F-Loc: Floor Localization via Crowdsourcing

Haibo Ye[1], Tao Gu[2], Xianping Tao[1], Jian Lu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]School of Computer Science and IT, RMIT University, Australia
yhb@smail.nju.edu.cn, tao.gu@rmit.edu.au, {txp, lj}@nju.edu.cn

Abstract—Traditional fingerprint based localization techniques mainly rely on infrastructure support such as GSM, Wi-Fi or GPS. They work by war-driving the entire indoor spaces which is both time-consuming and labor-intensive. With recent advances of smartphone and sensing technologies, sensor-assisted localization techniques leveraging on mobile phone sensing are emerging. However, sensors are inherently noisy, making this technique challenging for real deployment. In this paper, we present F-Loc, a novel floor localization system to identify the floor level in a multi-floor building on which a mobile user is located. It does not need to war-drive the entire building. Leveraging on crowdsourcing and mobile phone sensing, we collect users' Wi-Fi traces and accelerometer readings. Through advanced clustering and cluster manipulating techniques, we are able to build the Wi-Fi map of the entire building, which can then be used for floor localization. We conduct both simulation and field studies to demonstrate the accuracy, scalability, and robustness of F-Loc. Our field study in a 10-floor building shows that F-Loc achieves an accuracy of over 98%.

Keywords—Mobile Phone Localization, Floor Localization, Wi-Fi, Accelerometer.

1 INTRODUCTION

With the increasing pervasiveness of mobile phones, we have experienced an explosive growth of location based applications (LBAs) [4], [8], in which the location of a mobile user has to be known. In a multi-floor building environment, knowing the floor level of a mobile user is particularly useful for a variety of LBAs. For example, in a fire emergency, locating the floor level of a user quickly and accurately is essential for minimizing delays in emergency response, which is critical to life saving. In a shopping mall or an airport environment, a navigation service such as Google maps can prompt a mobile user with the floor map by knowing her/his current floor level. This is known as the floor localization problem, which we aim to determine the floor level in a multi-floor building on which a mobile user is located.

Indoor localization has been well studied in the literature. The inability of GPS indoors has led to approaches based on alternative signals. The fingerprint-based approach leveraging on Wi-Fi or GSM appears most. SkyLoc [15] operates on GSM fingerprints, but is not good enough to identify the correct floor level. It only guarantees a floor location within three adjacent levels. RADAR [3] and PlaceLab [13] use Wi-Fi and GSM signals. The idea is to war-drive the entire building in order to create a radio map between a physical location and its Wi-Fi/GSM fingerprints measured from nearby access points and base stations. Users can then pinpoint their locations by comparing their measured signal strength in the map. This approach requires a full Wi-Fi coverage (i.e., at any point, the signals of at least one Wi-Fi access point must be present), which may not be realistic in real worlds. However, the main drawback is that war-driving is both time-consuming and labor-intensive for large indoor areas. A cheap and scalable solution is desirable.

The recent advance of sensors embedded in smartphones has motivated a novel sensor-assisted localization approach [6], [7]. The accelerometer and compass can be used to measure the walking distance and direction of a mobile user. The user's location can be easily obtained by comparing the user moving trace and the map. However, sensors such as accelerometer and compass are highly noisy [12]. The user moving trace will increasingly diverge from the actual trace. Hence it requires careful calibration through certain fixed beacons. Crowdsourcing has been also used to reduce the war-driving effort [2], [17]. These works rely on detecting user activities using sensors such as accelerometer. However, to ensure reliable detection, they typically require user-specific training which is costly, and the high sampling frequency which may drain the battery power quickly. In addition, the detection may be often interrupted by users making or receiving phone calls. A new fingerprinting approach based on magnetometer sensor on smartphones has been proposed [5], [11] recently. The abnormalities of the magnetic field can be used as fingerprints for indoor localization. While this approach shares a similar idea as Wi-Fi fingerprinting, but they need even more war-driving cost.

With the recent barometer sensor embedded in many android phones and the coming iPhone 6, it opens a good opportunity for floor localization. However, existing barometer sensors are not perfect, the error of a reading for the same floor level typically varies from one to three levels. Furthermore, the readings are highly affected by the surrounding environments such as temperature and humidity, which may change from time to time. Muralidharan's most recent paper [14] study on the properties of mobile-embedded barometers across a number of buildings. He concludes that it is difficult to use the barometer to determine the actual floor that a user is on.

In this paper, we propose a novel Floor Localization approach (F-Loc) based on Wi-Fi fingerprinting and mobile phone sensing. While F-Loc leverages on Wi-Fi signal strength to build the fingerprinting map, it requires neither war-driving nor prior knowledge of the building. In addition, unlike RADAR [3] and PlaceLab [13], F-Loc does not require a full Wi-Fi coverage, making F-Loc more practical for real usage. First, since taking elevators is the most common way to travel in a multi-floor building and it shows a very clear pattern on acceleration sensor readings, we first recognize these activities using acceleration data. We then collect the Wi-Fi samples at each entrance and exit areas of an elevator, and cluster these Wi-Fi samples. By knowing floor change, we are able to order these clusters from the lowest to the highest, and each of them corresponds to a certain floor level. Finally, we expand these Wi-Fi clusters to include Wi-Fi samples for each floor using an expanding algorithm. To use F-Loc, a user scans a few Wi-Fi samples and queries the Wi-Fi map to get the current floor level which best matches these samples. In summary, we make following contributions:

