance of the smartphone to some nearby BLE beacons in the area. The RN profile is subject to a lightweight and lower computational load. (ii) A smartphone-IMU collector enables the user to retrieve mobility data via the **CoreMotion** framework [35] and to update the user location via **CoreLocation** [36]. (iii) Location estimation and calibration modules periodically collect data from (i) and (ii), run the tracking system, and display the current user location on the app screen. This app can track the location of the user device independently using the online RSS and IMU readings of the user device. It also helps to identify either stationary or moving to another floor location or inside an intended area.

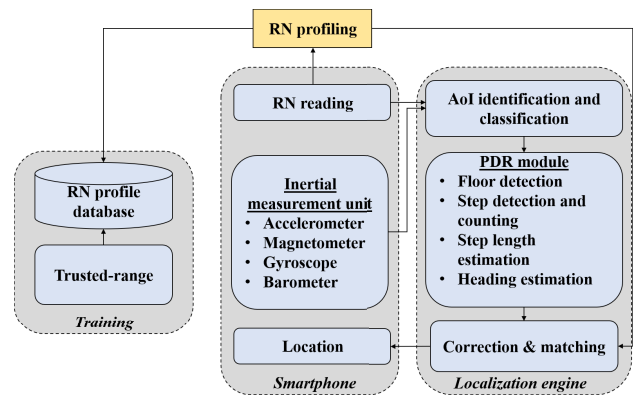


FIGURE 1. Our tracking app architecture.

A. AoI CLASSIFICATION

1) REGION CLASSIFICATION METRIC

To identify a feasible region of the point of interest, let ϕ be the dissimilarity function between two RSS vectors $\mathbf{f} = [f_1, \dots, f_n]^T$ and $\mathbf{f}' = [f'_1, \dots, f'_n]^T$ in \mathbb{R}^n , which is measured by the Euclidean distance between them as

$$\phi(\mathbf{f}, \mathbf{f}') = \sqrt{\sum_{j=1}^n \delta_j^2} = \sqrt{\sum_{j=1}^n |f_j - f'_j|^2}, \quad (2)$$

where $\delta_j = |f_j - f'_j|$ is the dissimilarity between two fingerprints. To retrieve a fingerprint \mathbf{f}^* in the fingerprint database space \mathcal{F} , we find the best match that achieves the highest similarity with respect to \mathbf{f} , that is,

$$\mathbf{f}^* = \arg \min_{\mathbf{f}' \in \mathcal{F}} \phi(\mathbf{f}, \mathbf{f}'). \quad (3)$$

Given a predefined threshold δ , if the dissimilarity δ_j satisfies $\delta_j \geq \delta$, two fingerprints f_j and f'_j are treated as two

different points in the fingerprint space. Otherwise, they are indicated at the same point. The distinguished fingerprint percentage is defined as

$$P_{df} = \frac{\#\{\delta_j : \delta_j \geq \delta\}}{\text{Number of samples}} \times 100\%. \quad (4)$$

Alternatively, P_{df} is the probability that two RSS samples are identified as different fingerprints.

2) FLOOR AND ROOM DETECTION BASED ON RSS PROFILE

For each scan, our localization app ran a feature detection algorithm and identified various key points. By matching key points from the database, we obtain the following information: (i) the current Wi-Fi AP/BLE beacon location to which the user may be close, and (ii) predictions on floor level and room level to which the user may belong. Before applying the location methods to the database, we need to define a subset of the associated BLE beacons and Wi-Fi APs, which can easily distinguish RNs. In our tracking app, we generated an RN map according to floor level and floor types, such as $\mathcal{F}_{map} = \{\mathbf{p}_f^j, Feature_j, j = 1, \dots, n_F\}$. Here, $\mathbf{p}_f^j = (x_f^j, y_f^j, z_f^j)$ is the location of the RN, $Feature_j = \langle \text{floor level, room/corridor type} \rangle$ is the RN characteristic, and n_F is the total number of RNs defined on the indoor map. For each RN, there is a unique identifier for easy retrieval on a smartphone. For example, it can extract Wi-Fi information, such as the MAC address and service set identifier from the AP. However, a BLE beacon advertising packet consists of three numerical identifiers: UUID, major number, and minor number. Major and minor numbers can be modified according to classification purposes. A lookup table consists of all RN features, so the smartphone easily retrieves related information from the RNs. In summary, we determined the current floor according to the following rules:

- (i) We select the RN with the strongest RSS signal. This point can be near the Wi-Fi AP or BLE beacon.
- (ii) In the case of hybrid measurements, if the smartphone recognizes that both the Wi-Fi AP or BLE beacon have the same range, we prioritize the location of that BLE beacon.
- (iii) If the smartphone detects that it is in a transition zone (e.g., a transition zone between two consecutive floors), the floor detection function is not updated until it receives a clearer signal from the other RNs. The floor detection was then processed according to (i).

With respect to room detection, we follow the above rules. However, it is more difficult to distinguish between two consecutive rooms, as shown in the experiment below. Therefore, we need to combine different signals, such as IMU readings, to accurately track the phone.

To test the floor/room detection scheme, we conducted the following two series called floor-to-floor (F2F) and room-to-room (R2R) experiments, which are illustrated in Fig. 2. The BLE beacons (in particular, RedBear iBeacon [37]) were placed on the ceiling based on the instructions in [5], [38].

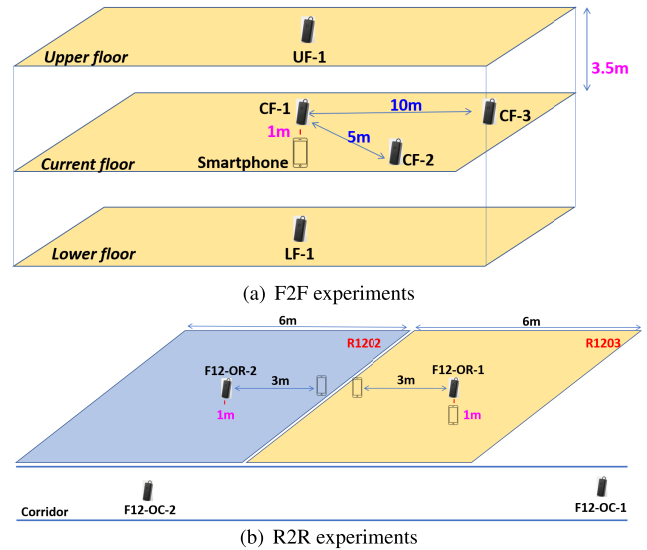


FIGURE 2. Experimental illustrations in two recognition scenarios.

A smartphone (e.g., iPhone 12) held by a user is under the iBeacon's location with an actual distance of 1m, but the ground distance is 0m. According to the figure, we draw the actual distance in cyan color and the ground distance in blue. For each experiment, the iPhone scans for 10 minutes each time during various times of the day (e.g., morning, noon, and night). Note that the smartphone can scan more than one RSS reading from a nearby BLE beacon within one second, but we take only one RSS value. If more than one value is recognized, our app takes the average of the RSS readings in one second. Thus, there are around 60 samples per minute and we obtain a dataset of 1800 RSS samples for a single BLE beacon during the collection day.

- **F2F experiment:** The iPhone collects RSS readings from nearby BLE beacons on the same floors at different ground distances (e.g., 0m, 5m, 10m) and BLE beacons from upper and lower floors. To convert a 3D distance d to the ground truth distance d_g , we use the following formula:

$$d_g = \sqrt{d^2 - h_d^2}, \quad (5)$$

where h_d is the difference between the ceiling height and the user's height. We then check whether the iPhone can correctly detect the current floor on which the user is at an acceptable rate (e.g., 70~80%).

- **R2R experiment:** We conduct the experiment in two adjacent offices, where each room has a BLE beacon. The iPhone is held by the user receiving the BLE signals at different ground distance locations (e.g., 0m and 3m).

