Wireless Netw (2016) 22:175–191
DOI 10.1007/s11276-015-0964-6

Construction of indoor floor plan and localization

Ahmad Abadleh[1] · Sangyup Han[1] · Soon Joo Hyun[1] · Ben Lee[2] · Myungchul Kim[1]

Published online: 16 May 2015
© Springer Science+Business Media New York 2015

Abstract Indoor positioning and tracking services are garnering more attention. Recently, several state-of-the-art localization techniques have been proposed that use radio maps or the sensors readily available on smartphones. This paper presents a localization system called construction of indoor floor plan and localization (CALL), which is based on a floor plan, access points (APs), and smartphone sensors. The public access points provide a number of reference points that can be used to build the floor plan automatically, and acquire the user initial position. Moreover, The APs are used to calibrate and adjust the user's position, distance, and direction in real time. Smartphone sensors are used to detect the number of steps and the direction. The proposed method is implemented on a smartphone and tested in real indoor environments. Our experiment with CALL demonstrates that using a static floor plan instead of radio map will avoid the costly database updates and searches that are usually required in other approaches due to signal attenuation. It shows that CALL outperforms the existing works in terms of accuracy and effectiveness in indoor localization.

Keywords Reference point · Maximum average received signal strength · Step counting · Floor plan

1 Introduction

Indoor localization and tracking using smartphones have been widely investigated and their importance is continually increasing as a result of the numerous applications that require indoor positioning systems, such as healthcare, child tracking, mobile accessibility for the blind, advertisements in indoor environments, and so on [1].

Many techniques have been developed in this track of indoor positioning services that provide localization and user trajectories. The most popular technique is fingerprinting approaches, which use wireless technologies such as Wi-Fi, Bluetooth, or other radios [2, 3]. Other techniques provide locations based on radio signals, such as Time of Arrival (TOA) [4, 5], trilateration [6], and Angle of Arrival (AOA) [7]. In radio signal-based localization techniques, such as TOA and AOA, at least three different signals, and sometimes additional hardware, are required in order to obtain the user's location. In addition, sensor-based techniques that rely on smartphone sensors suffer from noises and can therefore lead to inaccurate position estimations.

Fingerprinting-based techniques compare the measured radio signal with a radio map of a given area in order to locate the best match and generate the predicted location. This method has become the most popular localization technique for several reasons. First, additional hardware is not required. Second, the received signal strength (RSS) is

✉ Myungchul Kim
mck@kaist.ac.kr

Ahmad Abadleh
abadleh@kaist.ac.kr

Sangyup Han
ilu8318@kaist.ac.kr

Soon Joo Hyun
sjhyun@kaist.ac.kr

[1] Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Republic of Korea

[2] School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97330, USA

widely used in infrastructure networks. Third, it provides accuracy up to approximately 2 m [8]. However, this technique suffers from some significant problems that remain to be resolved. One of the problems is known to be the recalibration problem in which indoor environments must be resurveyed regularly due to signal attenuation and/or scattering when any changes occur when installing or removing a new access point (AP) or rearranging the furniture (which potentially interferes with signals). Another problem is the processing overhead required to locate the large radio map database. Complex search algorithms have been employed in an attempt to reduce the overhead and to obtain the closest position.

Another important issue in localization and tracking techniques is distance estimation. This is a critical issue because it directly affects the system accuracy. Some approaches use the characteristics of the radio signals, such as the path loss exponent, to estimate the distance [9]. However, this technique suffers from the signal multipath problem. Other approaches, such as that described in [1], use smartphone accelerometers to count the number of steps to calculate the distance. Despite this technique provides more accurate distance estimations, it suffers from miscounting of steps due to irrelevant smartphone movements.

The pervasiveness of smartphones that are equipped with numerous sensors, such as accelerometer and compass, enables to acquire location tracking of mobile users. Many smartphone-based localization techniques have been developed by using various smartphone sensors. LifeMap uses a global positioning system (GPS), accelerometer, and digital compass to track users [1]. Another technique called SMART uses Wi-Fi signals and smartphone sensors to perform localizations and map constructions [10]. Also, UnLoc [11] uses a number of landmarks with smartphone sensors to estimate the user position.

This paper proposes a hybrid technique called construction of indoor floor plan and localization (CALL). It builds the floor plan automatically using smartphone sensors and public Wi-Fi APs. Then the public Wi-Fi APs are used to detect the initial position of the user. Furthermore, the public APs are used to calibrate and adjust the user position during the tracking in real time. Smartphone sensors such as accelerometer and digital compass are used to detect the steps and the direction of the user. CALL relies on an assumption that the public APs are attached to the ceiling of the corridors with well-distributed topological locations, and share the same service set identifier (SSID).

The significant contributions of this work include the following five points:

- A technique to automatically build the floor plan by leveraging the smartphone sensors and Wi-Fi RSS.
- A new technique to dynamically detect any change in the environment, such as installing new AP;
- A novel technique to adjust the user position according to the reference points given by APs;
- A more accurate distance and direction estimation using a smartphone accelerometer and orientation sensor with respect to the floor plan database;
- Implementation of the proposed method (CALL) in an Android-based smartphone and experimental analysis in a real environment for the validation of the feasibility.

The rest of the paper is organized as follows: Sect. 2 discusses the related work. Section 3 describes the proposed system design and architecture. Section 4 presents our evaluation. Finally, Sect. 5 concludes this paper and discusses the future work.

2 Related work

The related work on indoor localization and tracking can be categorized into three foundational techniques: location-based techniques, tracking-based techniques, and map construction techniques.

2.1 Location-based techniques

Localization techniques use radio signals, smartphone sensors, and sound acoustics. In these approaches, the location is determined using a distance or angle estimation. Trilateration is a technique that requires the distances of three reference nodes to obtain the position as the intersection between the three circles formed [6]. In contrast, AOA requires angle estimations between the sender and receiver in order to obtain the location [7, 12]; this technique requires at least two reference points. The fingerprinting technique has two phases: an offline phase that collects a large number of radio signals to build a radio map and an online phase that measures the radio signals and then compares these with those in the radio map using several search algorithms to obtain the closest position.

RADAR is one of the earliest fingerprinting techniques, and it collects Wi-Fi signal strengths during the offline phase for 70 locations in four directions [13]. These signal strengths are then compared with the radio map to generate the best match using the K-nearest neighbors. The authors in [8] attempted to reduce the search process overhead in fingerprinting using a transfer function to obtain the constructed states from the offline radio map. Then, during the online phase, the same transfer function was used to construct the online reconstructed states, which was compared with the offline reconstructed states using a probabilistic weighting function.

In [14], an FM radio was used and its feasibility in indoor positioning was examined. Using a local FM transmitter, e.g., MP3/iPod transmitter, the accuracy of the approach improved, but the accuracy dropped significantly when a global FM transmitter (e.g., operating in global frequency range of 8–108 MHz) was used. The indoor positioning system discussed in [2] used a magnetic field as a location reference instead of radio signals. They built a magnetic map as an array of e-compasses. Then, the nearest neighborhood algorithm was used to obtain the position. A significant limitation of this approach is the need for additional hardware in order to build the magnetic map during the offline phase.

In [15], an offline and online algorithms for sensor networks were proposed to detect the position of the sensors based on RSS data. The system collects the RSS values with their corresponded location, and then the position is detected by calculating the distance between the node and the RSS collector node. In the online algorithm, the system groups the collected online RSS values based on a sliding window. Then it detects the location of the sensor if it collects enough number of credits, which is given to the sensor in each round. A localization algorithm based on Ultra Wide Band (UWB) was proposed in [16]. The algorithm uses some known coordinates anchor nodes to detect the target nodes locations based on path loss and Maximum Likelihood Estimator (MLE).

The significant limitations of these location-based approaches are caused by signal attenuation and scattering. For example, in fingerprinting, recalibration is required to rebuild the radio map during the offline phase. The overhead involved in the search process also poses a large challenge. In the trilateration methods, the distance estimation is the primary issue in providing good accuracy. However, the multipath and signal attenuation were obstacles in obtaining accurate distance measurements. In AOA-based techniques, additional hardware is required in order to estimate the angles, which causes it to become more expensive than other techniques.

2.2 Tracking-based techniques

Chon and Cha proposed LifeMap, which generates user trajectories using the accelerometer and digital compass in a smartphone [1]. They provided a technique to verify all possible movement directions of a user's smartphone to circumvent noisy sensors. However, the initial position of the user must be detected using GPS, which is ineffective in indoor environments. This means that the user must begin from an outdoor position and then move into the indoor environment in order to obtain accurate results.

Constandache et al. [17] proposed CompAcc and it uses the compass and accelerometer in the smartphone to detect the direction and displacement to determine the user's directional trail, which is then used to compute the path signatures. Then, the directional trail is compared with the path signatures to determine the user location. However, this approach requires the user's positions along a path to be estimated. In [18], a dynamic fingerprint approach is used by combing the acceleration data and the RSS values. The approach constructs the floor plan database and tracks the user based on these information. This approach needs to download an application on users' phone to gather the information, such as, acceleration and RSS.