1) We propose a novel floor localization approach to identify the floor level on which a mobile user is located. Compared to traditional Wi-Fi fingerprinting based approaches [3], [18], F-Loc requires neither war-drive nor the prior knowledge of the building, and the minimized need of infrastructure support making our approach more scalable.
2) By clustering Wi-Fi segments, F-Loc is able to work in low Wi-Fi coverage situations, which is more realistic as compared to other similar approaches. We only detect user taking elevators using acceleration data, which is more lightweight and efficient than other crowdsourcing based approaches [2], [17].
3) We conduct both extensive simulations and a field study to analyze the performance of F-Loc. The simulation use real Wi-Fi data from 3 different buildings and showing that our approach can get an accuracy of more than 95% to locate user in $\pm$ 0 level and 98% in $\pm$1 level. The field study includes 20 volunteers for 5 days shows that F-Loc can get an accuracy of 98% in $\pm$ 0 level.

The rest of this paper is organized as follows. Section 2 is the overview, followed by the detailed design. Section 3 describes our evaluation. Section 4 discusses the related work, and finally, Section 5 concludes the paper.

2 System Design

We give an overview of F-Loc in this section, as shown in Fig. 1. The system operates in two phases. In the first phase, F-Loc builds the Wi-Fi map automatically, as illustrated by the solid line arrows in Fig. 1. When a user travels up and down in the building, the mobile client software collects acceleration data and Wi-Fi signal strength. The activity recognition algorithm which runs on the mobile phone recognizes the activities of taking elevators up and down. The recognized activities, together with the Wi-Fi signals, will be uploaded to the cloud server as a user trace. The server runs the algorithms to find the total floor levels and generate the Wi-Fi map. The algorithms run incrementally. With enough traces collected, F-Loc is able to obtain the full Wi-Fi map for the entire building.

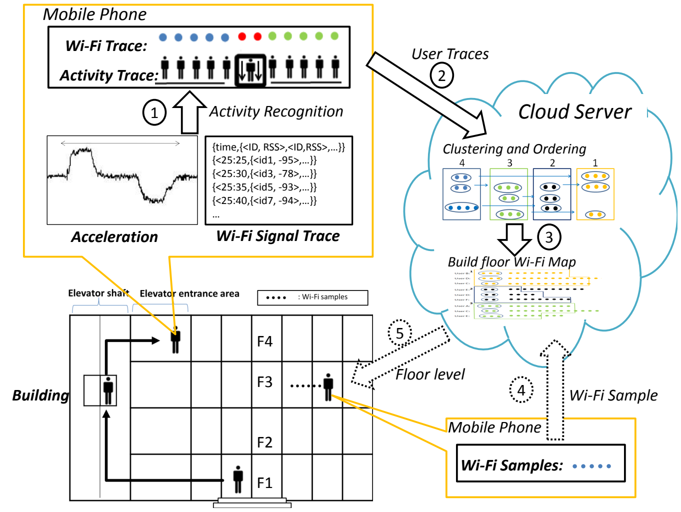
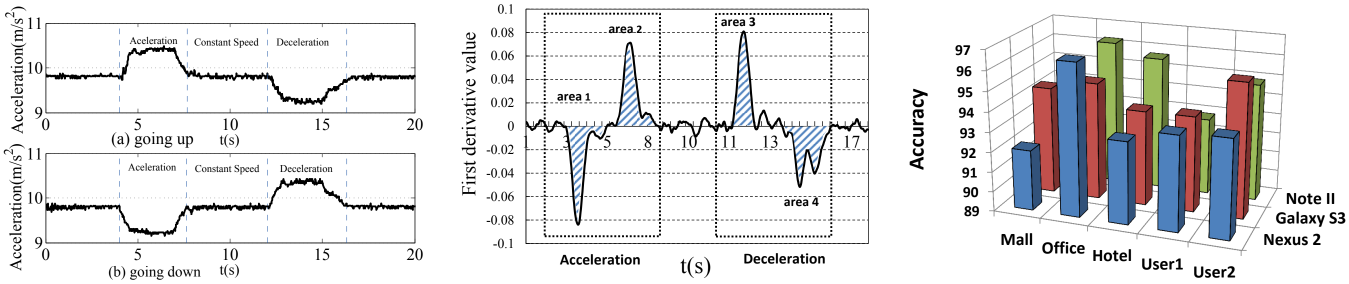


Fig. 1. Overview

To recognize the activities of taking elevators, we use acceleration data. For each taking elevator activity, based on the time detected when getting in and out, we obtain two groups of Wi-Fi samples, one at the entrance and the other at the exit area of the elevator. We now know which group is on a higher floor based on the recognition result. Using a clustering algorithm, we cluster the groups to generate a larger cluster for each floor, which consists of all the samples at the entrance or exit area near the elevator. We then sort all the clusters by the order of the groups in the clusters. The ordered clusters will have a one-to-one correspondence to the floor levels. Since each cluster has been mapped into a unique floor level, we obtain the Wi-Fi map at the entrance and exit area of the elevator for each floor level. In order to generate the full Wi-Fi map which covers the entire building, we design an algorithm to gradually expand the cluster using the Wi-Fi trace collected from users. The algorithm runs incrementally with new user traces added. With enough user traces, F-Loc is able to generate the full Wi-Fi map which covers the entire areas for each floor. It is worth knowing that the first phase to generate the map is once for all.