From the collected data, we calculate the floor/room detection rate as

$$\text{Detection rate} = \frac{\text{Number of success detected cases}}{\text{Total number of samples}} \times 100\%.$$

To evaluate the floor and room detection performance, we integrated the results from different experimental areas.

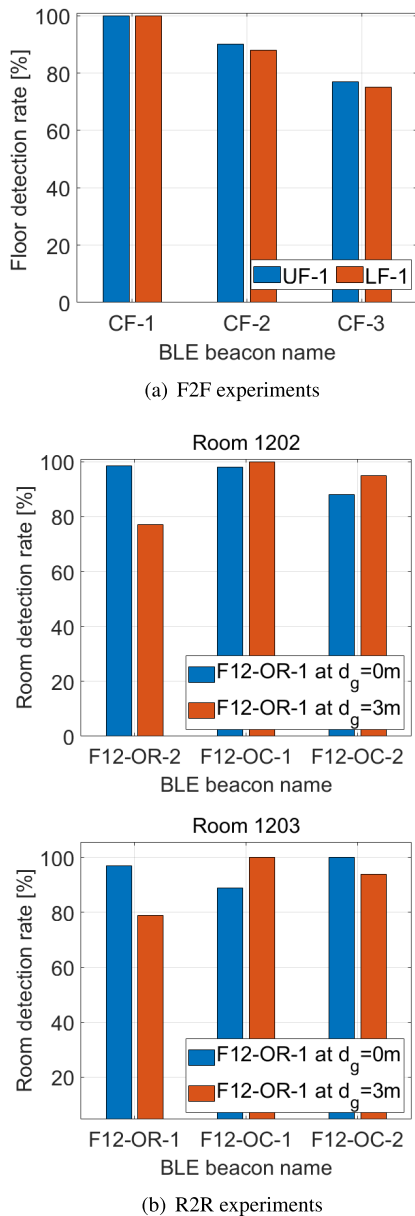


FIGURE 3. Detection rate performance with two recognition scenarios.

As shown in Fig. 3, the average detection rate is about 88.3% for floor detection and more than 80% for room detection. For the multi-room experiment, each room has a different room detection probability owing to the different number of obstacles in the rooms. From the results, we observe that under an appropriate BLE beacon installation, we achieve accurate detection results with sufficient confidence to be used for future tracking.

3) SMALL REGION CLASSIFICATION

After recognizing the floor and room to which the user may belong, we aim to decide sample locations into a distinguishable area based on signal classification. For a specific location, we observe the RSS inconsistency caused by signal fluctuations and obstacle blockages, and reveal the

RSS uncertainty with respect to the physical distance. We consider three types of environments in our tracking system: LoS, p-NLoS, and NLoS. In an LoS environment, the BLE signals received by smartphones from the same or close locations exhibit relatively stable RSSs. Thus, location errors from unmatched fingerprints do not fluctuate drastically, as is the case with RSS readings. In the p-NLoS environment, the BLE signals are blocked with low NLoS coefficients, such as glass windows, ambient dynamics, and human body blockages. Finally, in NLoS environments, BLE signals are blocked with high NLoS coefficients, such as metal doors and cinder walls. It can be recognized that the RSS variance is high because the RSSs fluctuate drastically. In addition, to differentiate the p-NLoS area from the NLoS area, the compass patterns from the IMU readings provide clues to distinguish between the two types of areas. In the NLoS area, the variance in the compass reading is much larger than that in the p-NLoS area. By performing several practical experiments in [5], we observe that it usually varies from $4\sim 10^\circ$, as observed from real user traces.

This space regionalization method applies to both Wi-Fi and BLE signals. Specifically, we adopt the classifier in [39] to separate a dataset into smaller ones and analyze their statistical features. In the following section, we present specifically BLE signal classification. During the offline phase, the distribution of the RSS readings from each BLE beacon was evaluated. Based on the fact that the probability of RSS measurements in LoS conditions is generally expected to be centralized, given an observation vector $\mathbf{x} \in \mathbb{R}^S$, we calculate the kurtosis as its statistical feature [40].

$$\kappa = \frac{\sum_{i=1}^S (x_i - \mu)^4}{\sigma^4}, \quad (6)$$

where μ and σ are the sample mean and standard deviation of the observation vector, respectively. We note that each data trace in our database was labeled with the corresponding scenario for online phase use. When we place a BLE beacon in the AoI, we also associate the probability that a region falls into the three aforementioned types.

4) TRANSITION REGION DETECTION

Previous works [24]–[26] exploited barometer changes to detect a user transfer from one floor to another. A barometer measures the atmospheric pressure value in millibars (mbar), which can be converted to height in meter. In particular, it computes the height h from the barometric pressure, as in [17].

$$\begin{aligned} h(p_0, p) &= \frac{T_0}{L} \left[1 - \left(\frac{p}{p_0} \right)^{\frac{RL}{gM}} \right] \\ &= 44330 \cdot \left[1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5.255}} \right]. \end{aligned} \quad (7)$$

Here, p and p_0 are the measured currently location air pressure and standard sea level pressure in mbar, respectively; $T_0 = 1,013.25\text{mbar}$ is standard sea level pressure,

TABLE 2. Experiments data on barometer measurements.

| Record | Average barometer (mbar) | | | Time (YYYYMMDD) | Weather condition | Temperature (°C) |
|--------|--------------------------|------------|------------|--------------------|----------------------|---------------------|
| | 11th floor | 12th floor | 13th floor | | | |
| 1 | 99.01 | 98.97 | 98.92 | 20210617 | Cloudy | 20~25 |
| 2 | 98.53 | 98.49 | 98.45 | 20210618 | Rainy | 20~22 |
| 3 | 98.53 | 98.48 | 98.45 | 20210619 | Rainy | 19~21 |
| 4 | 98.70 | 98.65 | 98.61 | 20210620 | Rainy | 19~21 |
| 5 | 99.24 | 99.20 | 99.15 | 20210621 | Sunny | 24~28 |
| 6 | 99.39 | 99.35 | 99.31 | 20210622 | Sunny | 20~25 |
| 7 | 99.71 | 99.67 | 99.63 | 20210623 | Rainy | 21~24 |

$L = 9.8^\circ\text{C}/\text{km}$ is the temperature lapse rate, $R = 287.057\text{J}/(\text{kg} \cdot \text{K})$ is the universal gas constant of air, $g = 9.807\text{m}/\text{sec}^2$ is the Earth's gravitational acceleration, and M is the molar mass of dry air.

It has been shown in [41] that weather patterns reflect the relationship between the barometric pressure and the temperature. Low temperatures can create higher air pressures than high temperatures because the air has denser molecular density. By calculating (7), we can determine the height of the current user's location, including the floor level and the transition area between the two floors (e.g., staircase). However, by performing practical experiments, we observed that this method depends on the weather conditions and the given data from a reference floor. Table 3 shows the barometer measurements collected on three office floors during a week. From the experimental results, we realize that the above pressure-altitude conversion method has the following disadvantages. First, the estimated altitude depends on the ambient temperature, which varies under different weather conditions. Thus, it may not be possible to distinguish the altitude between two adjacent floors, leading to inaccurate location calculations. Second, to perform the localization on a particular floor, we need to perform data normalization on one floor and then validate the pressure change of the other floors based on this reference floor data.

To overcome the aforementioned problems, we propose a method to detect users entering the transition zone between two floors according to the following signs.

- *Sign 1:* We frame the transition region in the AoI with a rectangle called the bound rectangle. The rectangle is defined by four given corner points. At a certain time, if the calculated location falls inside this rectangle, we predict that the user has entered the transition region between two adjacent floors.
- *Sign 2:* When the user goes up/down the stairs, the accelerometer values fluctuate significantly from the ones in normal walking, and we know that the user has entered the transition zone by the stairs. In this case, the variance of acceleration becomes larger, and the distance between two adjacent acceleration peaks becomes shorter than that of normal walking. If the user walks into the elevator, the radio signals are blocked and the acceleration patterns are mostly straight and long. Because we assume that the user does not spend much time in the transition zone, we do not investigate the characteristics of different functional areas in the zone.

