Smartphones based Crowdsourcing for Indoor Localization

Chenshu Wu, *Student Member, IEEE,* Zheng Yang, *Member, IEEE,* and Yunhao Liu, *Senior Member, IEEE*

Abstract—Indoor localization is of great importance for a range of pervasive applications, attracting many research efforts in the past decades. Most radio-based solutions require a process of site survey, in which radio signatures of an interested area are annotated with their real recorded locations. Site survey involves intensive costs on manpower and time, limiting the applicable buildings of wireless localization worldwide. In this study, we investigate novel sensors integrated in modern mobile phones and leverage user motions to construct the radio map of a floor plan, which is previously obtained only by site survey. Considering user movements in a building, originally separated RSS fingerprints are geographically connected by user moving paths of locations where they are recorded, and they consequently form a high dimension fingerprint space, in which the distances among fingerprints are preserved. The fingerprint space is then automatically mapped to the floor plan in a stress-free form, which results in fingerprints labeled with physical locations. On this basis, we design LiFS, an indoor localization system based on off-the-shelf WiFi infrastructure and mobile phones. LiFS is deployed in an office building covering over 1600m^2, and its deployment is easy and rapid since little human intervention is needed. In LiFS, the calibration of fingerprints is crowdsourced and automatic. Experiment results show that LiFS achieves comparable location accuracy to previous approaches even without site survey.

Index Terms—Indoor Localization, Floor Plan, RSS Fingerprint, Smartphones, Site Survey

✦

1 INTRODUCTION

The popularity of mobile and pervasive computing stimulates extensive research on wireless indoor localization. Many solutions are introduced to provide room-level location-based services, for example, locating a person or a printer in an office building.

The majority of previous localization approaches utilize Received Signal Strength (RSS) as a metric for location determinations. RSS fingerprints can be easily obtained from most off-the-shelf wireless network equipments, such as WiFi- or ZigBee-compatible devices. In these methods, localization is divided into two phases: training and operating. In the first stage, traditional methods involve a site survey process (a.k.a. calibration), in which engineers record the RSS fingerprints (e.g., WiFi signal strengths from multiple Access Points, APs) at every location of an interested area and accordingly build a fingerprint database (a.k.a. radio map) in which fingerprints are related with the locations where they are recorded. Next in the operating stage, when a user sends a location query with his current RSS fingerprint, localization algorithms retrieve the fingerprint database and return the matched fingerprints as well as the corresponding locations.

Although site survey is time-consuming, labor-intensive, and vulnerable to environmental dynamics, it is inevitable for fingerprinting-based approaches,

since the fingerprint database is constructed by locationally labeled fingerprints from on-site records. In the end of 2011, Google released Google Map 6.0 that provides indoor localization and navigation available only at some selected airports and shopping malls in the US and Japan. The enlargement of applicable areas is strangled by pretty limited fingerprint data of building interiors. If ordinary mobile phone users are able to participate in site survey by contributing their data, the burden of indoor map providers like Google will be effectively reduced.

The development of wireless and embedded technology has fostered the flourish of smartphone market. Nowadays mobile phones possess powerful computation and communication capability, and are equipped with various functional built-in sensors. Along with users round-the-clock, mobile phones can be seen as an increasingly important information interface between users and environments. These advances lay solid foundations of breakthrough technology for indoor localization.

On this basis, we reassess existing localization schemes and explore the possibility of using previously unavailable information. Considering user movements in a building, originally separated RSS fingerprints are geographically connected by user moving paths of locations where they are recorded, and they consequently form a high dimension *fingerprint space*, in which the distances among fingerprints, measured by footsteps, are preserved. In addition, we reform the floor plan of a building to the *stress-free floor plan*, a high dimension space in which the distance between two locations reflects their walking distance according to the real floor plan. The spatial similarity of stress-free floor plan and fingerprint space enables fingerprints labeled with real locations, which would

- *A preliminary version of this article appeared in Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (ACM MobiCom 2012).*
- *C. Wu, Z. Yang and Y. Liu are with the School of Software and TNLIST, Tsinghua University. C. Wu is also with the Department of Computer Science and Technology, Tsinghua University.*
  Email: wucs06@mails.thu.edu.cn, yangzheng@tsinghua.edu.cn, yunhao@greenorbs.com.

be done only by site survey previously. These observations motivate us to design practical, flexible, and rapidly deployed localization approaches with little human costs and intervention.