In the second phase, as illustrated by the dotted line arrows in Fig. 1, a user scans the Wi-Fi signal strength



(a) Acceleration readings when taking the elevator

(b) First derivative value of the readings when elevator going down

(c) Floor-change detection accuracy

Fig. 2. Floor-change detection

samples nearby, and queries the cloud server by these samples. Upon receiving the query samples, the cloud server runs the algorithm to find the best match and replies the user with the current floor level.

2.1 Floor-change Detection

Figure. 2(a) shows the acceleration readings when a person takes the elevator up and down by two floor levels. The acceleration readings show clear signatures when taking the elevator. Each travel process consists of three stages - acceleration, moving at a constant speed, and deceleration. The acceleration readings at the second stage keep at about 9.8 due to the gravity. For raw acceleration readings, we first filter the noise, and then smooth the values with a reasonable window size. We then extract the first derivative of the readings and the resulting curve is shown in Fig. 2(b) (elevator going down). We can see from the figure that the change of acceleration readings is transformed to a trough and crest pair when accelerating and a crest and trough pair when decelerating.

To detect the floor change activity, we calculate the area size of each crest or trough. If it meets certain conditions, an elevator change-floor activity is detected. In detail, each area is defined as a continuous and closed region formed by the x axis and the curve. The region is located below or upon the x axis, which should meet the following conditions: 1) For elevator going down, the order is trough-crest and crest-trough, in contrast with going up, 2) Lasted time between 5 and 60 seconds, 3) For all 4 areas, area size are the same and bigger than a threshold. After recognition, we define a detected activity of taking elevator as $A = \{st, et, dir\}$, where st and et is the start and stop time of the elevator, and dir is the direction of the elevator. The user's moving trace can then be defined as $MTrace =< A, \ldots >$. The activity detection is done in the mobile phone and the result $MTrace$ is stored and uploaded to the cloud server. We conducted experiments with two users using three different smartphones under real-life situations in three different buildings. Figure 2(c) shows the accuracy of detecting floor changes is about 94%.

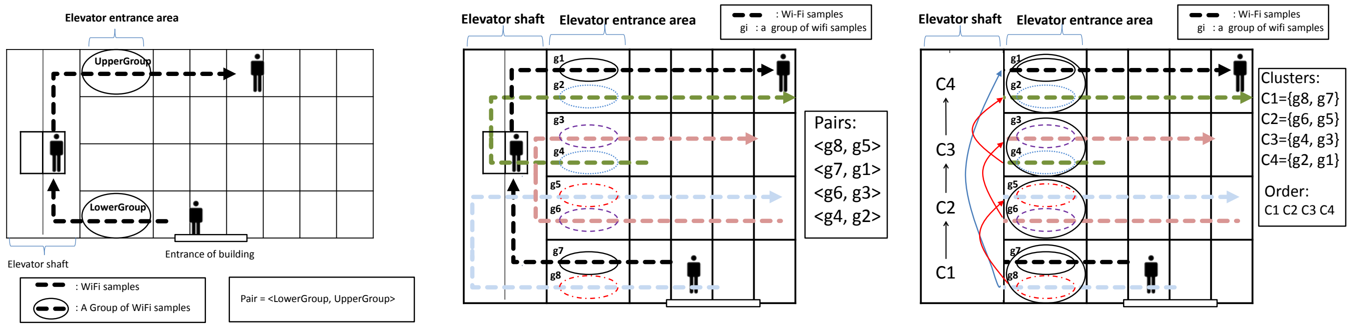
2.2 Wi-Fi Sample Groups

Other than performing activity recognition, a mobile client also scans and samples the Wi-Fi signal strength every three seconds. The structure of a Wi-Fi sample P is represented by $P =< t, \{< ID, RSS > \ldots\} >$, where P is the Wi-Fi sample, t is the time performing the scan, ID is the identity of a Wi-Fi access point, and RSS is the received signal strength of Wi-Fi. Each Wi-Fi sample contains a series of the $< ID, RSS >$ pairs, depending on the number of Wi-Fi access points scanned. The consecutive Wi-Fi samples ordered by time form the Wi-Fi trace, which is represented by $WTrace =< P, \ldots >$. Figure 3(a) shows a user's Wi-Fi trace in the building, and the dotted lines show the Wi-Fi trace, each dot represents a Wi-Fi sample. The mobile client then uploads the results of activity recognition (i.e., $MTrace$) and the Wi-Fi trace (i.e., $WTrace$) to the cloud server.