- *Sign 3:* The barometer change provides a clue to differentiate between upstairs and downstairs. Under the practical assumption that the ambient temperature on all floors is the same, the pressure decreases with increasing altitude. By observing this change, we can predict the direction of movement of the user in the region.

B. RSS-DISTANCE CONVERSION FOR RN SIGNALS

In this section, we use the RSS-based ranging technique to map the signal strength to a relevant distance. Because radio signals degrade during propagation, understanding radio attenuation in an AoI helps us to estimate this distance more accurately. Generally, with any RN type (Wi-Fi AP or BLE beacon), the same RSS-to-distance conversion can be applied with a different set of channel parameters (e.g., reference path loss at reference distance, path loss exponent, etc.). In special cases, such as iOS devices, which cannot read the RSS signal in dBm, the proximity conversion table in [5] is applied based on the Wi-Fi status bar. Below, we show how to develop a relationship between the RSS signal and the estimated distance for BLE RN.

An intuitive way to overcome the traditional fingerprint problem is to recognize and discard outdated entities before fingerprint matching. We denote $L(d)$ and $L(d_0)$ as the RSS values for BLE signals at actual distances d m and d_0 m from the smartphone, respectively. Mathematically, the relationship between the distance and the RSS can be represented by

$$L(d) = L(d_0) - 10n \cdot \log_{10} \left(\frac{d}{d_0} \right) - X_g, \quad (8)$$

where n is the path loss exponent, and X_g is a normal random variable with zero mean and standard deviation σ . The value of σ was empirically determined through practical measurements in the AoI. By collecting a series of reference RSS values versus distance, we can specify the parameters of the model (8) and use them to estimate the RSS-based range. For simplicity, the estimated distance d is calculated for each BLE beacon using the following formula:

$$\hat{d} = d_0 \cdot 10^{\frac{L(d_0) - L(d) - X_g}{10n}}. \quad (9)$$

The relationship between the estimated distance (9) and the ground truth distance can be found in (5). Under the current indoor conditions, we collected real-time RSS readings of a RedBear beacon measured by an iPhone 12 at a reference

distance $d_0 = 1\text{m}$. In order to find the related parameters for each area, we applied the minimum mean square error (MMSE) to the training data. Mathematically, we write

$$\text{MMSE}(d) = \frac{1}{S} \cdot \sum_S (\hat{d} - d_{\text{actual}})^2, \quad (10)$$

where S is the number of samples, \hat{d} is the distance obtained from (9), and d_{actual} is the actual distance from the dataset. Thus, the model in (8) has two parameters that can be tuned: the path loss exponent n and the standard deviation of Xg . Several experiments were conducted on the 12th floor of Hyungnam Memorial Engineering Building at Soongsil University. The area is approximately $30\text{m} \times 20\text{m}$, with the floor plan shown in Section VI.

The procedure to obtain the parameters n and Xg in (9) during the offline phase is described as follows. First, we place a BLE beacon at a specific location in the AoI. Second, a smartphone measures the RSS values from $d_{\text{actual}} = 0.5$ to 1.5m , where the sampling distance interval is 0.5m . As reported in [38], the reliable range of a RedBear beacon is approximately $2.5\sim 4\text{m}$; thus, we only collected the RSS measurements within a ground truth distance of 1.5m . Within this distance, the confidence level for classifying the BLE measurements is more than 95% [5]. Collecting far BLE beacons may not always be efficient because of significant RSS fluctuations. Hence, the best way to deal with signal ambiguity problems among nearby reference ones is to obtain the RSS values within a short distance and to discard the others. With respect to the BLE beacon placement as in Section VI, the results obtained from the measured data on iPhone 12 are given in Table 3. After this stage, the estimated values of n and Xg are given in (9) to obtain an accurate range when converting the RSS reading during the online phase.

TABLE 3. Obtained channel results from a collected dataset on iPhone 12.

| Parameter | Environment class | | |
|----------------|-------------------|--------|--------|
| | LoS | p-NLoS | NLoS |
| $L(d_0)$ [dBm] | -51.58 | -51.98 | -57.63 |
| n | 1.91 | 2.36 | 4.12 |
| Xg | 3.95 | 4.11 | 3.49 |
| MMSE | 2.85 | 3.54 | 4.26 |

Overall, when generating the indoor map for the app, we analyze the relevant RN signal propagation properties and other additional information such as IMU data to regionalize the trace map into different distinct regions and refer to the corresponding functionality of different areas. Coarse regionalization classifies the floor, corridors, rooms, transition areas, and so on, while fine regionalization classifies the selected region into smaller regions based on the probabilistic model. Finally, it yields a map of AoI.

IV. IMU-BASED PDR

In this section, we introduce some basic components of the PDR and the method used to obtain readings from

smartphone-based IMU sensors. From the readings, we learn about the relevant parameters to improve the PDR components to avoid accumulated errors from path deviations and external impacts along the walking path.

A. ACTIVITY RECOGNITION

In this section, we describe the three main user activities.

- *Idle*: Acceleration and barometer readings do not change much.
- *Normal walking*: Acceleration readings stably change with small variance.
- *Go up/down stairs*: In cases involving stairs, the accelerometer signature has large variations on acceleration and barometer readings.

Figure 4 gives examples of the acceleration behaviors of such movement states. We observe that the acceleration patterns are unique in each activity. Furthermore, each floor is divided into several small regions based on RN reading and IMU (acceleration and barometer) patterns. Each region corresponds to a potential motion state. For example, considering the path of a person walking from the exit, through the corridor, and into the room, the path can be divided into several parts such as the first part (transition area), the second part (corridor area with normal walking), and the third part (enter the room and connect with the RN in that room).

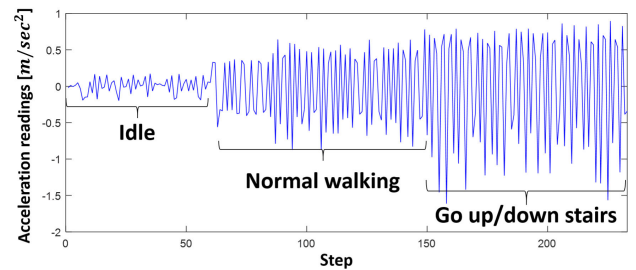


FIGURE 4. Accelerometer signatures corresponding to movement states.

B. PDR-RELATED COMPONENTS

To retrieve the user’s mobility, a typical smartphone-assisted IMU collector reads the corresponding measurements from several sensors such as barometers, accelerometers, gyroscopes, and magnetometers [13]–[16]. With these inputs, the PDR calculates them to track a user along the walking path. Mathematically, the user location $\mathbf{p}_k(x_k, y_k)$ at each step k is calculated as

$$\begin{aligned} x_k &= x_0 + \sum_{m=1}^k \lambda_m \cdot \sin(\alpha_m), \\ y_k &= y_0 + \sum_{m=1}^k \lambda_m \cdot \cos(\alpha_m). \end{aligned} \quad (11)$$

Here, $\mathbf{p}_0(x_0, y_0)$ is the starting location of the tracking path, which may be unknown or given. The parameters λ_k and

α_k are the estimated step length and heading direction angle obtained from the IMU, respectively. Next, we describe the three main functions of a typical PDR-based tracking system.