Jin et al. [19] developed SparseTrack, and it uses additional sensors to obtain more accurate step counts and displacement estimations to approximate the user trajectory. However, the need for additional sensors renders this technique inappropriate for current smartphones.

In UnLoc [11], a technique for correcting the user's position based on landmarks, such as Wi-Fi APs, elevators, and other items was developed. It relies on a collaborative method to find a small Wi-Fi area of which all locations overhear a distinct set of APs. However, it is not easy to find much of those small areas to improve the accuracy; therefore, UnLoc found only 12 areas in two buildings. Furthermore, different devices can have different RSS values, which reduce the feasibility of clusters methods. EZ [20] uses RSS of APs to detect the user position via genetic algorithm. It uses log-distance path loss model to detect the distance of an AP.

2.3 Map construction techniques

CrwodInside [21] presented a technique to build the floor plan by tracing the users during their daily life. This technique equips the users' phones with an application that sends some information, such as acceleration and Wi-Fi RSS to a server. After a period of time, the server builds the floor plan by clustering the users' traces. However, this technique requires GPS to determine the building by detecting the GPS signals lost near the building entrance. GPS signals can be available for a few meters inside the building, which may affect the accuracy of this technique since the initial position should be determined by the GPS signal. Nevertheless, their approach presents a new trend in building the floor plan, especially tracing the users, which can be used to enhance the floor plan construction techniques.

SLAM [22] is a technique in the mobile robotics to track a mobile robot in an unknown indoor environment. This technique builds the map automatically during tracking the robot.

SmartSLAM [23] improved and modified SLAM [22] approach by employing smartphone sensors. They presented a motion model using sensors, and observation model using Wi-Fi signals. Their models improved the accuracy of the tracking system, which they used to build the floor map. However, the generated map does not include all the components of the building. Moreover, it does not give the position coordinates.

ARIEL [24] presents a technique to automatically detect the room fingerprints based on Wi-Fi signature. Although this technique provides accuracy of 95 % compared to some existing works, but the system still needs much effort to build the database. In contrast, CALL needs two rounds of walking in the building to detect the position of the reference points. In [25], a technique to build an indoor floor plan using room fingerprints was presented. It relies on the observation that the RSS values in adjacent rooms are similar. This approach can detect the rooms and the hallways, and then connect them.

In summary, the existing works suffer from different problems. Fingerprinting technique needs to rebuild the huge radio map database from time to time due to signal attenuation, which needs a lot of effort. Trilateration techniques suffer from inaccurate distance estimation due to multipath. Sensor based techniques suffer from noisy sensors. Most of the existing works need to upload the floor plan, which is not available sometimes. The proposed method addresses the previous limitations in a number of ways. First, observing the RSS of APs provides a number of reference locations in indoor environments, which can then be used to adjust the user's position in real time. Second, reference points are used to build the floor plan automatically. Third, the proposed approach provides a more accurate distance and direction estimation using an accelerometer and an orientation sensor with the help of the reference points. Fourth, the proposed approach can dynamically detect any changes in the environment using the feedback system phase. Fifth, the proposed approach can reduce the effort of building the database by automatically update the database.

3 Proposed method

In this section, we present the overall design of the system as shown in Fig. 1. The system consists of several phases: Floor plan construction, Localization, and Feedback system.

During the floor plan construction phase, three steps are performed. First, the user surveys a building one time for calculating a RSS threshold for each public AP, which is usually attached to the ceiling of the corridors and can be detected using the SSID. Second, the user surveys the building one more time and the system generates the position of each public AP based on its RSS threshold, which is computed in the first round. Third, when the user moves inside the building, the system stores his/her path according to the position of the public AP. Afterwards, the floor plan with the public AP positions is constructed. Note, the public AP will be called as reference point hereafter.

In the localization phase, the position of the user is provided using the closest reference point, and then the system tracks the user using smartphone accelerometer and digital compass sensors respectively.

The feedback system phase observes the average RSS of the reference points and the data from the sensors in order to dynamically update the RSS thresholds and the positions of the reference points during the localization phase. It also stores the paths of the users during the localization phase to improve the floor plan.

CALL relies on the following assumptions; First, each building has a number of public APs (reference points), which share the same SSID and attached to the ceiling of the corridors or rooms with well-distributed topological locations. It means that the network designer takes into account the distance between each two APs; therefore, the interference and overlapping will be very limited between the public APs. This assumption is applicable since attaching them to the ceiling gives better coverage. Second, the server contains a database of the MAC addresses for all the reference points and their corresponding building and floor number. This assumption can be easily applied since the network administrator already has the MAC addresses of all the public APs. Third, the user should hold the smartphone in his/her hand in order to watch advertisements, YouTube, TV program, and so on. Fourth, the multipath impact on Wi-Fi RSS is not considered due to its negligible impact [26] because we consider the situation when the user is approaching and near an AP in order to guarantee that the signals are clear and in line of sight.

3.1 Floor plan construction phase

Providing the floor plan automatically makes the system more dynamic and flexible, since not all the buildings have floor plans that can be converted to digital ones. Moreover, the need to distinguish the floor plan components such as corridors and rooms makes the dynamic floor plan construction very important. For instance, the proposed system corrects the direction and predicts the coming reference point according to the corridor or the room that the user is located.

The floor plan consists of several parts, the positions of the reference points, the corridors, the room, and the hallways.

The challenge is how to accurately provide the floor plan with the position of the reference points. In this section, we provide the details about the floor plan construction.

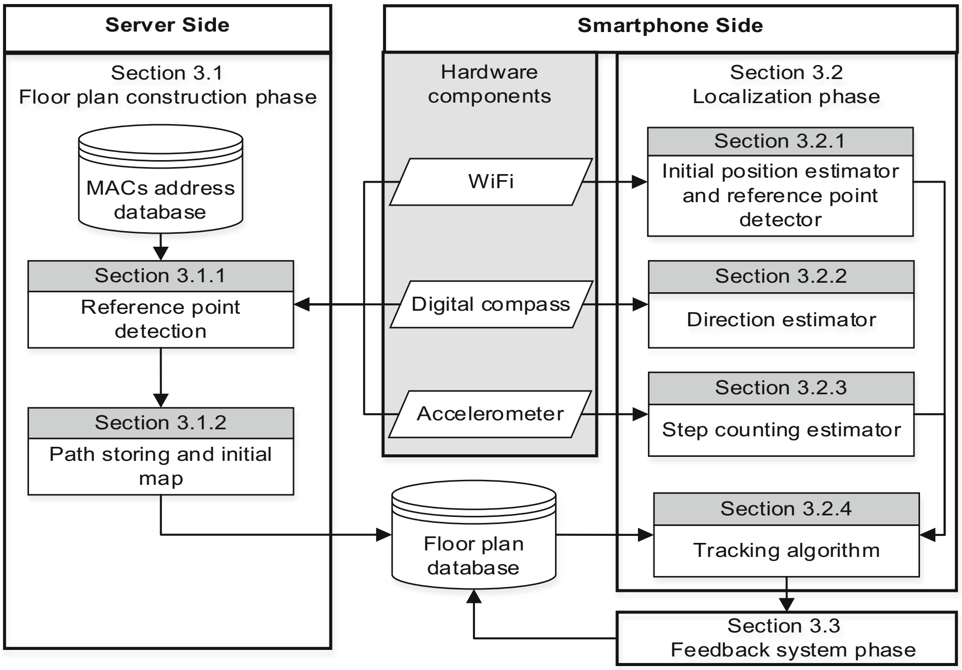


Fig. 1 System architecture

In order to provide the floor plan, our idea is to detect the positions of the reference points and then the floor plan is estimated by storing the trajectories of the users during their movement throughout the floor. The reference points will be used to detect the positions of the users. Also, the reference points will be used to adjust the position of the users in order to mitigate the error in distance estimation. Then by storing the trajectories of the users, the structure of the floor plan can be estimated over time.

3.1.1 Reference point detection

Our system detects the position of a reference point when the user passes under that reference point. Then the first detected reference point will be considered as a reference for the detection of remaining reference points.

The general procedure of detecting the reference points is as follows: The user should survey the building twice. The first survey is to determine the RSS thresholds, which are required in the detection process of the reference points. And the second survey is to detect the position of each reference point. In the first survey, the user should survey the floor along a path that he/she traverses, and then the system computes the average RSS values for every time window (e.g., 2 s) and stores the RSS average in the server. In the second survey, as soon as the RSS value surpasses the RSS threshold of a reference point, then the system computes the RSS average for every time window surpassing the RSS threshold. The position of the reference point is detected when the RSS average is the strongest among the computed RSS averages. Usually the strongest RSS average is for the time window when the position of the user is under a reference point. Then, accelerometer and orientation sensors as well as the RSS average values are utilized to detect the position of remaining reference points with respect to the first detected reference point. The distance and the direction between the reference points are computed using the accelerometer and the orientation sensors, respectively.