On receiving the traces, the server obtains many pairs of Wi-Fi sample groups from $WTrace$ according to the detected activities in $MTrace$. For example, given the following $WTrace$ of user a,

$$WT_a =< P_0, P_1, \ldots, P_{32}, P_{33}, P_{34}, \ldots, P_{100}, P_{101} >$$

and $MTrace$ $MT_a =< A_1, A_2, A_3 >$. The elevator stopped twice during user a's travel to the destination floor (note that we know this by examining the time interval between A_1 and A_2, A_2 and A_3, respectively). So the time he enters into and exits from the elevator is $A_1.st$ and $A_3.et$, respectively. Based on the time stamps of the traces, we found that $P_{32}.t < A_1.st < P_{33}.t$ and $P_{34}.t < A_3.et < P_{35}.t$. We can infer that $LowerGroup = \{P_{28}, P_{29}, P_{30}, P_{31}, P_{32}\}$ is scanned when the user is waiting for elevator, and $UpperGroup = \{P_{36}, P_{37}, P_{38}, P_{39}, P_{40}\}$ is scanned after the user exits from the elevator, and $UpperGroup$ is on a higher floor than $LowerGroup$, as shown in Fig. 3(a). In F-Loc, we obtain five Wi-Fi samples at each entrance/exit area for each taking elevator activity, which are represented by the $LowerGroup$ and $UpperGoup$, respectively, when elevator is going up and vise versa. So we get an ordered pair $Pair =< LowerGroup, UpperGroup >$ where $LowerGroup$ is the group on the lower floor and



(a) Obtaining an ordered pair (b) Form all ordered pairs from user traces (c) Clustering and pruning to find the Wi-Fi map in the elevator area

Fig. 3. Clustering and pruning

$UpperGroup$ is the one on the higher floor.

2.3 Clustering and Pruning

With more users involved, we obtain a series of ordered Wi-Fi group pairs. Each pair contains two groups of Wi-Fi samples scanned at the entrance/exist area of the elevator, as shown in Fig. 3(b). For the same entrance area, these Wi-Fi samples will have similar signal strength from nearby Wi-Fi access points. We then cluster these groups so that the Wi-Fi groups belonging to the same entrance/exit area will be put into the same cluster. Using this low and up relation among groups we can order these clusters, as shown in Fig. 3(c). We show the details below.

We use the hierarchical clustering algorithm called CURE [9]. Initially, each Wi-Fi group is a cluster, and in each step it merges two closest clusters until a certain number of clusters are formed. The CURE algorithm is less sensitive to outliers, and it fits in our situation where there may exist outliers. However, the CURE algorithm cannot be directly applied because the resulting number of clusters f is unknown. In F-Loc, we adapt the CURE algorithm and focus on designing the distance function and determining when to stop clustering.

In the design of the distance function, we use the Euclidean distance to calculate the distance between two Wi-Fi samples. Note that for each Wi-Fi sample at the entrance/exit area of an elevator, there may exist different Wi-Fi access points. The distance function between P_i and P_j is computed as

$$Distance(P_i, P_j) = \sqrt{\sum_{k=1}^{n}(P_i.ID_k - P_j.ID_k)^2}$$

where $P_i.ID_k$ is the RSS value of ID_k, and ID_k represents the ID of a Wi-Fi access point.

The computation of the Euclidean distance follows the two rules below:

***Rule** 1:* If a Wi-Fi access point ID_j present in P_j, but not in P_i, we add ID_j to P_i, and set its RSS value to -100, meaning that the received signal strength of ID_j is minimum in P_i.

Formally, given that *1)* $L_1 : ID_j \notin P_i$; *2)* $L_2 : ID_j \in P_j$; and *3)* $L_3 : P_i.ID_j = -100$. Rule.1 is then formulated as follows:

$$R1 : L_1 \wedge L_2 \rightarrow L_3.$$

***Rule** 2:* If the groups of the two Wi-Fi samples come from an ordered pair, meaning the two samples are scanned from two different floors, the distance of the two samples are set to infinity.

Formally, given that *1)* $L_4 : \exists pair =< g_i, g_j >$; *2)* $L_5 : P_j \in g_j$; *3)* $L_6 : P_i \in g_i$; and *4)* $L_7 : P_i \in g_i$. Rule.2 is then formulated as follows:

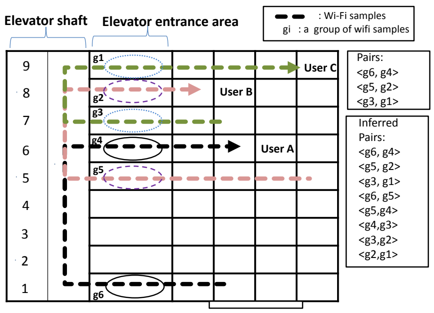
$$R2 : L_4 \wedge L_5 \wedge L_6 \rightarrow L_7.$$

During clustering, we have two observations. First, if after we merge two clusters into a new cluster, and if exists an ordered pair whose groups are all in this new cluster, we reach a contradiction and we should not merge these two clusters. In addition, all clusters from the elevator area on the same floor should contain Wi-Fi signals from similar access points that appear most often. So if two clusters contain signals from different access points, we should not merge them. Based on the observations, we have the following rules.

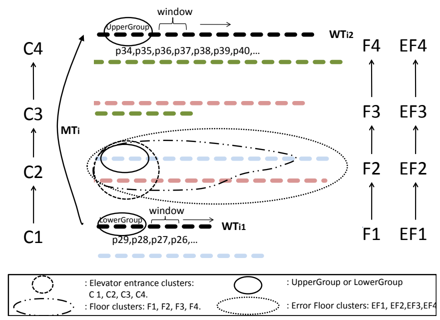
***Rule** 3:* For an ordered pair, the distance of cluster c_i and c_j is set to infinity if the following holds. $LowerGroup \in c_i \wedge UpperGroup \in c_j$.

***Rule** 4:* If in two clusters, there are different access points appeared often, the distance of the two clusters is set to infinity.