1) STEP DETECTION AND STEP COUNTING

From [42], the main principle for step detection is the recognition of a repetition of accelerometer readings. In particular, depending on the user's walking style, a step is recorded if the corresponding acceleration measurements form a peak followed by a valley in the time domain. Mathematically, given an accelerometer reading a_t , we define a step as follows.

$$\{a_t \geq a_{upper}, a_{t+\Delta t} \leq a_{lower}, 0 < \Delta t < 1\}. \quad (12)$$

In practice, we set $a_{upper} = 0.3\text{m/sec}^2$ and $a_{lower} = -0.3\text{m/sec}^2$ for normal walking, and $a_{upper} = 0.7\text{m/sec}^2$ and $a_{lower} = -0.7\text{m/sec}^2$ for walking on stairs. Moreover, we assume that a user normally takes two steps per second, so the time spacing between two adjacent accelerometer peaks/valleys must be greater than one-third of the walking step period. In particular, if the walking step period is 1sec, the time spacing between two adjacent accelerometer peaks/valleys must be greater than 1/3sec. Once a step is recognized, the pedometer adds this step to its counters. Using a smartphone pedometer, it is possible to achieve a step counting accuracy above 95% [12].

2) STRIDE LENGTH ESTIMATION

The stride length λ_0 is defined as the distance between two adjacent steps. This value depends on the height, walking style, and speed of the user. Using a given accelerometer signal, calculating the step length can be obtained in several ways, such as the regression [43], Weinberg [44], Kim et al. [45], Scarlett [46], and artificial intelligence models [47]. In this work, we use the following regression model to obtain a suitable step length for the PDR.

$$\lambda = \pi_0 \cdot f + \pi_1 \cdot \text{Var}(\mathbf{a}) + \pi_2, \quad (13)$$

Here, f is the step frequency that can be learned from experiments [5], [31], $\text{Var}(\mathbf{a})$ is the acceleration variance in each step, and π_0, π_1 , and π_2 are the regression coefficients [5].

3) HEADING DIRECTION ESTIMATION

To obtain the direction of the steps as the user walks, we use a smartphone gyroscope and compass in the magnetometer. For each sensor, we have different heading direction information according to its characteristics. The compass-based reading is stable in a long walking path, but can be affected by metal material along the path, while the gyroscope-based reading avoids magnetic fields but suffers from accumulated errors. In our previous works [5], [48], we considered a combination of two sensors in order to achieve a good prediction of the heading direction. In this section, we denote α_g as the newest gyroscope and α_c as the orientation calculated from the accelerometer/magnetometer. The modified heading

direction angle is updated as follows.

$$\alpha_m = b \cdot (\alpha_m + \alpha_g \cdot dt) + (1 - b) \cdot \alpha_c, \quad (14)$$

where dt is the short time interval from the last gyroscope data until the present. The value of parameter b depends on which data are primarily used. For example, $b = 0.98$ means that we want to use the gyroscope data as the heading direction for the PDR. Let N_{num} be the number of footsteps that the user has completed along a given route, then the smartphone measures the traveled distance, as in [48].

$$d_{pdr} = \sum_{k=1}^{N_{num}} \lambda_k. \quad (15)$$

Note that (15) is usually used to check the user's instantaneous trace with the reference RN range for efficient deviation detection. This procedure is described in section V.

C. SMOOTHING

To provide better IMU data for PDR functioning, low pass filters are common used in mobile devices in some forms of weight smoothing [49] or averaging window [17]. In terms of weight smoothing, a weight smoothing parameter ν can be applied on raw IMU data x_i as follows.

$$(\text{New value}) = (\text{Last value}) \cdot (1 - \nu) + x_i \cdot \nu. \quad (16)$$

Here, the last calculated value is added to the most recently collected one x_i according to a weight factor ν . If ν is close to 0, it means the new value does not much change when smoothing. Otherwise, it allows x_i to affect the updated value. For example, in **CoreMotion** [35], the output acceleration values after applying a soft weight smoothing (e.g., $\nu = 0.005$) are identified as raw acceleration readings for motion tracking.

In this section, we applied a method called the simple moving average (SMA) [10] to smooth out data spikes. It works as a rolling average within a given window size. Mathematically, we denote ω as the size of the averaging window, and the SMA does not work until ω values have been collected. Once the collection has been completed, it finds the mean of the most recent ω data values in a stream.

$$(\text{New value}) = \frac{1}{\omega} \cdot \sum_{i=k}^{k+\omega-1} (\text{Last value})_i. \quad (17)$$

The maximum frequency is usually hardware-dependent (e.g., **CoreMotion** can use the supported frequency at 100Hz). The selected window size ω should be chosen to be large enough to smooth out the data, while being not too large to cause delay in response time because of longer calculation. In our paper, with a sampling frequency of 20Hz, we selected the window size $\omega = 3 \sim 5$ for PDR process.

There are various filter options such as Kalman filters and their variants [47], [50], [51] and particle filters [14], [42], which are computational expensive. In this work, note that we did not focus on optimizing these filters. Depending on the applications, we can choose the appropriate filters depending on the design choices.

V. ONLINE TRACKING PROCEDURE

In this section, we show how our proposed tracking app works in a typical indoor multiple-floor environment. A diagram showing how the application works are shown in Fig. 5. A tracking scheme consists of an indoor map (e.g., RN list and deployment map) and a mobile user holding a smartphone (e.g., iPhone 12). The tracking problem is defined as follows. At the m -th step, the iPhone carried by a mobile user periodically collects online RSS readings from N_m near RNs. We denote the RSS reading vector as $\mathbf{r}_m = [r_m^1, \dots, r_m^{N_m}]^T$, where each component of this vector corresponds to the RSS from N_m RNs. Note that the online measurement vector is first pre-filtered to determine the N_m^f RNs that have the strongest RSS readings. The indoor tracking app uses the vector \mathbf{r}_m and the corresponding IMU data to detect the current feasible area (floor, room, transition area) and chooses a suitable channel propagation according to this area. It also updates the user location via the PDR, such as the previous user's estimated location $\mathbf{p}_{m-1}(x_{m-1}, y_{m-1})$, the step length λ_m , and the heading direction α_m to calculate the current user's location $\mathbf{p}_m(x_m, y_m)$.

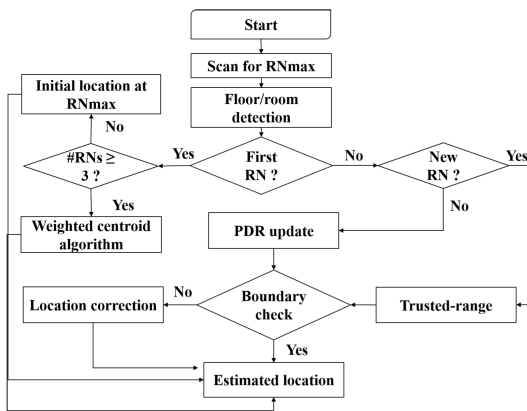


FIGURE 5. Flowchart of our tracking app.

1) INITIALIZATION

First, we assume that the RN database has been saved on the iPhone; thus, our tracking app can easily access related information of any RN inside the AoI. This information includes their locations, and the signal profiles are built during the offline phase. When the user starts the app, he/she can navigate his/her location by exploring the characteristics of the RN space. In our tracking app, if the RNs are of the same type, we seek the strongest RN as a starting location for the tracking path, i.e., $\mathbf{p}_0(x_0, y_0)$. Otherwise, if there are multiple RNs in the area with the same RSS value, we use a weighted centroid algorithm to determine $\mathbf{p}_0(x_0, y_0)$ as

$$\mathbf{p}_0(x_0, y_0) = \sum_{i=1}^{N_b} w_i \cdot \mathbf{a}_i, \tag{18}$$

where N_b is the number of available RNs $\{\mathbf{a}_i, i = 1, \dots, N_b\}$ that appear on the iPhone's scanning list. The weight

factors w_i are assigned according to the strongness level of the RN signals, which are obtained from our previous work [5]. If the RNs are different from each other, we assign priority to the RN that has the shortest calculated distance from its signal to the iPhone, according to Section III.