In this study, we propose *LiFS* (Locating in Fingerprint Space), a wireless indoor localization approach. By exploiting user motions from mobile phones, we successfully remove the site survey process of traditional approaches, while at the same time, achieve competitive localization accuracy. The key idea behind LiFS is that human motions can be applied to connect previously independent radio fingerprints under certain semantics. LiFS requires no prior knowledge of AP locations, which is often unavailable in commercial or office buildings where APs are installed by different organizations. In addition, LiFS' users are in no need of explicit participation to label measured data with corresponding locations, even in the training stage. In all, LiFS transforms the localization problem from 2D floor plan to a high dimension fingerprint space and introduces new prospective techniques for automatic labeling.

To validate this design, we deploy a prototype system and conduct extensive experiments in a middle-size academic building covering over 1600m^2. Experiment results show that LiFS achieves comparable location accuracy to previous approaches even without site survey. The average localization error is 5.8 meters and can be reduced to be about 2 meters by incorporating trajectory matching, while the room-level localization error is about 11%.

The rest of the paper is organized as follows. We discuss the state-of-the-art of indoor localization technology and multi-dimensional scaling in Section 2. Section 3 presents the system overview of LiFS. The construction of stress-free floor plan is introduced in Section 4. Section 5 shows how to transform RSS fingerprints into high-dimension fingerprint space. In Section 6, we promote several techniques to establish the relationship between stress-free floor plan and fingerprint space. The localization scheme is introduced in Section 7. The prototype implementation and experiments are discussed in Section 8. Design details and limitations are discussed in Section 9. We conclude the work in Section 10.

2 RELATED WORK

Wireless Localization

In the literature of indoor localization, many techniques have been proposed in the past two decades. Generally, they fall into 2 categories: fingerprinting-based and model-based.

Fingerprinting-based techniques. A large body of indoor localization approaches adopt fingerprint matching as the basic scheme of location determination. The main idea is to fingerprint the surrounding signatures at every location in the areas of interests and then build a fingerprint database. The location is then estimated by mapping the measured fingerprints against the database. Researchers have striven to exploit different signatures of the existing devices or reduce the mapping effort. Most of these techniques utilize the RF signals such as RADAR [2], Horus [41], improved upon RADAR, LANDMARC [22], Active-Campus [11], PlaceLab [16] and OIL [26]. Surround-Sense [1] performs logical location estimation based on ambience features including sound, light, color, WiFi, etc. In two recent works, FM radio [5] and Channel Frequency Response [30] are explored to use as fingerprints. All these approaches require site survey over areas of interests to build a fingerprint database. The considerable manual cost and efforts, in addition to the inflexibility to environment dynamics are the main drawbacks of fingerprinting-based methods.

Model-based techniques. These schemes calculate locations based on geometrical models rather than search for best-fit signatures from pre-labeled reference database. The prevalent log-distance path loss (LDPL) model, for instance, builds up a semi-statistical function between RSS values and RF propagation distances [6], [17]. These approaches trade the measurement efforts at the cost of decreasing localization accuracy. [35] investigates several approaches based on AP locations and radio propagation models, and reports average error greater than 5 meters. Apart from power-distance mapping, Time of Arrival (ToA) [42], Time Difference of Arrival (TDoA) [27], and Angle of Arrival (AoA) [23], [44] have brought a host of alternative perspectives to capture geometric relationship between signal transmitters and receivers.

Simultaneous Localization and Mapping (SLAM)

While the robotics and computer vision communities have developed techniques for jointly estimating the locations of a robot and a map of an environment, the nature of wireless signal strength prohibits the use of standard SLAM techniques [21], [34]. These techniques typically depend on two facts: 1) the ability to sense and match discrete entities such as landmarks or obstacles detected by sonar or laser range-finders; 2) precisely controlled movement of robots to depict discovered environments. Both of them are unreasonable for smartphone-based localization [37].