The above rules have been used in the clustering algorithm. The algorithm first finds the nearest two clusters, then check the two clusters using the above rules, after this, if their distance were not set to infinity, it will merge the two clusters. It iterates until the distance of the nearest two clusters is infinity, and then output the result clusters.



(a) Inferring Wi-Fi ordered pairs



(b) Cluster expending

Building Type	Floor levels	Average Wi-Fi coverage	Elevators in area1 normal	Elevators in area2 freight	Each elevator area Wi-Fi samples	Each floor Wi-Fi samples
Shopping Mall	5	4	3	2	289	1256
Office Building	10	4	2	1	194	809
Hotel Apartment	20	7	4	1	143	445

(c) Building information in simulation

Fig. 4. Build the Wi-Fi map

After clustering, ideally we get f clusters whereas each cluster corresponds to a floor level of the building. However, in reality, we may obtain more than f clusters due to noise. Hence, we perform cluster pruning to remove noisy clusters. First we filter out the smallest clusters. We then build a directed acyclic graph (DAG) of clusters, and find the longest path in the DAG which has an one-to-one mapping to the floor levels. The details are illustrated as follows.

We first order the clusters using the following rule. Given that *1)* $L_8 : \exists g_i \in C_m$; *2)* $L_9 : \exists g_j \in C_n$; *3)* $L_{10} : \exists pair_i =< g_i, g_j >$; and *4)* L_{11}: C_m is lower than C_n . where g_i and g_j are two Wi-Fi sample groups, C_m and C_n are two clusters, and p_i is an ordered Wi-Fi group pair. The rule is then formulated as follows.

$$R : L_8 \wedge L_9 \wedge L_{10} \rightarrow L_{11}.$$

In reality, due to noise, there may exist ordered pairs showing that C_m is lower or higher than C_n at the same time. In this situation, we vote for majority to determine which one is lower. By ordering clusters, we are now ready to build a weakly connected directed acyclic graph, in which each vertex represents a cluster and each directed edge represents the cluster order, pointing from a lower level to a higher level. Finally, we choose the longest path in the DAG, and the vertices in this path have a one-to-one correspondence to the floor levels. Note that if there are more than one longest paths, we can merge them based on the following rule. Since the clusters in the longest path have a one-to-one correspondence to the floor levels, the clusters which correspond to the same floor level from different paths can be merged. As a result, we obtain a unique, longest path. We now finished building the Wi-Fi map which contains the Wi-Fi signatures in the entrance/exit area of the elevator on each floor. Note that, if we build more than one weakly connected DAG, meaning that the building has more than one elevator area, we handle the graphs independently in the same way.

2.4 Inferring Wi-Fi Ordered Pairs

As described in Section 2.3, we use ordered Wi-Fi group pairs to determine the order of the clusters (i.e., which one is higher or lower). The more ordered pairs we collect, the faster the ordered clusters we obtain. We observe from $MTrace$ that the activities of taking elevators by different users have some overlaps. These overlaps imply that some users appear in the same elevator (i.e., they encounter each other) for a period of time. Based on this observation, we can interrelate the activity traces of different users, and use their temporal relation to sort the Wi-Fi groups. In this way, we are able to infer and obtain more ordered pairs, as shown in Fig. 4(a).

2.5 Cluster Expanding

After clustering and pruning, we get one cluster for each floor level. Each cluster contains the Wi-Fi signatures at the extrance/exit area of the elevator on each floor. We now expand each elevator-area cluster (C_i) to its corresponding floor-level cluster (F_i) which consists of the Wi-Fi signatures for the entire area of floor i using the Wi-Fi traces we collected, as shown in Fig. 4(b). The main idea is to gradually expand each C_i, by adding a few Wi-Fi samples following the samples of the groups (i.e., all $UpperGroup$ and $LowerGroup$ in C_i) in the Wi-Fi segments. We add five samples each time until we find these samples do not belong to floor i.

The details of cluster expanding work as follows. We expand all the $LowerGroup$ and $UpperGroup$ of each C_i gradually. In each step, we expand all the $LowerGroup$ and $UpperGroup$ by a sliding window. The window is set to 5 Wi-Fi samples. The expanding continues for all the samples until the Wi-Fi samples chunked by a window do not belong to this floor (i.e., C_i). First, for each cluster C_i, we generate a temp floor cluster EF_i, as shown in Fig. 4(b). EF_i contains all the Wi-Fi segments in floor i. We then get the Wi-Fi samples by a sliding window. For each window of Wi-Fi samples, we compute the distance between these Wi-Fi samples to each EF_i, and find the nearest cluster EF. If these Wi-Fi samples and cluster EF belong to the same floor, the expanding process continues, otherwise it stops. The main idea of the algorithm of finding the nearest cluster is that, for all Wi-Fi access points in the window of Wi-Fi samples, find a cluster EF_i they appear most frequently, the EF_i

is then the nearest cluster. In summary, we expand each elevator-area cluster C_i to floor-level clusters F_i, and finally, we build the Wi-Fi map for the entire building. With more users involved in data collection, the coverage of the Wi-Fi map gets increased until the full coverage. The Wi-Fi map will be stored in the cloud server for users to access the floor localization service.