2) LOCATION ESTIMATION

As the user is moving, the tracking app exploits the PDR method based on smartphone-IMU measurements to perform the localization. In particular, the current location of the smartphone $\mathbf{p}_m(x_m, y_m)$ is calculated as

$$\begin{aligned} \mathbf{p}_m(x_m, y_m) &= \mathbf{p}_{m-1}(x_{m-1}, y_{m-1}) + \Delta \mathbf{p}_m \\ &= \begin{bmatrix} x_{m-1} \\ y_{m-1} \end{bmatrix} + \lambda_m \begin{bmatrix} \sin(\alpha_m) \\ \cos(\alpha_m) \end{bmatrix}. \end{aligned} \tag{19}$$

Here, λ_m is the estimated stride length. The value α_m is the estimated heading direction at point $\mathbf{p}_m(x_m, y_m)$. Both λ and α_m were obtained from Section III.

3) LOCATION CORRECTION

At every step, the smartphone uses the tracking app to keep track of its location. For each tracking update, we check whether the location estimate properly follows the desired path. Thus, the aim of the location calibration module is to determine if the user is still within the reliable range of the strongest RN in the scan list. The module generates instructions according to the following two rules:

- (i) Without loss of generality, we denote the strongest RN location in the scan list as \mathbf{a}_1 and its estimated distance from (9) as \hat{d}_1 . If $\hat{d}_1 \leq d_{lim}$ ($d_{lim} = 5\text{m}$ for Wi-Fi AP and 1.5m for BLE beacon), the tracking app sets \mathbf{a}_1 as the corrected location of the current user.
- (ii) Otherwise, we consider the dissimilarities among the estimated ranges (9) of N_m^f nearby RNs and the PDR walking distance (15) as

$$\varphi_m = \sum_{i=1}^{N_b} w_i \cdot |\hat{d}_i^m - d_{pdr}^m|. \tag{20}$$

To provide correct walking hints at the correct locations, when the dissimilarity veers off the desired path of the thickness ϵ (i.e., $\varphi_m \geq \epsilon$), the module attempts to match $P_m(x_m, y_m)$ to the intersection of the circle $C(\mathbf{a}_1, \hat{d}_1)$ made by \mathbf{a}_1 with radius \hat{d}_1 and the line segment made by $P_{m-1}(x_{m-1}, y_{m-1})$ and $P_m(x_m, y_m)$ as

$$P_m(x_m, y_m) = C(\mathbf{a}_1, \hat{d}_1) \cap \overrightarrow{P_{m-1}P_m}. \tag{21}$$

4) BOUNDARY CHECK

Any sub-region in the AoI can be bounded by a rectangle represented by the four corners. As illustrated in Fig. 6 (a), an office room is marked on the map by four red squares. For the corridor case, it becomes simpler by saving the corridor width, the starting point, and the ending point of the corridor. Therefore, only a finite number of markers are stored in the database, making data storage simpler and easier.

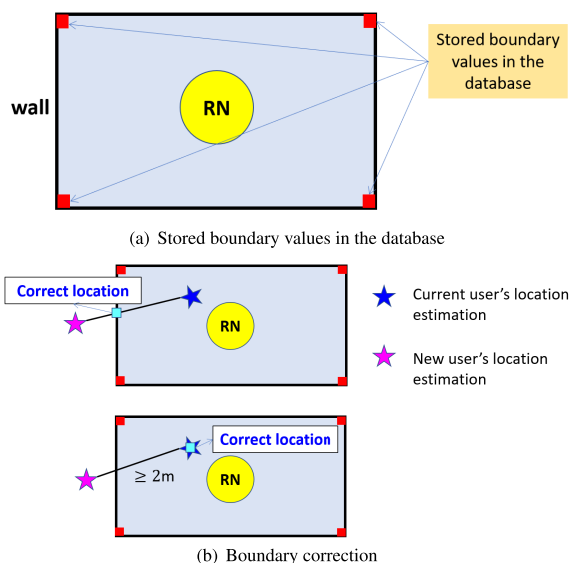


FIGURE 6. Boundary check and correction.

To present smooth visuals for the user experience, we rely on the following two principles so that the calculated jump distance between two estimated steps is not too large compared to the user step length. (i) The currently estimated step (e.g., \mathbf{p}_m) is inside the closed rectangle, and the next estimated step \mathbf{p}_{m+1} indicates that it is out of the rectangle. Then, the correction point is the intersection of the adjacent rectangle edge and the line connecting the two points \mathbf{p}_m and \mathbf{p}_{m+1} , as shown in Fig. 6 (b) (first image). (ii) In cases where the distance between the current estimated location and the next one is too large, for example, more than 2m, we understand that there is a large fluctuation in the input signals, and the updated point remains at the current location until new RN signals are received, as shown in Fig. 6 (b) (second image).

In summary, we adopted a fusion scheme between the probabilistic approach for RN signals and the smartphone-based PDR system to perform self-localization and tracking. To investigate the potential causes of limited localization accuracy, we performed various practical experiments and revisited the characteristics of the RN localization scheme. First, floor level detection can be improved by employing an RSS profile-based method and mixed data of barometer/acceleration from the smartphone-IMU. Taking advantage of the characteristics of specific regions, we used different signal readings to achieve a precise level for each area detection. Second, RN signals are reliable over short communication ranges, and distant RN locations may lead to large location errors. Biased RSS measurements caused by p-NLoS/NLoS effects can be controlled using a prior probability model during the training phase. Third, the smartphone-IMU extracted user mobility information to reduce tedious site survey efforts, while improving tracking accuracy. In addition, our localization app is more friendly because it does not expect the user to start at a specific location for tracking.

VI. EXPERIMENT RESULTS

We conducted several experiments in a typical office building to demonstrate the effectiveness of our tracking scheme. We built a tracking app on an iPhone 12 with iOS 14.7 and 128 GB storage. Throughout this section, we make the following assumptions.

- (i) First, in order to scan nearby Wi-Fi AP/BLE beacons, the iPhone needs to turn on Wi-Fi/Bluetooth function in the setting manner. All Wi-Fi APs and BLE beacons used in the test area were aggregated into an RN list. The ratio of the number of nodes between Wi-Fi and BLE depends on the choice of design. For the readers' convenience, we mark the letter "B" for BLE and "W" for Wi-Fi for simplicity in the RN name.
- (ii) Second, assuming that we take advantage of the existing Wi-Fi infrastructure in the building, BLE beacons just need to be placed in some locations to support room detection and to corroborate the tracking algorithm. In this case, the number of BLE beacons should be sufficient to cover the AoI and not conflict with the current Wi-Fi communication range. In terms of BLE-only deployment, the AoI is covered by a reasonable BLE beacon density; that is, each BLE beacon only covers a separate area without causing interference and ensuring appropriate coverage for the area between two adjacent beacons. We follow the BLE deployment principles in [5] for planning BLE beacons over the AoI.
- (iii) Third, we also label the RN list using two categories: "office room" and "office corridor" by using the MAC address for Wi-Fi APs and the UUID, major value, and minor value for BLE beacon. For example, the UUID parameter gives us a unique number that allows the identification of the beacon manufacturer or owner, while the major and minor values can be assigned as a certain floor number and a room number, respectively. During the online phase, only beacons with the same category can join the localization algorithm described in Section V.
- (iv) The tester holds a mobile iPhone in hand and walks through the AoI. The screen of the iPhone is kept facing up when the user is using it. Depending on the user's activities, the IMU data are recorded with either short intervals (e.g., go up/downstairs) or long intervals (e.g., idle or normal walking).