WiFi-SLAM [9] uses the Gaussian process latent variable models to relate RSS fingerprints and models human movements (displacement, direction, etc.) as hidden variables. When a small portion of RSS measurements are tagged with the real coordinates, semi-supervised localization [28] estimate the others' locations according to RSS dissimilarity. GraphSLAM [12] further improves WiFi-SLAM regarding computing efficiency and relying assumptions. Similar in leveraging human mobility, Zee [29] devises techniques for accurate dead-reckoning using smartphones and places recorded user paths into an indoor map according to the constrains imposed by the map (e.g., that a user cannot walk through a wall or other barrier marked on the map), such that wireless fingerprints are related to locations.

Different from previous SLAM solutions and [29], LiFS only measures walking steps and is free of using dead-reckoning based on noisy inertial sensors of smartphones. In the proposed solution, neither

digital compass nor gyroscope is involved. Instead, we use accelerometer (as pedometer) to record only the number of footsteps, which can be accurately measured by nowadays smartphones, with respect to the displacement and directions of users' movements. Locations are computed through the deterministic MDS method. The mapping of discovered world and the ground-truth one has not been specifically discussed in SLAM and the solution relies on global references. In contrast, LiFS exploits the geometry of fingerprint space to construct fingerprints databases.

Multidimensional Scaling

Multidimensional scaling (MDS) [4] is a set of related statistical techniques often used in information visualization for exploring similarities or dissimilarities in data. An MDS algorithm starts with a matrix of item-item dissimilarities, then assigns a location to each item in d-dimensional space, where d is specified a priori. For sufficiently small d ($d = 2, 3$), the resulting locations may be displayed in a 2D graph or a 3D structure.

Seeing inter-device distances as a metric of dissimilarity, many approaches of network localization adopt MDS as a tool for calculating the locations of wireless devices [8], [31]. For example, in wireless sensor networks, sensor nodes are capable of measuring the distances to neighboring nodes by RSS, ToA, TDoA, etc. MDS is used to assign a coordinate to each node such that the measured inter-node distances are as much preserved as possible. Some researchers propose MDS to figure out WiFi AP locations [14]. In their approach, AP-AP distances are determined by a radio attenuation model. Although being similar to our solution in terms of the usage of MDS, it is neither for user localization nor fingerprinting-based.

3 OVERVIEW

3.1 Data Collection

User participation is essential in the initial period at the online stage. Untrained users walk in a building following daily activities. Mobile phones, carried by users, collect WiFi RSS characteristics (a.k.a. RSS fingerprints or signatures) at various locations along user movement paths, and the walking distances are also recorded. Walking distances are measured as footsteps from the readings of integrated accelerometers in mobile phones. Similarly, accelerometers also infer the starting and finishing moments of user paths. LiFS harnesses the walking distance between two end-points (denoted by corresponding fingerprints) along a user path to establish the geographical relationship among fingerprints. During data collection, users can be even unaware of the collection task in which they are actually involved.

3.2 System Architecture

In this subsection, we present the system architecture of LiFS, as shown in Figure 1. The working process of LiFS consists of two phases: training and operating.
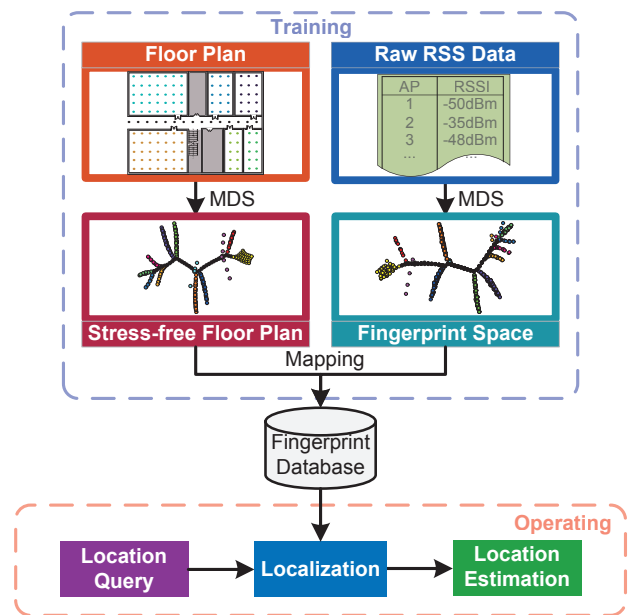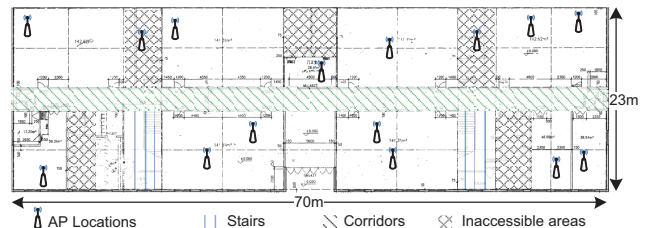