Since the reference points are the bases of our idea, detecting the accurate position of the reference point is the most critical challenge in our work. For this reason, we study the RSS when the user passes by a reference point to come up with an indicator that represents the time when the user passes under a reference point. RSS values are fluctuated and unstable; therefore, measuring the individual RSS values will be inaccurate. To overcome this issue, we measure the average RSS for a time window (e.g., 2 s). Our idea is to use the maximum average of the RSS as an indicator of the closest point from a reference point. In order to determine whether or not the maximum of the RSS average during a time window leads to the closest reference point, several experiments were conducted in the CS Department building and IT Convergence building at KAIST University during the daytime when many students were active and at nighttime when fewer students were active. These measurements were taken as a user passed a reference point.

Figure 2 demonstrates that the closer a user is to a reference point, the higher the RSS average will be, which means that at the time of the maximum average RSS value, the user is at the closest point to the reference point.

Depending on our observation that the maximum average RSS value gives the closest point from a reference point, we made our approach as follows: First, the user should survey the floor along a path he/she traverses, and then the system calculates the average RSS values for every time window (e.g., 2 s) and stores the RSS average in the server. In this experimentation three time windows associated with an AP are sufficient for calculating the RSS threshold considering the user walking speed.

In Fig. 3, the solid and dotted lines represent the AP1 and AP2 RSS thresholds. Figure 3 demonstrates that there are a small number of RSS average values above the RSS thresholds. This indicates that the RSS threshold can determine a small area surrounding a reference point (e.g., three values above the RSS threshold mean 6 s walking if the time window is 2 s, which means that the area is small since the maximum time that the user needs to pass this area is 6 s for the regular walking).

The values above the lines represent the values of the RSS averages when the user is at the closest point to the reference point.

Second, the user should survey the building one more time. When the RSS average value for a window surpasses the RSS threshold for a reference point, then the user will be in the vicinity of that reference point. After that, the system observes the RSS average for each time window surpassing the RSS threshold until the strongest RSS average is determined, and then the system considers this place as the position of that reference point.

To sum up, the user needs to survey the building twice. The first survey is to determine the RSS thresholds. The second one is to find the position for each reference point by calculating and observing the RSS average for each time window surpassing the RSS threshold until the strongest RSS average is detected.

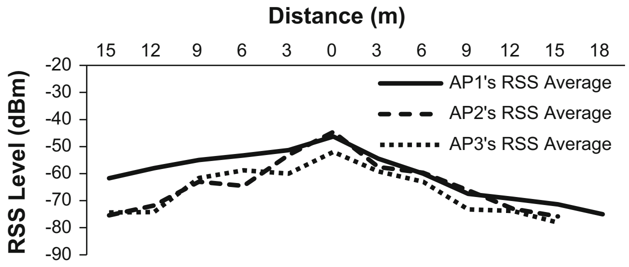


Fig. 2 RSS average within ±15 m from an AP

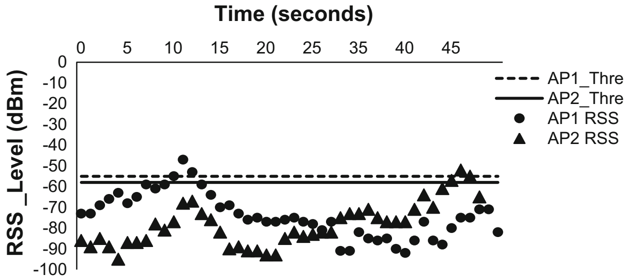


Fig. 3 The RSS average values when a user passes by two reference points

Our experiments demonstrate that the error of calculating the position of the reference points did not exceed 3 m (more details in Sect. 4).

3.1.2 Path storing and initial floor plan

After the reference points have been detected, the next step is to store the paths of the users inside the building. CrowdInside [21] takes advantage from the users' movements during their daily life, such as moving inside their homes and offices, by recording their movements in the database, and then over time, the layout of the building can be detected.

Our approach is similar to CrowdInside [21], with a few modifications. It works as follows: First, when a user encounters a reference point during his/her normal movement in the building, the system finds the user's initial position as the position of the closest reference point. Note here, the database has a prior knowledge of MAC addresses of each reference point and which building and floor number that these MAC addresses belong to. This information should be provided by the network administrator before the system works. Second, using smartphone sensors, accelerometer and digital compass, the system keeps observing the movement of the user during a time window (e.g., 2 s). It stores the number of steps, the direction, and the average RSS value as a record in the database. Over time, number of paths will be stored in the database. These paths with the position of the reference points will form the initial floor plan.

The floor plan consists of multiple paths; each path consists of number of points, which consist of RSS average value, number of steps, and the orientation average. The point is drawn every time window (e.g., 2 s).

The initial floor plan consists of the positions of the APs and the relative corridors layouts. To mitigate the error in distance and direction estimation, the values of RSS average, number of steps, and orientation average will be reset every time window after storing those values in the database. Resetting these values makes the system avoid the accumulative error in distance and direction estimations.

Although the initial floor plan consists of the positions of the reference points and the corridors layout, but later on, the system feedback is activated each time a user uses the system. When the users walk in the building and ask their positions, the system feedback sends the RSS average, steps, and the direction every time window as a point to the server. The server keeps storing the points to form the paths. Because the users visit many and different places in the building during their movement, such as different rooms or corridors, the server will be able to form the floor plan and attach the rooms and the hallways.

3.2 Localization phase

During the localization phase, the average RSS is verified in order to acquire the MAC address of the closest reference point, which represents the input for the *InitialPositionEstimator*. The *InitialPositionEstimator* uses the MAC address of the reference point with the maximum average RSS as input and verifies the *database* to determine which floor of the building this reference point belongs to. The current building floor that the user is located in and the closest reference point will be stored in a *result database* as the initial position of the user. When the user begins moving, the data from the accelerometer and digital compass sensors are monitored and collected in order to acquire the direction and the number of steps of the user. During the user movement, the Wi-Fi scanning process searches for the maximum of the RSS's average, which represents the peak value. If the peak value is detected, the distance and position are corrected based on the location of the reference point. The maximum of the RSS's average is monitored because only the step count using acceleration is insufficient to obtain accurate distance information. Therefore, the position of the reference point is used to correct the distance. Finally, the stored data in the database will be used to obtain the user's trajectory.

3.2.1 Initial Position and Reference point detector (IPR)

One of the significant contributions of this paper is the adjustment of the user's position in real time according to the stored reference points. Step counting algorithms, such as the Dead Reckoning algorithm [1], suffer from irrelevant user movements, such as the user's hand shaking, which causes increases in the error bounds in the distance estimation as the distance increases. If a fixed step length, which is usually used to determine the distance, is added, then the error bound becomes more serious.

The initial position of a user is estimated when the reference points are detected. This is undertaken by searching for the maximum of the RSS's average from these reference points. Then, the MAC address is extracted from the beacon frame, which is then used to extract the reference point's location.

The initial position estimator uses only the reference points that are stored in the database to detect the initial position. Several experiments were conducted in order to investigate the relationship between distance and RSS values when a user passes a reference point. Figure 4 presents an example regression analysis, which was conducted for three different APs. It shows that there is a strong correlation between the distance and RSS values because the R^2 is above 0.5 in most cases and the slopes are negative, which implies that there is a strong negative correlation.

The negative correlation between the distance and RSS values gives an indicator that the RSS value can be used to determine how far is a user from an AP.

Figure 4 also illustrates that there are some extreme RSS values due to the instability of the signals; therefore, preprocessing should be performed in order to eliminate the influences of these values.

In order to eliminate the extreme values, the procedure is simple. First, the RSS values are stored and grouped for each window of time (e.g., one second). Second, the RSS median, max, and min values are computed. Third, the RSS value is marked as extreme value if its difference with the median RSS value of the time window is greater than the mid value between the maximum and the minimum RSS values of the time window ((|Median − RSS|) > (|Max − Min|)/2). The system eliminates the extreme RSS values by excluding them from the average computing. For example, the correlation value for AP1's RSSs versus distance was 0.6 for the first five beacons that were received during a window time of one second with a beacon time interval of one hundred milliseconds; however, after smoothing the data, the correlation was improved to 0.7.

In summary, the initial position can be estimated as follows: the RSS average is computed for a window of

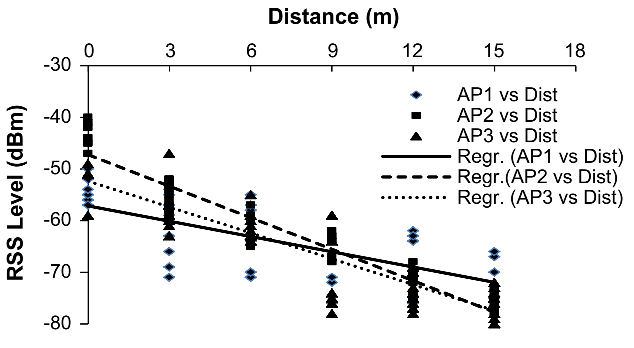


Fig. 4 Regression example of an RSS with distance

time. The IPR observes the RSS average until it begins to decrease, which will be the time when the user passes under a reference point.