The series of our experiments is described as follows. We conducted our experiments on three floors (11th, 12th, and 13th floors) of Hyungnam Memorial Engineering Building at Soongsil University. The size of each floor in this building is approximately 20m × 40m. The distance between two consecutive floors is approximately 3.5m. The staircase connecting the two floors consists of two segments, each of which has 11 stairs. On each floor, there are two types of elevators at both ends of the corridor: the single elevator can reach all floors and the double elevators only goes to either odd or even floors. Currently, there are two Alcatel Lucent IAP-305 2×/3× 11ac Wi-Fi APs deployed over each

floor. A detailed description of the AP's specifications can be found in [52]. For a hybrid tracking system, we used RedBear beacons [37] to support tracking. All of them were placed in either office or office corridor. The beacons were attached to the ceiling of the floor to avoid disturbances when people passed. The iPhone 12 carried by a user collects both IMU data and RSS readings from detectable Wi-Fi AP/RedBear beacons simultaneously, and it then inputs them to the tracking app. The estimated steps along the tracking path were also recorded on the iPhone storage for further analysis. Because the three floors have the same structure, the number of BLE beacons placed on each floor is the same. Regarding the localization metric, we defined the accuracy rate (AR) to evaluate the localization performance. Given a localization error bound ϵ , it can be obtained by the percentage of the number of estimated points that fall inside the path of thickness ϵ .

$$AR(\epsilon) = \frac{\#\{P_m(x_m, y_m) : P_m \subset \mathcal{P}(\epsilon)\}}{N_{step}} \times 100\%. \quad (22)$$

Here, N_{step} is the actual number of steps in which the user has finished the desired path $\mathcal{P}(\epsilon)$ with a thickness ϵ .

Before going through the experimental results, we made a few notes about the visual representation of our app. To plot a point in 3D space and update its next point, we must represent the ground truth location on the floor where the user is currently located, and represent the height of the floor. Usually, we only provide indoor floor plans in 2D space, which show specific details of the relationships between rooms and other areas on each floor. Here, it was not easy for us to express smooth 3D tracking results, while satisfying the above conditions. In the following step, we present the calculated user's location on the floor under the 2D map. When the user detects that it enters the transition area, the app plots the location in terms of the vertical direction.

To demonstrate the performance of our tracking app, several experiments were conducted in the building, as shown in Fig. 7. The user carrying the iPhone 12 collects the RSS and IMU measurements at every step, and walks at a normal speed along the path. This figure shows the demo tracking results of the proposed work in two cases: (a) floor detection with RN RSS-based profile and (b) floor detection with barometer readings in the floor transition area. The grid spacing of the map display was 1m. In this figure, we plot the desired path in green. The thickness of this path indicates a given localization error bound (e.g., $\epsilon = 2m$). The RN locations are indicated by a bright yellow circle. The starting and ending points of the actual path are plotted as star and square shapes, respectively. In Track 1, we observe that when the user moves on the corridor of a certain floor, the tracking results are quite good. Then, when the user reaches the transition staircase area, it creates a large jump in the estimation location. This is clearly shown in the transition from the 12th floor to the 13th floor because this area is not covered by any RN signal, the user's location is updated only when the phone connects to the next RN. However, in Track 2, exploiting barometer readings helps the app to track vertically within



(a) Track 1: Floor detection with RSS-based profile.



(b) Track 2: Floor detection with barometer readings.

FIGURE 7. Demonstration of our proposed tracking scheme in a three-floor scenario.

the transition area. The visual representation of the estimated steps is a refined state along with the user's movements. From the results, we observe that the user would have a better 3D experience if we combine the two schemes on the AoI. In particular, we apply a floor detection scheme with an RN RSS-based profile within corridor/room regions and floor detection with barometer readings in the floor transition area.

In Fig. 8, we consider a practical route when a user walks in a sequence including several complex areas in the building.

- 11th floor (F11) activities: Enter a door on the F11 >> move along corridor F11 >> enter room 1102 >> exit room 1102 and continue moving on corridor F11 until the user meets the stairs to go upstairs to the next floor. On this floor, we evaluate the tracking results when the user walks from the office corridor to the office room, and vice versa.
- 12th floor (F12) activities: Enter a corridor >> enter room 1202 and exit the room >> enter room 1203 and out >> back on F12 corridor and take the single elevator to go up to the next floor. On this floor, we observe the performance when the user enters two adjacent rooms, and we check whether the current room detection is correct.

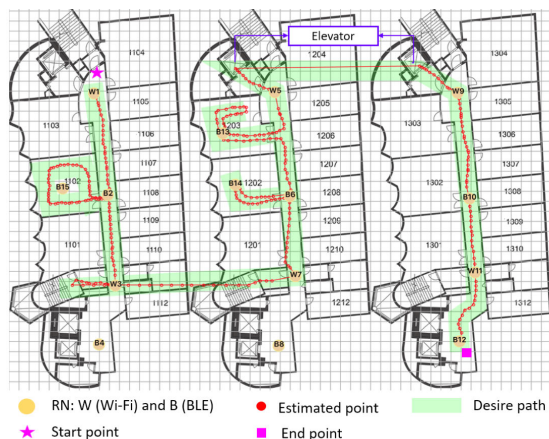


FIGURE 8. More demonstration of the proposed tracking scheme in another three-floor scenario.

- 13th floor (F13) activities: Exit the elevator \gg walk along the F13 corridor to the end.

Note that the user’s walking turns are different among office rooms because there are different numbers of obstacles (e.g., tables and bookshelves) and the distribution of furniture is heterogeneous among rooms. All regions on the map were recognized. We noticed that users rarely get close to the walls and corners of an office room. With the strong signals of the RN placed in the middle of the room, the tracking results obtained are more accurate. With respect to the corridor, because the corridor in the AoI is quite straight, the localization performance has less error than in other areas. For the joint area of the corridor and the room, when the current user is in the room and receives a strong RN signal in the corridor, the app automatically updates the user’s locations based on the correction function in Section III. For the transition area between two floors, we observe that the calculations of floor height and the direction based on IMU readings are sufficient to identify this area correctly, but the tracking results are not as good as in other regions. This is because the use of IMU data for long travel distances can lead to a common problem of veering off the walking path. The situation is terminated only when the smartphone finds a new RN, so it updates itself according to the RN information.

Figure 9 shows a comparison of the average AR versus error bound with different tracking situations: mixed Wi-Fi/BLE scheme with RSS profile floor detection [5] called as “conventional,” proposed scheme with BLE-only, proposed work with mixed staircase-elevator, and proposed work with staircase-only. For each route, a user walks several times on the same route during a day and evaluates the average AR values of path estimation. During the user’s walk, there are people who pass by as well as other obstacles such as bookshelves and metal doors. Although we can utilize smartphone-IMU and a single Wi-Fi AP when the Wi-Fi AP density is low, it is still applicable for tracking with BLE only. Note that when we replace existing Wi-Fi APs with BLE beacons, the current BLE beacon density should be the same

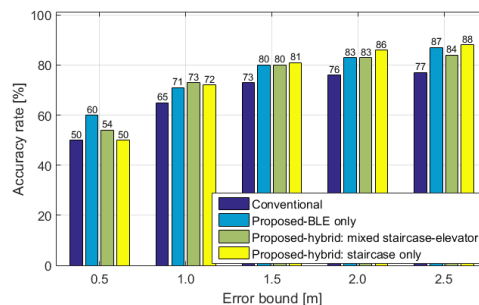


FIGURE 9. Accuracy rate versus error bound.

as that of Wi-Fi, or slightly denser, so it has a low or comparable deployment cost. In this figure, in the office corridor, the number of deployed BLE beacons is the same as the old Wi-Fi AP density. In this case, the average spacing between two BLE beacons is about 10m in the corridor and 6~7m in the room. First, we observe that the hybrid scheme in [5] provides good estimated results only when the user moves on a particular floor. When the user exits the floor and enters the transition zone, its location is no longer updated until he/she moves to the next floor and uses the RNs for location correction. It can be seen that this problem has been completely overcome by our proposed solution. For visualization, as the user ascends/descends the stairs, the corresponding IMU readings provide an additional source to localize the user at every step in this region. Second, the proposed mixed Wi-Fi/BLE achieves slightly better accuracy than the one with BLE only. Tracking results in a simple path (staircase-only) are better than those in the complex path (including the mixed staircase-elevator). Overall, the results show the benefits of the assisted smartphone-IMU sensors and demonstrate the practical efficacy of the proposed scheme.