2.6 Accessing F-Loc

To access F-Loc, a mobile user travelling in the building first scans a few Wi-Fi samples (e.g., k samples), and sends them to the cloud server. The server runs the $findMyFloor$ algorithm, and replies the user with the floor level. It uses k Wi-Fi samples to find the nearest floor cluster F_i, and the floor level of F_i is the result. The choose of k depends on the Wi-Fi coverage in the building. Low Wi-Fi coverage needs more samples, and high Wi-Fi coverage needs less samples. In our studies, described in Section. 3, k = 3 is a reasonable setting for the building where we ran our field trial.

3 Evaluation

3.1 Simulation Methodology

We design a simulator to evaluate the efficiency and scalability of F-Loc. We use real Wi-Fi data by collecting Wi-Fi signal strength from three real-world buildings (i.e., a shopping mall, an office building, and a hotel). Figure 4(c) describes the detailed information of each building, and Wi-Fi signal strength samples collected. For each building, we have three users and three different phones, we collect Wi-Fi samples for each floor, and divide the samples into two sets—one for samples at the elevator entrance/exit areas, and the other for samples at all other areas of the floor. We collect a Wi-Fi signal strength sample every three seconds. For each entrance/exit area of an elevator, we select five consecutive samples. The number of Wi-Fi samples can be adjustable according to the size of the entrance/exit of an elevator. Bigger entrance/exit area requires more samples. Our experiments show that taking five samples at each entrance/exit area works well in reality. These Wi-Fi sample sets will be used in our simulation.

The simulator models the process of taking the elevator up and down in a multi-floor building. It works as follows. The simulation process is divided into cycles, and each cycle simulates the process that the elevator goes up from the ground floor, with people entering and leaving the elevator from or to any levels, until the elevator is empty. Based on the elevator taking simulation results, we can get the ordered pairs of $LowerGroup$ and $UpperGroup$ used for clustering and ordering. In real situations, the activity recognition algorithm may not work perfectly. To make the simulation process more realistic, we add more errors to the floor change recognition process.

The parameters of the simulator are listed as follows. *1*) *miss detection*: an activity of taking the elevator occurs, but it is not detected. *2*) *not exist*: an activity of taking the elevator is detected, but actually it is not occurred. *3*) *wrong detection*: an activity of taking the elevator is detected wrongly (e.g., activity direction, start and end time). *4*) *Wi-Fi coverage*: the average number of Wi-Fi access points obtained in each scan. *5*) *user sample size*: the number of times users taking the elevator. A user sample is defined when a user gets in and out of the elevator. The performance metrics used in the paper are summarized as follows. *1*) *Floor accuracy*: it is defined as an accuracy of detecting the correct floor level for a mobile user, and it is measured by ± 0 level, ± 1 level, or ±2 levels. *2*) *Number of floors detected*: it is defined as the number of floor levels detected by F-Loc after the clustering and pruning.

3.2 Simulation Results

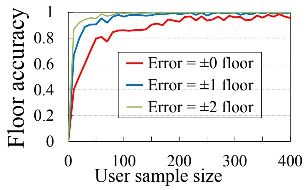
Figures 5(a) and 5(b) show F-Loc achieves *floor accuracy* of about 95% in ± 0 level, 98% in ± 1 level, and 100% in ±2 levels with 400 user samples in a 5-floor building, and 800 user samples in a 10-floor building. Figure 5(d) shows an accuracy of 90%, 96%, and 100%, respectively, with 1500 user samples in a 20-floor building. Figure 5(a), 5(b) and 5(c) show the *floor accuracy* in the shopping mall and the office building when the *Wi-Fi coverage* is 4 and 2. From the real Wi-Fi data we collected, the average Wi-Fi coverage for both the shopping mall and the office building is 4. To test F-Loc under low Wi-Fi coverage situations, we reduce the *Wi-Fi coverage* by to randomly remove some access points in the Wi-Fi samples until the average *Wi-Fi coverage* becomes 3 and 2.

Figures 5(e) and 5(f) show that lower *Wi-Fi coverage* lead to lower accuracy, but F-Loc still achieves an accuracy of more than 85% in ± 0 level even the average Wi-Fi coverage is only 2. Figures 5(g), 5(h), and 5(i) show the *floor accuracy* with different settings of *wrong detection* in a 10-floor building. Figure 5(g) shows the accuracy in ± 0 level with a *wrong detection* rate of $0\%, 5\%, 10\%, 15\%$, 50%. Higher *wrong detection* rate leads to lower accuracy, but F-Loc keeps high accuracy with *wrong detection* rate lower than 15%.

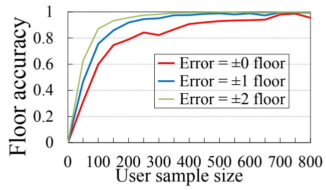
Figure 5(j), 5(k), and 5(l) shows the correct *number of floors detected* when the *Wi-Fi map accuracy* is higher than 75%. It demonstrates that the accuracy of F-Loc finding the clusters with one-to-one mapping to the floors when at least 75% samples in each floor cluster belongs to the floor. Each figure shows the accuracy with different *Wi-Fi coverage* settings. A higher accuracy of *number of floors detected* implies a higher floor accuracy.

3.3 Field Study

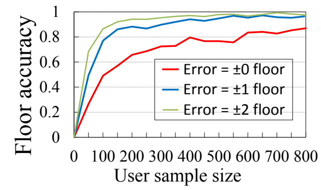
To evaluate F-Loc under the real-world situations, we conduct a field study which involves twenty users for five days in a 10-floor building, the floor plan is show in Fig. 6. They use their own smartphones (e.g., Samsung, HTC, and Moto) which are used in their daily lives.