The MAC address will be extracted from the beacon and the location of the user is determined as the location of the reference point based on the database. Figure 5 illustrates the scenario of detecting the initial position.

The maximum value of the average represents the time when a user passes under a reference point; this value is called the global maximum. However, due to the instability of the signals, the local maximum, which means that the average RSS decreases before reaching the global maximum might be obtained as shown in Fig. 5. In order to avoid adjusting the position based on the local maximum, the IPR tracks the maximum average for the last two continuous windows when the average begins to decrease. If the average increases again, then the IPR ignores the current maximum; otherwise, it stops and adjusts the user's position based on the maximum average, which is the global maximum. For example, if the window time is one second, then the IPR tracks the average for the last two seconds until it finds the global maximum; therefore, the reference point requires three time windows in order to be determined.

IPR stores the measured averages when the user is walking and stops measuring if the user is stationary.

The significant advantage of the proposed CALL method over the existing methods is that a number of reference points are used to adjust user's position in real time. The second task of the reference points is to adjust the user's position each time the user encounters a reference point; thus, the error bound is limited as discussed in Sect. 4.

When the IPR detects the initial position, the reference point is determined. Depending on the database, which has information about the sections and directions, the upcoming reference points can be easily determined (e.g., suppose AP2 and AP3 are on the west of AP1. If the initial reference point is determined using AP1 and the direction of the user is west, then IRP keeps observing RSS from AP2 and AP3 only). The IPR observes the average RSS for the upcoming reference points until it locates the global maximum; therefore, the IPR neglects other RSSs from other reference points.

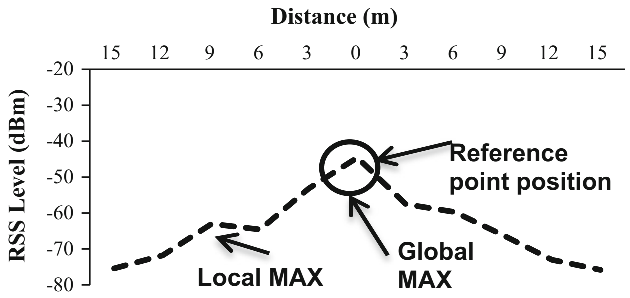


Fig. 5 RSS average with local and global MAX

3.2.2 *Direction estimator*

The direction of the user must be estimated after determining its initial position, and this is used as an indicator for the next building section that the user is moving toward.

Han and Kim used a smartphone orientation sensor, which provides three values that represent the azimuth (the angle measured clockwise from the magnetic north of the Earth to the y-axis of the smartphone), pitch (rotation around the x-axis), and roll (rotation around the y-axis), to perform the mobility prediction [27]. One issue with orientation sensors is that they are easily affected by user movements such as shaky hands. In order to obtain more accurate results, the sensor values must be measured over a period of time. In the proposed CALL system, the situation where the user is holding the smartphone in their hand in order to watch advertisements, YouTube, TV program, and so on is considered. Then, the sensor data collection begins when the user enters a building section.

The average azimuth is computed for each section during the period of time between the beginning of the section and the end. Computing the average azimuth value for a period of time can help to mitigate the errors in direction estimation. For instance, if there is an irrelevant movement, such as hand shaking or if the user passes by any magnetic source, then the azimuth suddenly changed. To overcome this problem, the direction estimator ignores any value and marks it as noise if there is any abrupt difference in azimuth happened.

Due to the noise from the sensor, the direction estimator uses the low pass filter to keep the azimuth values within a range. This procedure helps to reduce the error in the direction estimation, but because the error is accumulative, the system needs to mitigate this accumulative error. To do so, the direction is recalibrated each time a user encounters a reference point; therefore, the direction will be reset frequently during the user movements.

One of the advantages of CALL is predicting the next reference point according to the user direction. For instance, suppose that the reference points y is located in the west of reference point x and reference point z is located in the east of x. If the detected reference point is x and the direction of the user is west, then smartphone will decide that the next reference point is y; therefore, the search algorithm will keep looking for the reference point y and ignores the reference point z from the search process.

3.2.3 Step counting estimator

Accurate distance estimation is a critical issue in indoor positioning systems. Therefore, in order to accurately estimate the distance using smartphone sensors, e.g., an accelerometer, the most widely used technique is to count steps using the Peak Detection Algorithm (PDA) [28].

The PDA uses an accelerometer sensor to locate peaks, where each peak represents a step. The algorithm functions as follows. If the current accelerations are greater than the movement standard deviation threshold for duration *W*, then a movement is detected. A peak is detected when the current acceleration is greater than the sum of the mean and standard deviations for a duration [i.e., between (current time—*W)* and current time]. Finally, the distance is calculated by multiplying the number of steps by the fixed step length.

The primary drawback of this technique is miscounting steps due to shaky hands or other irrelevant smartphone movements. Another problem is that the step length can vary. Therefore, the PDA is prone to errors. In order to address these problems, the real distances are stored in the database and are used to obtain the correct distances.

The step counting estimator performs a data smoothing before detecting the peaks to mitigate the noise from the accelerometer sensor. High pass filter is used to mitigate the error by computing the real acceleration after eliminating the force of the gravity.

The proposed CALL system corrects the user position to be the position of the reference point when the user encounters a reference point. In order to correct the distance, suppose that (x, y) is the position of a reference point, (x_1, y_1) is the user position according to PDA, and PDA_DIS is the distance measured by the PDA; then, the difference in the distance between the user position based on the PDA and the reference point is:

$$\text{Diff\_DIS} = \sqrt{(x_1 - \mathrm{x})^2 + (y_1 - \mathrm{y})^2}. \quad (1)$$

The distance is corrected using Eq. 1 by adding or subtracting the Diff_DIS from the measured distance, as follows:

$$\text{Distance} = \begin{cases} \text{PDA\_DIS} + \text{Diff\_DIS}, & \mathrm{x1} < \mathrm{x} \\ \text{PDA\_DIS} - \text{Diff\_DIS}, & \mathrm{x1} \geq \mathrm{x} \end{cases}. \quad (2)$$

In summary, PDA detects the current position according to the previous position based on the step counting and the direction. Therefore, the user can move without restrictions throughout the floor. If a reference point is detected, the current position will be adjusted according to the reference point position in order to solve the accumulative error due to the miscounting steps in PDA.

The error bound in the distance and position estimation of the proposed CALL is very limited, because each reference point has a fixed location in the building according to the database.

3.2.4 Tracking algorithm

The tracking algorithm which is shown in the flowchart shown in Fig. 6 works as follows. First, if the user is walking inside the building, the tracking algorithm fetches the received beacon frame to extract the RSS and MAC address. If the average RSS within a time window is a global maximum value, then a query will be sent to the server with the MAC address. The server returns the building and the floor number, since it has a table that contains the MAC addresses and the corresponding floor numbers. Second, the tracking algorithm detects the initial position as discussed in Sect. 3.2.1, and then determines the direction and the number of steps of the user as shown in Sects. 3.2.2 and 3.2.3.

Third, if the initial position is already estimated when the global maximum is found, it means the user has encountered a reference point during his movement; therefore, the tracking algorithm updates the position and distance as discussed in Sects. 3.2.1 and 3.2.3. Feedback system keeps tracks of the average RSS, the direction, and the number of steps in the database.