Throughout the experiment results, we observe that our conventional work [5] achieves reasonable accuracy and a fair complexity level when tracking the user on a particular floor, the user does not spend a lot of time in the transition area, and the user does not perform complicated movements. Therefore, this tracking scheme can be effectively applied for these situations. When we are continuously tracking the user over several floors in a row and using a complex set of sensors and additional information from the RNs in the AoI, the calculation of the user’s location must be consecutive to create a smooth representation of calculated location points on the app as well as representing the movements among different regions. As explained Section I, instead of focusing much on the accuracy improvement gained by tracking algorithms, we are more interested in having a comprehensive understanding of adaptation effects. By conducting several experiments in a certain area, we evaluate the accuracy and the effectiveness of the adaption methods.

VII. CONCLUSION AND DISCUSSION

In this paper, we present a simple indoor multiple-floor tracking scheme based on hybrid Wi-Fi/BLE for smartphones. We also built an iOS app to capture, visualize, and display

user mobility, which is measured by smartphones using the hybrid IMU-RSS signal information in a coordinated manner. In particular, by performing several practical experiments, we aim to obtain a comprehensive understanding of indoor radio maps, while validating the accuracy and effectiveness of the smoothing and adaptation effects. The final tracking system has a lightweight database, is capable of identifying user location by region, and selects an appropriate signal-to-location model. Experimental results show that the proposed tracking system achieves a good performance, while enabling adaptation with/without currently fixed infrastructure. Thus, it is beyond the achievements of previous approaches.

By performing this study, we identify several practical issues that need to be carefully addressed when integrating our tracking scheme with other hybrid systems. First, localization accuracy does not always serve as the sole criterion to prove that a tracking system is effective. In addition, an effective system should be stable over a long time and energy aware without much human effort by using energy more efficiently or collecting a minimal number of measurements for localization purposes. Although the proposed scheme works well inside a typical medium-size building, many experiments need to be carried out in large spaces (e.g., malls, subway stations, and airports). Second, even current smartphones can effectively track user mobility (e.g., detect whether the smartphone is moving or if it is context-aware of the user path), it can still lead to bad estimated results over long travel distances owing to uneven inconsistency among different dimensions of hybrid data (e.g., Wi-Fi/BLE observation and IMU data). To overcome this problem, it should obtain a sufficient number of reference traces and control RSS quality along a path when the user is walking.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2020R1A2C2010006). (*Thu L. N. Nguyen and Tuan D. Vy contributed equally to this work.*)

REFERENCES

- [1] F. B. Elbahhar and A. Rivenq, Eds., *New Approach of Indoor and Outdoor Localization*. Rijeka, Croatia: InTech, 2012, Sec. 1.
- [2] R. Elbakly, H. Aly, and M. Youssef, "TrueStory: Accurate and robust RF-based floor estimation for challenging indoor environments," *IEEE Sensors J.*, vol. 18, no. 24, pp. 10115–10124, Dec. 2018.
- [3] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Proc. 14th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2015, pp. 178–189.
- [4] H. Shin, Y. Chon, Y. Kim, and H. Cha, "A participatory service platform for indoor location-based services," *IEEE Pervas. Comput.*, vol. 14, no. 1, pp. 62–69, Jan. 2015.
- [5] T. D. Vy, T. L. N. Nguyen, and Y. Shin, "A precise tracking algorithm using PDR and Wi-Fi/iBeacon corrections for smartphones," *IEEE Access*, vol. 9, pp. 49522–49536, 2021.
- [6] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.
- [7] N. Kim and Y. Kim, "A novel RSS-ratio position estimation scheme for Wi-Fi networks," in *Proc. Int. Conf. Elect., Electron. Mechatron. (ICEEM)*, vol. 34, Dec. 2015, pp. 61–64.
- [8] I. H. Alshami, N. A. Ahmad, and S. Sahibuddin, "A light WLAN radio map for floor detection in multi-floor environment localization," in *Proc. 9th Malaysian Softw. Eng. Conf. (MySEC)*, Dec. 2015, pp. 135–139.
- [9] J. Zuo, S. Liu, H. Xia, and Y. Qiao, "Multi-phase fingerprint map based on interpolation for indoor localization using iBeacons," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3351–3359, Jan. 2018.
- [10] Z. Chen, Q. Zhu, and Y. C. Soh, "Smartphone inertial sensor-based indoor localization and tracking with iBeacon corrections," *IEEE Trans. Ind. Informat.*, vol. 12, no. 4, pp. 1540–1549, Aug. 2016.
- [11] P. Spachos and K. N. Plataniotis, "BLE beacons for indoor positioning at an interactive IoT-based smart museum," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3483–3493, Sep. 2020.
- [12] H. Zou, Z. Chen, H. Jiang, L. Xie, and C. Spanos, "Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon," in *Proc. IEEE Int. Symp. Inertial Sensors Syst.*, Mar. 2017, pp. 1–4.
- [13] H.-H. Hsu, J.-K. Chang, W.-J. Peng, T. K. Shih, T.-W. Pai, and K. L. Man, "Indoor localization and navigation using smartphone sensory data," *Ann. Oper. Res.*, vol. 265, no. 2, pp. 187–204, Jan. 2017.
- [14] J. Liu, R. Chen, L. Pei, R. Guinness, and H. Kuusniemi, "A hybrid smartphone indoor positioning solution for mobile LBS," *Sensors*, vol. 12, no. 12, pp. 17208–17233, 2012.
- [15] A. Poullose, O. S. Eyobu, and D. S. Han, "A combined PDR and Wi-Fi trilateration algorithm for indoor localization," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIC)*, Feb. 2019, pp. 72–77.
- [16] Q. Lu, X. Liao, S. Xu, and W. Zhu, "A hybrid indoor positioning algorithm based on WiFi fingerprinting and pedestrian dead reckoning," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6.
- [17] A. Mackey, P. Spachos, L. Song, and K. N. Plataniotis, "Improving BLE beacon proximity estimation accuracy through Bayesian filtering," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3160–3169, Apr. 2020.
- [18] Y. Li, Y. Zhuang, H. Lan, X. Niu, and N. El-Sheimy, "A profile-matching method for wireless positioning," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2514–2517, Dec. 2016.
- [19] L. Sun, Z. Zheng, T. He, and F. Li, "Multifloor Wi-Fi localization system with floor identification," *Int'l. Jour. Distribute Sensor Net.*, vol. 11, no. 7, pp. 1–8, Jul. 2015.
- [20] A. Razavi, M. Valkama, and E. Lohan, "K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–7.
- [21] 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.
- [22] Y. Zhang, D. Li, and Y. Wang, "An indoor passive positioning method using CSI fingerprint based on AdaBoost," *IEEE Sensors J.*, vol. 19, no. 14, pp. 5792–5800, Jul. 2019.
- [23] C. Zhou, J. Liu, M. Sheng, Y. Zheng, and J. Li, "Exploiting fingerprint correlation for fingerprint-based indoor localization: A deep learning based approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5762–5774, Jun. 2021.
- [24] Y. Li, Z. Gao, Z. He, P. Zhang, R. Chen, and N. El-Sheimy, "Multi-sensor multi-floor 3D localization with robust floor detection," *IEEE Access*, vol. 6, pp. 76689–76699, 2018.
- [25] M. Zhou, B. Wang, Z. Tian, J. Wang, and Q. Zhang, "A case study of cross-floor localization system using hybrid wireless sensing," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
- [26] W. Jaworski, P. Wilk, P. Zborowski, W. Chmielowiec, A. Y. Lee, and A. Kumar, "Real-time 3D indoor localization," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–8.
- [27] J. C. A. Herrera, P. G. Plöger, A. Hinkenjann, J. Maiero, M. Flores, and A. Ramos, "Pedestrian indoor positioning using smartphone multi-sensing, radio beacons, user positions probability map and IndoorOSM floor plan representation," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2014, pp. 636–645.
- [28] C.-C. Lo, T.-H. Chiang, T.-K. Lee, L.-J. Chen, and Y.-C. Tseng, "Wireless location tracking by a sensor-assisted particle filter and floor plans in a 2.5-D space," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