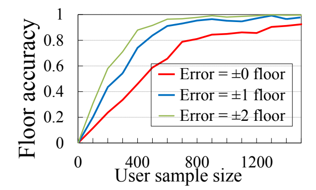
(a) Floor accuracy in 5-floor mall with a Wi-Fi coverage of 4



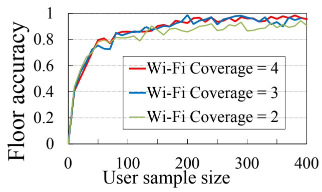
(b) Floor accuracy in 10-floor office with a Wi-Fi coverage of 4



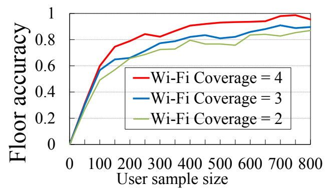
(c) Floor accuracy in 10-floor office with a Wi-Fi coverage of 2



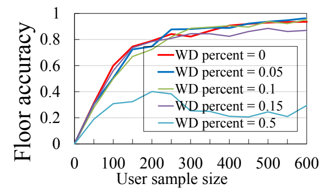
(d) Floor accuracy in 20-floor hotel with a Wi-Fi coverage of 7



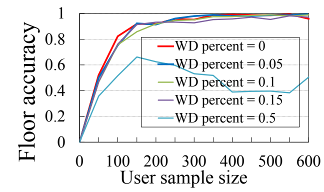
(e) Floor accuracy in 5-floor mall with different Wi-Fi coverage



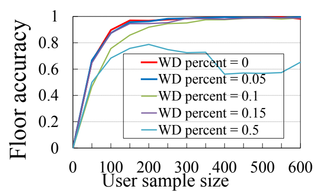
(f) Floor accuracy in 10-floor office with different Wi-Fi coverage



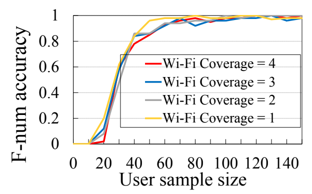
(g) Floor accuracy of ± 0 level



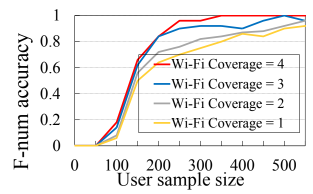
(h) Floor accuracy of ± 1 level



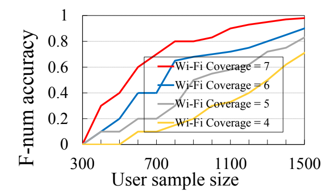
(i) Floor accuracy of ± 2 levels



(j) 5-floor shopping mall with different Wi-Fi coverage



(k) 10-floor office with different Wi-Fi coverage



(l) 20-floor hotel with different Wi-Fi coverage

Fig. 5. Simulation results (parameter setting: right detection= 70%, miss detection=19%,wrong detection=10%, not exist = 1%)

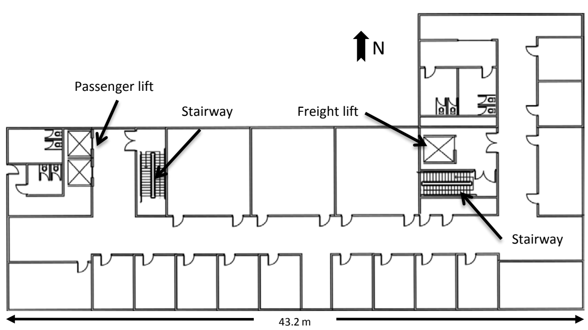


Fig. 6. Floor Plan

Each smartphone is equipped with an embedded 3-axis accelerometer and the Wi-Fi connectivity. We conduct the field study as follows. Each smartphone is installed with the data collection software we design. Once started, this software continuously collects acceleration readings at a rate of 10 samples per second, and scans Wi-Fi signal strength every 3 seconds. All the samples will be logged in a data file. The software runs in the background so that the users are still able to use their mobile phones as usual. We do not give special instructions to control their behaviors during the study, instead, all the users are told to perform their daily routines. Since F-Loc runs in the background, they are even not aware of the experiments we intend to run. We collect the data for five days, and use the data for the experiments, described as follows.

Figure 7 shows the floor accuracy of F-Loc using the real data from our field study. We have in total 274 user samples for the taking the elevator activity. To calculate the accuracy, we randomly choose 195 user samples from 274 user samples as input to build the Wi-Fi map. We also collect 200 new groups of Wi-Fi samples, each group tagged with a floor level. The Wi-Fi map is used to locate the floor of each group. This experiment runs over 100 times, and we obtain an average accuracy of 98.8% for locating a user at $\pm$ 0 level with the user sample size of 195. The result shows that F-Loc works well in real situations with high accuracy. Note that we have only 20 users for 5 days. With more users involved, we can get user samples much faster. For example, for a 10-floor office building with 600 users, we can collect at least 600 samples in a day assuming everyone takes the elevator at once a day. Compared to the result in the simulation, the field study requires fewer user samples to achieve the same accuracy. This is because we have a tight parameter setting in the simulation.