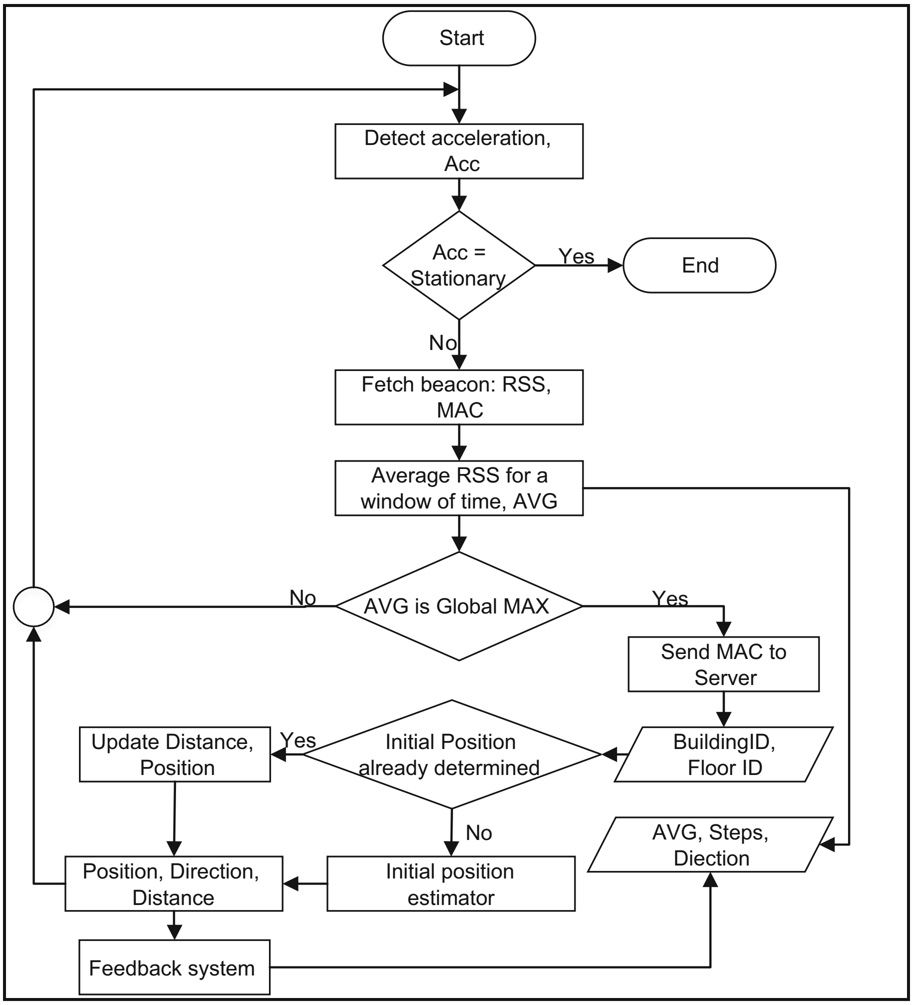
3.3 Feedback system phase

This phase is embedded in the localization phase. It means that during the localization phase, the average RSS values are observed and the data is sent to the server. In the server side, reference points' RSS thresholds and positions will be updated, and the paths of the users will be stored as well to update the floor plan each time a user uses the localization phase as shown in Sect. 3.1.

After the initial floor plan is estimated using two rounds, then the system feedback is used to enhance the RSS threshold, the floor plan, and the position of the reference point. As a result, the other users do not need to survey the building again, feedback system provides all the information such as, the RSS thresholds, the floor plan, and the position of the reference points. Therefore, the system needs the two rounds only once.

This phase makes the system dynamic and flexible for any change in the building, such as installing new reference point or removing one. Another benefit is to make the position of the reference points more accurate by calculating the position of each reference point each time a user uses the localization and then the average of these positions will be computed, which makes the system more robust. RSS threshold values will be more accurate and more

Fig. 6 Tracking algorithm flowchart



representative by calculating the thresholds average regularly. The feedback system enhances the layout of the floor plan by storing the paths during the localization phase. As long as the user uses the localization phase, the visited places in the building will be increased. So the more paths we store, the more accurate floor plan we get.

4 Evaluation

4.1 Experiments setup

The proposed CALL was evaluated in three different environments at KAIST: the CS Department, the IT Convergence building, and the KI building.

CS Department has four floors with dimensions of 120 m length and width of 70 m. IT Convergence has nine floors and dimensions of 70 m length and width of 30 m. KI building has four floors, the first floor is a huge open space with dimensions of 100 m length and width of 120 m. We chose this environment because it can be is similar with a shopping mall.

The experiments were conducted in the first and second floors of CS Department, which have 8 and 7 reference points respectively, with average distance of 20 m between each two of them. And in floors seventh to ninth of IT Convergence, which have 8 reference points each, with average distance of 17 m between each two of them. The last experiment was conducted in KI building first floor to test the ability of CALL in the open space buildings.

The proposed method was implemented on an Android-based Galaxy S III smartphone and an Android-based LG Optimus LTE2 smartphone, which are both equipped with accelerometer sensors, orientation sensors, and Wi-Fi. The experiments were conducted during the daytime when the RSS levels may be affected by obstructions between the APs and the user's smartphone.

4.2 Experiments results

In this section, we present the result of floor plan construction, localization, and feedback system.

4.2.1 Floor plan construction

This section presents the experiment results for detecting the reference points and storing the users' paths, the following subsections provide the details:

- *Detecting the reference points*

This subsection presents the result of the experiments of detecting the reference points in several floors of CS Department, IT Convergence building (N1), and KI building. Table 1 shows the obtained results.

Table 1 shows that the average distance error of detecting the reference points of different floors in different building is quite similar, and does not exceed 3 m. Also Table 1 shows that the system can detect most of the available reference points.

Figure 7(a) demonstrates an example floor plan of the reference points in 8th floor of IT Convergence building. The figure shows the real position of the reference points and the estimated position of the detected reference points. In order to verify the ability of the feedback system for detecting the remaining reference points, we chose 8th floor, since reference point detection found 6 reference points out of 8 in that floor.

- *Storing the path of user and initial floor plan*

Figure 7(b) shows the path of a user, which starts when the user encounters a reference point.

A dot in the path is drawn every time window (e.g., 2 s), and it represents some information, such as acceleration, average RSS, and digital compass value. Figure 7(b) and (c) show the initial floor plan, which consists of one path and does not give any components of the building except the corridors. The initial floor plan, which consists of the reference points and relative corridor layout, will be used in the localization phase.

The difference between this technique and CrowdInside [21] is that we need little effort to form the layout of the floor. Even though the initial floor plan is not accurate, but the feedback system improves the floor plan over time (Sect. 3.3).

4.2.2 Localization

The experiments were performed using two scenarios. Scenario 1 was performed on the eighth floor in the IT Convergence building. In this scenario, the user began from one reference point. Then, the user moved inside the hallway in the building. Several reference points were used in this scenario. In order to verify the efficiency of the proposed CALL, the user passes all reference points. Figure 8(a) illustrates the user trajectory and reference point locations.

Figure 8(b) presents Scenario 2 that was conducted on the first floor of the CS Department building. In scenario 2, the user began in a corridor and moved inside the building. The user changed direction several times during the movement.

Figure 8(b) also presents the user's trajectory and reference point locations. The real path of the user is drawn with a green solid line, the CALL path is drawn with a red long dashed line, and the PDA with compass path is drawn with a blue small dotted line. As seen from Fig. 8(a), the CALL path is very similar to the real path, while the accuracy of the PDA decreases as the also shows that PDA path is relatively similar to the real path because PDA performs well in the simple topologies where the distance is short and the user movements are stable. 'D' in Fig. 8(b) represents an incident in that the user encounters a reference point. As seen in the figure, the position has been adjusted directly to be the location of the reference point (nearest black dot).

4.2.3 Feedback system

The purpose of the feedback system is to improve the accuracy of the positions of the reference points and the floor plan. Figure 9 shows the reference point positions after five times of conducting the localization phase. The feedback system keeps track of the position of reference points each time the localization phase is used, and then the averages of

Table 1 Reference point detection results

Floor	No. of reference points	No. of detected reference points	Avg. error (m)
N1 9th F	8	8	2.9
N1 8th F	8	6	3
N1 7th F	8	8	2.6
CS 1st F	8	8	2.6
CS 2nd F	7	6	2
KI	5	5	2.9

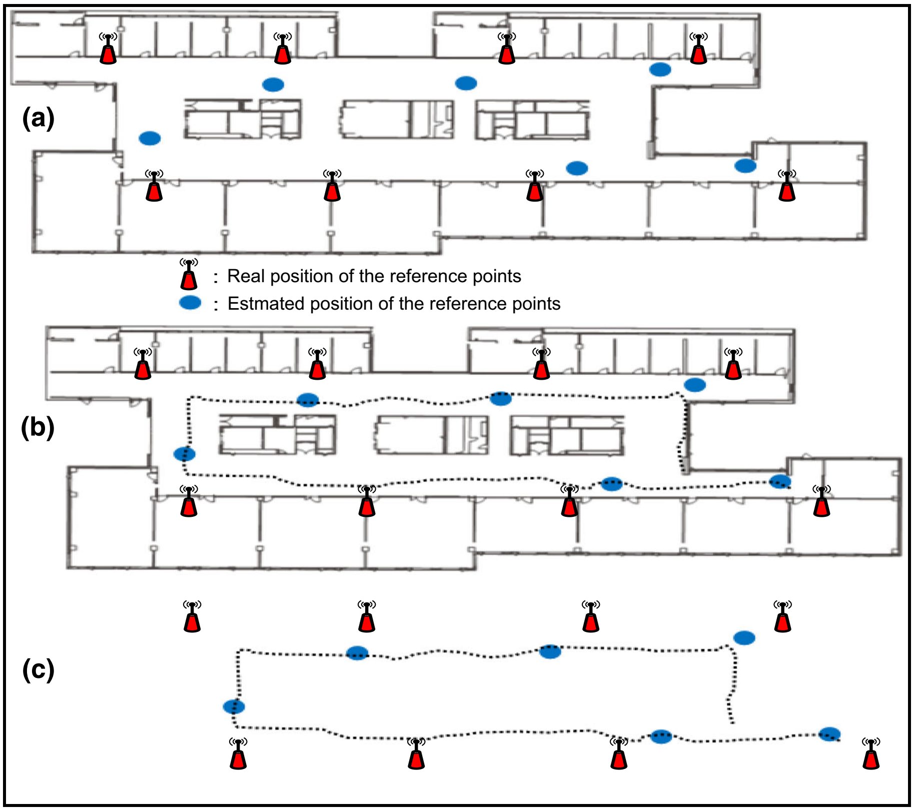

(a)
: Real position of the reference points
: Estmated position of the reference points
(b)
(c)

Fig. 7 The process of detecting the reference points position and the initial floor plan. **a** The initial reference points positions, **b** the user path based on the estimated reference points, **c** the initial floor plan, which consists of the estimated reference points and the user path with the respect of the real position of the reference points

all positions are computed as new positions of the reference points. The results of the feedback system are new positions of the reference points and the floor plan. In other words, the system feedback updates the database automatically during the localization; therefore, the more the localization phase is used, the more accurate position of the reference points and floor plan will be.