- [29] Apple Developer. *Core Bluetooth*. Accessed: Sep. 12, 2020. [Online]. Available: <https://developer.apple.com/documentation/corebluetooth>
- [30] L. Mengual, O. Marbán, and S. Eibe, "Clustering-based location in wireless networks," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6165–6175, 2010.
- [31] T. D. Vy, T. L. N. Nguyen, and Y. Shin, "Inertial sensor-based indoor pedestrian localization for iPhones," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2019, pp. 200–203.
- [32] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 689–705, Jun. 2007.
- [33] Apple Developer. *CNCopyCurrentNetworkInfo*. Accessed: Sep. 12, 2020. [Online]. Available: <https://developer.apple.com/documentation/systemconfiguration/1614126-cncopycurrentnetworkinfo?language=occ>
- [34] Apple Developer. *iOS Wi-Fi Management APIs*. Accessed: Sep. 12, 2020. [Online]. Available: https://developer.apple.com/library/archive/qa/qa1942/_index.html
- [35] Apple Developer. *Core Motion*. Accessed: Nov. 12, 2020. [Online]. Available: <https://developer.apple.com/documentation/coremotion>
- [36] Apple Developer. *Core Location*. Accessed: Nov. 12, 2020. [Online]. Available: <https://developer.apple.com/documentation/corelocation>
- [37] FCC ID Database. *Beacon B1 Instruction*. Accessed: Oct. 20, 2020. [Online]. Available: <https://fccid.io/2ABXJ-B-B1/User-Manual/User-Manual-2223286.pdf>
- [38] T. D. Vy and Y. Shin, "iBeacon indoor localization using trusted-ranges model," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 1, pp. 1–13, Jan. 2019.
- [39] F. Carpi, L. Davoli, M. Martalo, A. Cilfone, Y. Yu, Y. Wang, and G. Ferrari, "RSSI-based methods for LOS/NLOS channel identification in indoor scenarios," in *Proc. 16th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2019, pp. 171–175.
- [40] S. M. Kay, *Fundamentals of Statistical Signal Processing, Estimation Theory*, vol. 1, Upper Saddle River, NJ, USA: Prentice-Hall, 1993, Ch. 6
- [41] H. Ye, W. Yang, Y. Yao, T. Gu, and Z. Huang, "BTrack: Using barometer for energy efficient location tracking on mountain roads," *IEEE Access*, vol. 6, pp. 66998–67009, 2018.
- [42] S. Traini, L. Sciuillo, A. Trotta, and M. Di Felice, "Practical indoor localization via smartphone sensor data fusion techniques: A performance study," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–7.
- [43] M. Kourogi, T. Ishikawa, and T. Kurata, "A method of pedestrian dead reckoning using action recognition," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, May 2010, pp. 85–89.
- [44] H. Weinberg, "AN-602: Using the ADXL202 in pedometer and personal navigation applications," *Analog Devices Tech. Rep.*, 2002. [Online]. Available: http://www.analog.com/static/imported-files/application_notes/513772624AN602.pdf
- [45] 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. 3, nos. 1–2, pp. 273–279, 2004.
- [46] J. Scarlett. *Enhancing the Performance of Pedometers Using a Single Accelerometer*. Accessed: Sep. 12, 2020. [Online]. Available: <https://www.analog.com/en/analog-dialogue/articles/enhancing-pedometers-using-single-accelerometer.html>
- [47] J. Hannink, T. Kautz, C. F. Pasluosta, J. Barth, S. Schülein, K.-G. Gaßmann, J. Klucken, and B. M. Eskofier, "Mobile stride length estimation with deep convolutional neural networks," in *Proc. IEEE J. Biomed. Health Inform.*, vol. 22, no. 2, pp. 354–362, Mar. 2018, doi: [10.1109/JBHI.2017.2679486](https://doi.org/10.1109/JBHI.2017.2679486).
- [48] T. D. Vy, T. L. N. Nguyen, and Y. Shin, "A smartphone indoor localization using inertial sensors and single Wi-Fi access point," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–7.
- [49] G. Milete and A. Stroud, Eds., *Professional Android Sensor Programming*. Hoboken, NJ, USA: Wiley, 2012, Ch. 6.
- [50] Q. Wang, H. Luo, L. Ye, A. Men, F. Zhao, Y. Huang, and C. Ou, "Pedestrian heading estimation based on spatial transformer networks and hierarchical LSTM," *IEEE Access*, vol. 7, pp. 162309–162322, 2019.
- [51] B. Koo, S. Lee, M. Lee, D. Lee, S. Lee, and S. Kim, "PDR/fingerprinting fusion indoor location tracking using RSS recovery and clustering," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, Busan, South Korea, Oct. 2014, pp. 699–704.
- [52] Alcatel Lucent Enterprise. *Alcatel-Lucent OmniAccess 320 Series Access Points: Service Multi-Tenant Network Management*. Accessed: Jun. 20, 2021. [Online]. Available: <https://www.al-enterprise.com/-/media/assets/internet/documents/omniaaccess-ap320-series-access-points-datasheet-en.pdf>



THU L. N. NGUYEN received the B.S. degree in mathematics and computer science from the Vietnam National University Ho Chi Minh City-University of Science, Ho Chi Minh City, Vietnam, in 2011, and the M.S. and Ph.D. degrees in information and telecommunication engineering from Soongsil University, Seoul, South Korea, in 2014 and 2019, respectively. She is currently working with the Communication and Intelligent Laboratory, Soongsil University, as a Postdoctoral Researcher. Her research interests include wireless communication, indoor/outdoor-based localization, and Bluetooth low energy.



TUAN D. VY (Student Member, IEEE) received the B.E. degree in mathematics and computer science from the Vietnam National University Ho Chi Minh City-University of Science, Ho Chi Minh City, Vietnam, in 2011, and the M.S. and Ph.D. degrees in information and telecommunication engineering from Soongsil University, South Korea, in 2016 and 2021, respectively. His research interests include Bluetooth low energy, indoor localization, and location-based services.



KWAN-SOO KIM (Student Member, IEEE) received the B.S. degree in computer science from Soongsil University, Seoul, South Korea, in 2020, where he is currently pursuing the M.S. degree in information and telecommunication engineering. His research interests include indoor localization, deep learning, and pedestrian dead reckoning.



CHENXIANG LIN received the B.S. degree in computer science and engineering from Korea University, in 2021. Currently, he is pursuing the M.S. degree in information and telecommunication engineering with Soongsil University, Seoul, South Korea. His research interests include indoor localization and deep learning.



YOAN SHIN (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, in 1992. From 1992 to 1994, he was with Microelectronics and Computer Technology Corporation (MCC), Austin, TX, USA, as a member of the technical staff. Since 1994, he has been with the School of Electronic Engineering, Soongsil University, Seoul, and is currently a Professor. From September 2009 to August 2010, he was a Visiting Professor with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada. His research interests include localization and mobile communications. He has served as an/a Organizing/Technical Committee Member of various prominent international conferences, including VTC 2003-Spring, ISITA 2006, ISPLC 2008, APCC 2008, ISIT 2009, APWCS 2009, APWCS 2010, APCC 2010, ICTC 2010, APWCS 2011, ICTC 2011, ISAP 2011, APWCS 2012, APCC 2012, and WCNC 2020. In particular, he was the Technical Program Committee Chair of APWCS 2013, the General Co-Chair of APWCS 2014 and APWCS 2015, and the General Vice Chair of ICC 2022. He is the President-Elect (President of the Year 2022) of the Korean Institute of Communications and Information Sciences (KICS).

• • •