Fig. 1: System architecture.

Fig. 2: Floor plan of the experiment field.

The major output of training phase is a fingerprint database in which an RSS fingerprint and its corresponding location are associated. The fingerprint database is further used in operating phase to process location requests. We describe the training and operating phases in detail next.

Training Phase. The core task of training phase is to build the fingerprint database. We divide this task into 3 steps: (1) transforming floor plan to stress-free floor plan; (2) creating fingerprint space; (3) mapping fingerprints to real locations.

A floor plan shows a view of a building structure from above, including the relationships between rooms, spaces, and other physical features. The geographical distance between two locations in a floor plan is not necessary to be the walking distance between them due to the block of walls and other obstacles. Hence, we propose stress-free floor plan, which puts real locations in a floor plan into a high dimension space by multidimensional scaling (MDS) [4], such that the geometrical distances between the points in the high dimension space reflect their real walking distances. Through stress-free floor plan, the walking distances collected by users can be accurately and carefully utilized.

Fingerprint space is a unique component in LiFS, different from traditional approaches. According to the inter-fingerprint distances, MDS is used to create a high dimension space, in which fingerprints are represented by points, and their mutual distances
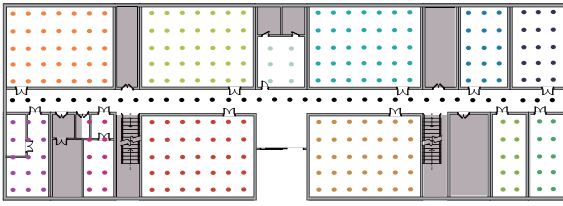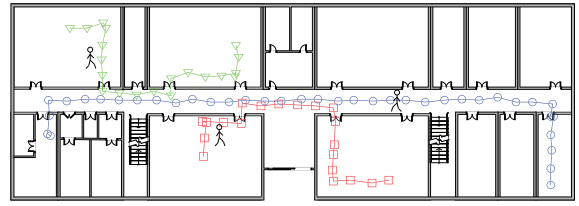
Fig. 3: Floor plan with sample locations.

Fig. 4: Moving paths.

are preserved. In traditional approaches, fingerprints are geographically unrelated, losing the possibility of building fingerprint space.

In fingerprint database, fingerprints are associated with their collecting locations (i.e., fingerprints are labeled with locations). Such associations are achieved by mapping fingerprint space (fingerprints) to stress-free floor plan (locations). In LiFS, the fingerprint database is updated continuously according to newly collected data, such that the database reflects the up-to-date radio signal distribution. As shown in Figure 1, fingerprint database, as the core component, connects training and operating phase.

Operating Phase. When a location query comes, usually an RSS fingerprint sent by a user, LiFS takes it as a keyword and searches the fingerprint database. The best matched item is viewed as the location estimation and sent back to users. To find the best matches, many searching algorithms can be used. In this design, we adopt a simple one, the nearest neighbor algorithm. More specifically, we assume that a fingerprint f is collected at the same location as f', if f' is the most similar to f in the fingerprint database. Besides the classical nearest neighbor algorithm, we also propose a continuous trajectory matching scheme to reduce the localization error caused by the fingerprint ambiguity for mobile users. In this scheme, a user's location is estimated based on his/her moving trajectory, instead of one single RSS report, by measuring successive RSSs and the accompanying mobility information when a user is moving.