According to Fig. 9(a) and (b), which show the reference point positions and the floor plan, we argue that over time, the general layout of the floor will be formed, since the number of visited places will be increased.

The maximum error of localization was 6 m due to the inaccuracy in initial floor plan. However, after performing the feedback system, which enhances the initial floor plan, the error is reduced to less than 3 m.

4.3 Error analysis

The findings of our experiments indicate that there are some errors in reference point detection system and localization system. In this section we present the error analysis for both systems.

4.3.1 Reference point detection

In order to evaluate the robustness of the reference point detection system, Fig. 10 demonstrates the cumulative distribution function (CDF) graph, which shows the accumulative error in detecting the reference points in various floors.

Figure 10 illustrates that the precision of detecting the reference points within 3 m is 80 % in the three environments even though these environments have different dimensions. Thus, the system provides sufficient accuracy of detecting the positions of the reference points, which will be used to detect the users' positions afterward.

After performing the feedback phase with five times of localization, the average error was reduced to 2.5 m in the first floor of the CS building and 2 m on the eighth floor of the IT convergence building. The results show the impact of the feedback system in enhancing the accuracy of the system.

4.3.2 Localization

The findings of our experiments indicate that there are some errors between the real and estimated locations, which originate from the uncertainty in the measurements from the distance estimation and initial position estimation. The error bound of CALL can be derived by calculating the distance difference between the real position and the CALL position using Euclidean distance.

Formally, let (x^t, y^t) is the real user position at time, t. and let (x_1^t, y_1^t) is the CALL user position at time, t. Then:

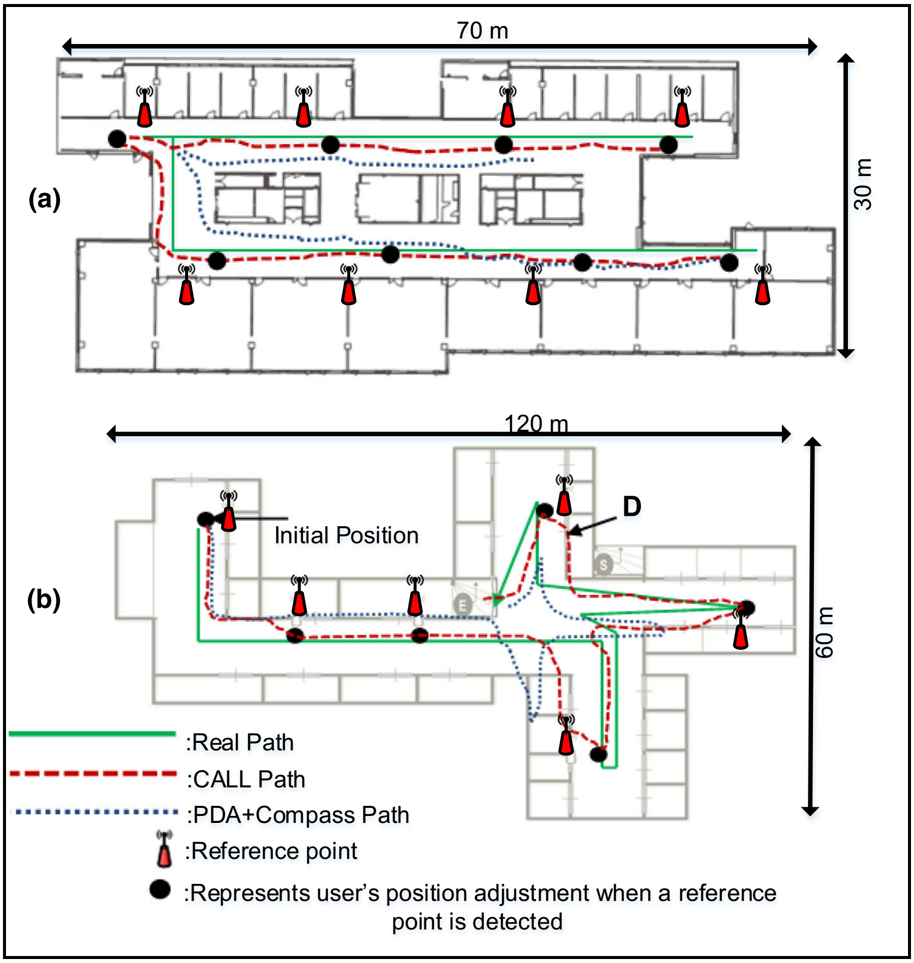


Fig. 8 Localization scenarios: **a** scenario 1 in IT convergence building, **b** scenario 2 in CS building

$$\text{Error} = \sqrt{(x_1^t - x^t)^2 + (y_1^t - y^t)^2} \qquad (3)$$

In Scenario 1 [Fig. 8(a)], the mean error and standard deviation error for the CALL were 2 and 0.9 m, respectively. In contrast, the PDA had a mean error of 5 m and standard deviation of 2.2 m. In Scenario 2 [Fig. 8(b)], the CALL mean error and standard deviation were 3 and 1.1 m, respectively. The PDA had a mean error of 6.3 m and standard deviation of 3 m. In both scenarios, the mean error of estimating the initial position for the CALL was 3 m, while the standard deviation was 1 m for both scenarios.

Considering only the accuracy is not enough to test the robustness; therefore, the location precision, which means how consistently the system works, can be used to determine the robustness of the system.

Location precision is determined by computing the CDF. Figure 11 demonstrates the CDF graph, which shows that the CDF of distance error of 3.8 m is 0.9, which means CALL has a location precision of 90 % within 3.8 m. Also, the CDF of distance error of 2.2 m is 0.6, which means a precision of 60 % within 2.2 m.

4.4 Comparison

We have compared CALL with different approaches as follows:

- CALL versus fingerprinting:
- Database size: CALL collects the RSS average values of the reference points during the first survey, in the end of the first survey (one round of walking), CALL calculates the RSS threshold for each reference point. During the second survey (another round of walking), CALL detects the position of the reference points using the strongest RSS average value with respect to the RSS threshold. In the end, the database of CALL contains the MAC addresses and the RSS threshold for each reference point. It means that the database is very small compared to fingerprinting approach, which

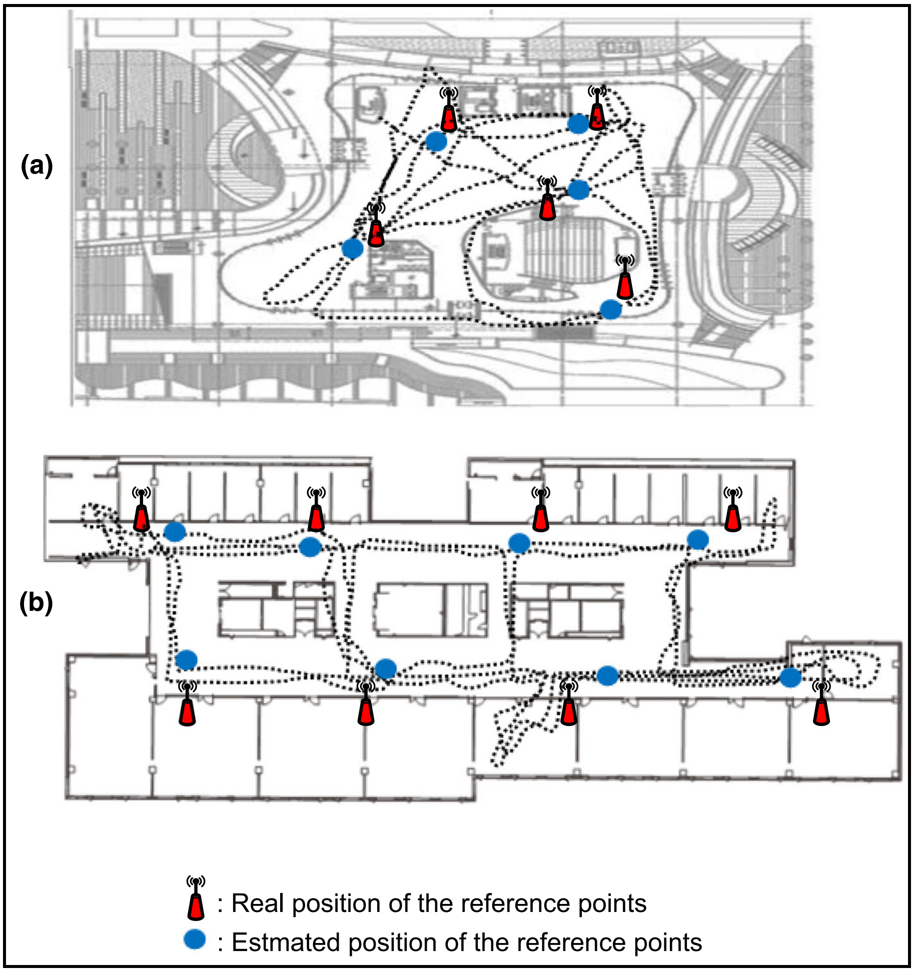

(a)
(b)
: Real position of the reference points
: Estmated position of the reference points