4 Related Work

Many fingerprint based techniques for indoor localization have been proposed such as [3], [15], [18]. They mainly rely on Wi-Fi signal strength, and they are capable of achieving a high accuracy in an indoor environment. However, like RADAR [3] has to war-drive the entire building in order to obtain the radio map. War-driving is very time-consuming and labor-intensive. SkyLoc [15] uses GSM fingerprints to locate a user's floor

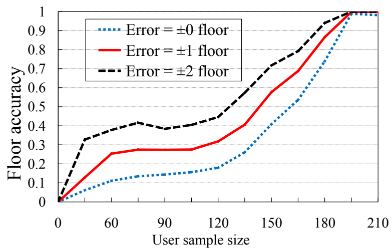


Fig. 7. Accuracy of F-Loc in the 10 floor office building

level in a multi-floor building. They report an accuracy of 73% for locating a user to the right floor, and 95% within 2 floors. But the GSM signals vary significantly in indoor environments, and the training process in SkyLoc is time-consuming. It has a poor scalability since war-driving and training are required for every building. Several recent approaches [2], [17] use crowdsourcing to construct the floor map for localization; however, they need many energy-draining sensors and they require complicate training to achieve good accuracy.

Muralidharan's most recent paper [14] study on the properties of mobile-embedded barometers across a number of buildings. But failed to solve the problem of using the barometer to determine the floor of a user. In another typical solution proposed by Wang in [16]. They track the user using barometer readings, but they need to know the initial floor of the user and the height of all floors. Furthermore, a miss or wrong detection of the floor will cause serious errors in the latter localization. Sensor-assisted localization methods [6], [7] have been proposed, making use of embedded sensors available on smartphones. These systems typically use accelerometer and electronic compass. However, careful calibration is needed from time to time due to the limitations of the sensoring technology. Escort [6] leverages on fixed beacons for calibration, and CompAcc [7] makes use of possible walking paths extracted from Google Maps [1]. FTrack [10] detect user activities of changing floors, and track their floor levels based on their initial locations. It requires neither infrastructure nor training. The main problem of this approach is that they cannot handle some practical issues such as different user walking patterns and a variety of ways to carry/use mobile phones, which may affect the accuracy and limit the feasibility. F-Loc detects user activities of changing floor by elevator only, but it has no strict assumption of users walking pattern or the ways to carry/use mobile phones.

5 Conclusion and Future Work

This paper presents a novel, scalable floor localization scheme. Leveraging on mobile phone sensing and crowdsourcing, F-Loc requires neither any infrastructure nor any prior knowledge of the building. Different from similar approaches, F-Loc does not require war-driving, and it works well in buildings with low Wi-Fi coverage. F-loc do not rely on high accuracy of activity recognition, making it more realistic for real-world deployment. We collect real Wi-Fi data from three different buildings, and conduct both simulation and field studies to demonstrate the performance, scalability, and robustness of F-Loc. For our future work, we will further improve F-Loc by enhancing the clustering and pruning algorithms. We also plan to offer F-Loc as a free service to Google's play store and the Apple store for public use, and test F-Loc under real-life situations.

6 Acknowledgment

This work was supported by the China National High-tech R&D 863 program (2012AA011205), the NSFC under Grants 91318301, 61373011, 61321491.

References

[1] Google maps.

[2] M. Alzantot and M. Youssef. Crowdinside: automatic construction of indoor floorplans. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 99–108. ACM, 2012.

[3] P. Bahl and V. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Israel.*

[4] G. Bruner II and A. Kumar. Attitude toward location-based advertising. *Journal of Interactive Advertising*, 7(2):3–15, 2007.

[5] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman. Indoor location sensing using geo-magnetism. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 141–154. ACM, 2011.

[6] I. Constandache, X. Bao, M. Azizyan, and R. Choudhury. Did you see bob?: Human localization using mobile phones. In *Proceedings of the 16th annual international conference on mobile computing and networking*. ACM, 2010.

[7] I. Constandache, R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *INFOCOM2010*. IEEE.

[8] S. Gaonkar, J. Li, R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *Proceeding of the 6th international conference on mobile systems, applications, and services*. ACM, 2008.

[9] S. Guha, R. Rastogi, and K. Shim. Cure: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM, 1998.

[10] Y. Haibo, G. Tao, Z. Xiaorui, et al. Ftrack: Infrastructure-free floor localization via mobile phone sensing. *Percom*, 2012.

[11] T. G. HongWei Xie and J. L. XianPing Tao, Haibo Ye. M-loc: A practical magnetic fingerprinting approach to indoor localization using smartphones. *UbiComp 2014: Ubiquitous Computing*.

[12] C. Jekeli. *Inertial navigation systems with geodetic applications*. Walter De Gruyter Inc, 2000.

[13] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place lab: device positioning using radio beacons in the wild. *Pervasive Computing*, pages 301–306, 2005.

[14] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal. Barometric phone sensors–more hype than hope! ACM HotMobile 2014.

[15] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara. The skyloc floor localization system. in PerCom 2007.

[16] H. Wang, H. Lenz, and Szabo. Fusion of barometric sensors, wlan signals and building information for 3-d indoor/campus localization. In *(MFI 2006), S*, 2006.

[17] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. ACM, MobiSys 2012.

[18] G. Zaruba, M. Huber, F. Kamangar, and I. Chlamtac. Indoor location tracking using rssi readings from a single wi-fi access point. *Wireless networks*, 13(2):221–235, 2007.