Fig. 9 Feedback system phase for updating the reference point's position and the floor plan during localization phase. **a** The reference points positions after five times of localization in KI building, **b** the reference points positions after five times of localization in N1 building

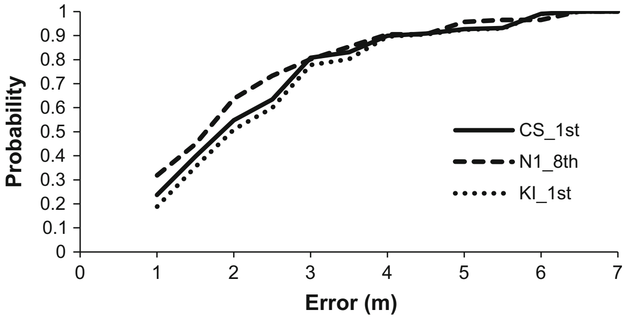


Fig. 10 CDF of reference points detection error for CALL

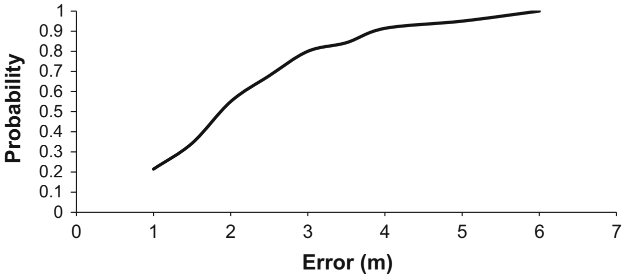


Fig. 11 CDF of positioning error for CALL

needs to collect the data for long period of time (at least 20–30 fingerprints for each marked place).

- Database calibration: Fingerprinting technique needs to build the radio map database from time to time due to any changes in the environment, such as installing a new AP. In constant, CALL has the feedback system phase, which updates the RSS thresholds automatically during the localization phase.
- Energy and time: we have measured the time and the energy for CALL to compare it with fingerprinting.

In term of energy, CALL consumes low energy for surveying the building (e.g., 164 mW in IT convergence building). The consumed energy for localization is very low, because only few beacons are needed and the search algorithm is linear. In contrast, RADAR consumes high energy for surveying the building, because of the huge number of collected fingerprints. For time measurements,

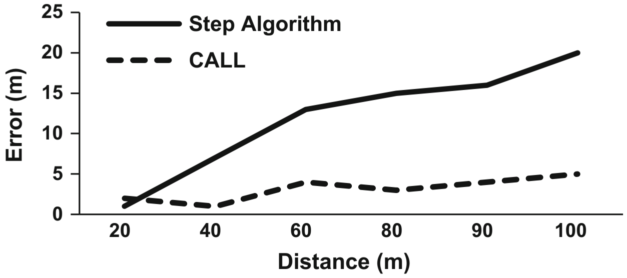


Fig. 12 Distance estimation using CALL and step algorithm

CALL needs few minutes in each floor for surveying the building, because only two rounds of walking are needed. RADAR needs to survey the building for long period of time in order to collect the fingerprints. CALL and RADAR need few seconds for localization.

Accuracy: CALL provides accuracy of 2–3 m, while RADAR system has accuracy of 3–5 m. Some fingerprinting approaches have better accuracy if the number of fingerprints is increased, which affects the time of surveying and rebuilding the database.

CALL versus smartphone sensors based approach:

CALL corrects the user's position and distance estimation each time he/she encounters a reference point. The current smartphone based approaches rely on the step detection algorithm to count the number of steps based on the peak value of the acceleration. This algorithm suffers from irrelevant smartphone movements, such as hand shaking; therefore, the more distance that the user moves, the bigger error will be. CALL solves this problem by correcting the position and the distance each time the user encounters a reference point, thus, the error is limited. Figure 12 shows the errors in distance estimation using CALL and the conventional step detection.

CALL has almost fixed error of 3 m in distance estimation. In contrast, the step detection algorithm, which is used by most of the smartphone based approaches has an accumulative error.

- CALL versus hybrid techniques:

UnLoc [11] and EZ [20] are ones of the recent papers, which are related with CALL. The details of UnLoc and EZ are described in Sect. 2. Table 2 shows the comparison between CALL and the aforementioned localization systems.

5 Discussion

One of the limitations of CALL is detecting the initial position. The user should encounter a reference point in order to detect his/her initial position. In the future work, we will provide the initial position even if the user is not close to any reference point. Another limitation could be the circular open space buildings, CALL performs well in these kinds of buildings but still needs to be enhanced since all the reference points are facing each other, which means predicting the next reference point will be more sophisticated. Another limitation is the direction estimation, where we assume that the user holds the phone. In the future work, we will consider when the user is not holding the phone.

6 Conclusion

This paper presents an indoor positioning floor plan construction and tracking system. The proposed approach builds the floor plan by detecting the reference points represented by public APs in the building. Typically, these reference points are attached to the ceiling of the corridors and have well-distributed topological locations. Based on this, the initial position of the user is determined using the nearest reference point with the strongest signal in average during a designated time window. In order to overcome the instability of the received signal strength, the search process is restricted to only the reference points recorded in the database. The user's direction is estimated using the orientation sensor after obtaining the average value during the user movement. In order to obtain the user movement, the acceleration data was gathered and the step counting technique can be used. In order to overcome the miscounting problem due to irrelevant movements of the smartphone, the AP locations in the building floor plan database are referenced to correct any errors in distance estimations. To improve the accuracy of the floor plan, a feedback phase is used to collect the sensors data and received signal strength averages during the localization system. As the feedback system accumulates the data, the positioning accuracy, the reference points' locations, and the layout of the floor plan are enhanced. The experiments with the proposed CALL system in a real environments showed that the distance estimation had a mean error of 2 m and the initial position

Table 2 Difference between CALL and other hybrid approaches

System	Accuracy (m)	Advantage	Disadvantage
CALL	2–3	Robust to any change in the environment	Needs reference point for initial position
EZ	5	No war-driving	Needs GPS
UnLoc	1–2	No war-driving	Needs door location

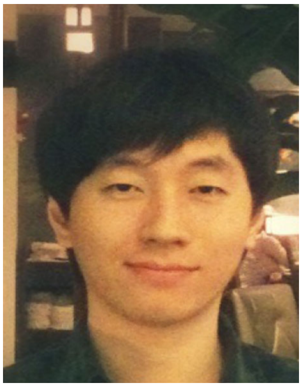
accuracy had a mean error of 3 m. Moreover, the mean error of detecting the reference points position did not exceeds 3 m. With the fixed reference points' locations, our proposed CALL system could avoid the problems of signal attenuation and/or scattering, and additional overhead of database search for tracking the user.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) Grant founded by the Korea government (MEST) (No. 2012R1A2A2A01008244). We would like to thank you for your kindness, it will be our pleasure to publish our paper with your valuable journal. We would like to express our grateful to all the members who work on this paper.

References

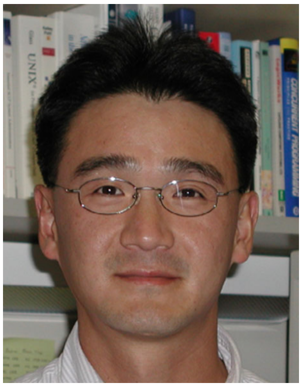
1. Chon, Y. & Cha, H. (2011). LifeMap: Smartphone-based context provider for location-based services. In *Proceedings of the IEEE PerCom* (pp. 58–67).
2. Chung, J., Donahoe, M., Schmandt, C., Kim, I. J., Razavai, P., & Wiseman, M. (2011). Indoor location sensing using geomagnetism. In *Proceedings of ACM MobiSystem* (pp. 141–154).
3. Baniukevic, A., Sabonis, D., Jensen, C.S., & Lu, H. (2011). Improving Wi-Fi based indoor positioning using bluetooth add-ons. In *Proceedings of IEEE Conference on Mobile Data Management (MDM)* (pp. 246–255).
4. Wang, J.S. & Shen, Z.X. (2002). An improved MUSIC TOA estimator for RFID positioning. In *Proceedings of IEEE Radar Conference* (pp. 478–482).
5. Zhao, F., Yao, W., Logothetis, C.C., & Song, Y. (2006). Comparison of super-resolution algorithms for TOA estimation in indoor IEEE 802.11 wireless LANs. In *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)* (pp. 1–5).
6. Yang, Z., & Liu, Y. (2010). Quality of trilateration: Confidence based iterative localization. *IEEE Transactions on Parallel and Distributed Systems, 21*(5), 631–640.
7. Niculescu, D., & Nath, B. (2001). Ad hoc positioning system (APS). In *Proceedings of IEEE Conference on Global Telecomm* (Vol. 5, pp. 2926–2931).
8. Fang, S. H., & Lin, T. N. (2010). A dynamic system approach for radio location fingerprinting in wireless local area networks. *IEEE Transactions on Communications, 58*(4), 1020–1025.
9. Mazuelas, S., Bahillo, A., Lorenzo, R., Fernandez, P., Lago, F., Garcia, E., Blas, J., & Abril, E. (2009). Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks. In *IEEE Journal of Selected Topics in Signal Processing* (pp. 821–831).
10. Peng, Z., Dan, W., & Yi, S. (2010). SMART: Simultaneous indoor localization and map construction using smartphones. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).
11. He, W., Souvik, S., Ahmed, E., Moustafa, F., Moustafa, Y., & Choudhury, R.R. (2012). No need to war-drive: Unsupervised indoor localization. In *Proceedings of ACM MobiSystem* (pp. 197–210).
12. Roshanei, M., & Maleki, M. (2009). Dynamic-KNN: A novel locating method in WLAN based on angle of arrival. *IEEE Symposium on Industrial Electronics and Applications (ISIEA), 2*, 722–726.
13. Bahl, P., & Padmanabhan, V.N. (2000). RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM* (Vol. 2, pp. 775–784).
14. Matic, A., Papliatseyeu, A., Osmani, V., & Mayora-Ibarra, O. (2010). Tuning to your position: FM radio based indoor localization with spontaneous recalibration. In *Proceedings of IEEE PerCom* (pp. 153–161).
15. Zhang, Y., Bao, L., Yang, S. H., Welling, M., & Wu, D. (2010). Localization algorithms for wireless sensor retrieval. *Computer Journal, 53*, 1594–1605.
16. Wu, D., Bao, L.C., & Li, R.F. (2011). Robust localization protocols and algorithms in wireless sensor networks using UWB. In *Ad Hoc and Sensor Wireless Networks* (pp. 219–243).
17. Constandache, I., Choudhury, R.R., & Rhee, I. (2010). CompAcc: Using mobile phone compasses and accelerometers for localization. *In Proceedings of IEEE INFOCOM*.
18. Wu, C., Yang, Z., Liu, Y., & Xi, W.. (2012). WILL: Wireless indoor localization without site survey. In *Proceedings of IEEE INFOCOM* (pp. 64–72), March 25–30.
19. Jin, Y., Motani, M., Soh, W.-S., & Zhang, J. (2010). SparseTrack: Enhancing indoor pedestrian tracking with sparse infrastructure support. In: *Proceedings of IEEE Conference on Computer Communications* (pp. 1–9).
20. Chintalapudi, K., Padmanabha Iyer, A., & Padmanabhan, V.N. (2010). Indoor localization without the pain. In *Proceedings of ACM MobiCom* (pp. 173–184).
21. Alzantot, M. & Youssef, M. (2012). Crowdinside: Automatic construction of indoor floorplans. In *ACM SIGSPATIAL*.
22. Durrant Whyte, H. & Bailey, T. (2006). Simultaneous localization and mapping. In *IEEE Robotics and Automation Magazine* (pp. 99–108).
23. Shin, H., Chon, Y., & Cha, H. (2012). Unsupervised construction of an indoor floor plan using a smartphone. In *IEEE Transaction on Systems, Man, and Cybernetics* (pp. 889–898).
24. Jiang, Y., Pan, X., Li, K., Lv, Q., Dick, R.P., Hannigan, M., & Shang, L. (2012). ARIEL: Automatic Wi-fi based room fingerprinting for indoor localization. In *proceedings of ACM UbiComp* (pp. 441–450).
25. Jiang, Y., Xiang, Y., Pan, X., Li, K., Lv, Q., Dick, R.P., Shang, L., & Hannigan, M. (2013). Hallway based automatic indoor floorplan construction using room fingerprints. In *Proceedings of ACM UbiComp* (pp. 315–324).
26. Sen, S., Lee, J., Kim, K., & Congdon, P. (2013). Avoiding multipath to revive inbuilding WiFi localization. In *Proceedings of ACM MobiSystem* (pp. 249–262).
27. Han, S., Kim, M., Lee, B., & Kang, S. (2012). Directional handoff using geomagnetic sensor in indoor WLANs. In *Proceedings of IEEE PerCom* (pp. 128–134).
28. Chon, Y., Talipov, E., & Cha, H. (2012). Autonomous management of everyday places for a personalized location provider. In *IEEE Transactions on Systems, Man, and Cybernetics* (pp. 518–531).

Ahmad Abadleh received his B.A. in Computer Science from Mutah University in 2002, and his M.S. in Computer Science from the University of Jordan in 2010, and his Ph.D. from the Korea Advanced Institute of Science and Technology (KAIST) in 2015. He is currently a faculty member of IT Department at Mutah University. His research interests include indoor localization, and wireless MAC protocol.



Sangyup Han received his B.A. in Computer Engineering from Kyungpook National University in 2010, and his M.S. in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST) in 2011. He is currently a Ph.D. student in Computer Science at the KAIST. His research interests include mobility support in IEEE 802.11, multimedia streaming in wireless and mobile Internet, and wireless MAC protocol.

Soon Joo Hyun is currently a faculty member of CS Department of KAIST and the professor of intelligent database systems research laboratory (iDB lab). He received his M.E. degree from the Faculty of Engineering, Katholike Universitate Leuven (KUL) and a Ph.D. degree from the University of Floridain 1987 and 1995, respectively. He worked with Bell Telephone/ITT, Belgium and Electronics and Telecommunications Research Institute (ETRI) Korea during the period of 1984–1997 before he joined ICU (Information and Communications Univ.) as a founding faculty member in 1998. He led many research projects in the areas of Context-aware Ubiquitous Computing and Sensor Network Databases. He recently led an industry project to develop a sensor network database management system, SNQL/SNQP for fixed and mobile sensor networks.



Ben Lee received his B.E. degree in Electrical Engineering in 1984 from the Department of Electrical Engineering at State University of New York (SUNY) at Stony Brook, and his Ph.D. degree in Computer Engineering in 1991 from the Department of Electrical and Computer Engineering at the Pennsylvania State University. He is currently a Professor of School of Electrical Engineering and Computer Science at Oregon State University. He has published over 100 conference proceedings, book chapters, and journal articles in the areas of embedded systems, computer architecture, multithreading and thread-level speculation, parallel and distributed systems, and wireless networks. He received the Loyd Carter Award for Outstanding and Inspirational Teaching and the Alumni Professor Award for Outstanding Contribution to the College and the University from the OSU College of Engineering in 1994 and 2005, respectively. He also received the HKN Innovative Teaching Award from Eta Kappa Nu, School of Electrical Engineering and Computer Science, 2008. He has been on the program and organizing committees for numerous international conferences, including 2003 International Conference on Parallel and Distributed Computing Systems (PDCS), 2005–2011 IEEE Workshop on Pervasive Wireless Networking (PWN), and 2006, 2007, and 2009 IEEE International Conference on Pervasive Computing and Communications (PerCom). He was also a Keynote speaker at the 2014 International Conference on Ubiquitous Information Management and Communication. He is currently the Chair for the Social, P2P and Multimedia Networking, Services and Applications track for the 2016 IEEE Consumer Communications and Networking Conference (CCNC). His research interests include wireless networks, embedded systems, computer architecture, multithreading and thread-level speculation, and parallel and distributed systems.

Myungchul Kim received his B.A. in Electronics Engineering from Ajou University in 1982, M.S. in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST) in 1984, and Ph.D. in Computer Science from the University of British Columbia, Vancouver, Canada, in 1993. Currently, he is with the faculty of the KAIST as Professor. Before joining the university, he was a managing director in Korea Telecom Research and Development Group during 1984–1997 where he was in charge of research and development of protocol and QoS testing on ATM/B-ISDN, IN, PCS and Internet. He has published over 100 conference proceedings, book chapters, and journal articles in the areas of computer networks, wireless mobile networks, protocol engineering and network security. His research interests include Internet, protocol engineering, mobile computing, and information security.