4 STRESS-FREE FLOOR PLAN

In architecture and building engineering, a floor plan is a diagram, showing a view from above of the relationships between rooms, spaces, and other physical features at one level of a structure. Dimensions are usually drawn between the walls to specify room size and wall length. The floor plan of our experiment field is shown in Figure 2. The geographical distance between two locations in a floor plan does not necessarily equal to the walking distance between them due to the block of walls and other obstacles. Hence, ground-truth floor plans come into conflict with the measured distances during data collection. Figure 2 also illustrates the distance mismatch phenomenon. The walking distance of two marked locations is greatly larger than their straight-line distance since walls are not easily passed through by users.

To address the distance mismatch problem, we propose the concept of *stress-free floor plan*. We sample an area of interests at the intersecting locations of a mesh of grids in a floor plan, as shown in Figure 3.

The length l of a grid can be 1-3 meters according to the general performance of fingerprinting-based localization methods. Overmuch large or small values of l will decrease location accuracy or gain marginally or even scarcely. In our experiment, we set $l = 2$m. By calculating the distances between all pairs of sample locations, we have the distance matrix $D = [d_{ij}]$, where d_{ij} is the walking distance between two sample locations p_i and p_j in the floor plan. Using D as an input, MDS maps all p_is into a d-dimension Euclidean space. In a stress-free floor plan, the Euclidean distance between a pair of points reflects the walking distance of their corresponding locations in a real floor plan. Stress-free floor plans are often hardly embeddable in a low dimension space due to excessive distance constraints. For the convenience of observation, we set $d = 2, 3$ and the resulting stress-free floor plans in 2D and 3D visualization are shown in Figure 5 and 6, respectively, where points with the same color represent the sample locations from the same area.

5 FINGERPRINT SPACE

This section discusses the techniques for constructing fingerprint space based on the data collected by users.

5.1 Fingerprint Collection

Suppose m APs in an area A. For each location in A, the RSS fingerprint at this location can be denoted as a vector $f = (s_1, s_2, \ldots, s_m)$, where s_i is the RSS of the ith AP and $s_i = 0$ if the signal of the ith AP cannot be detected. Let d'_{ij} denote the distance between the positions of f_i and f_j. We set $d'_{ij} = +\infty$ temporarily if the distance record between f_i and f_j is not available. We measure d'_{ij} as follows. Suppose at somewhere a mobile phone records f_i; Along with walking users, it moves to another position and records f_j. In this case, d'_{ij} is the number of footsteps during the movement.

RSS fingerprints are collected during users' routine indoor movements. Users walk in a building and their mobile phones record RSS fingerprints along their walking paths , as well as the footsteps between every pairs of two consecutive fingerprints. As illustrated in Figure 4, fingerprints (denoted as squares, circles, or triangles) are recorded along three walking paths and the line segments between fingerprints indicate their distances in terms of footsteps.

After fingerprint collection, we have a set of fingerprints $F = \{f_i, i = 1 \ldots n\}$ (n is the number of records) and a distance matrix $D' = [d'_{ij}]$, both of which are essential for constructing the fingerprint space.
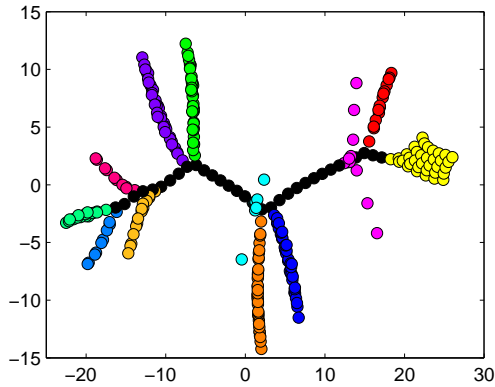
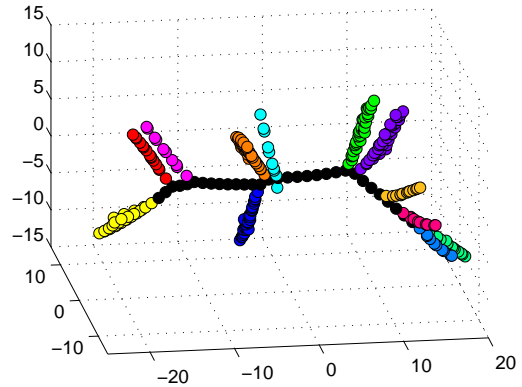Fig. 5: 2D stress-free floor plan.
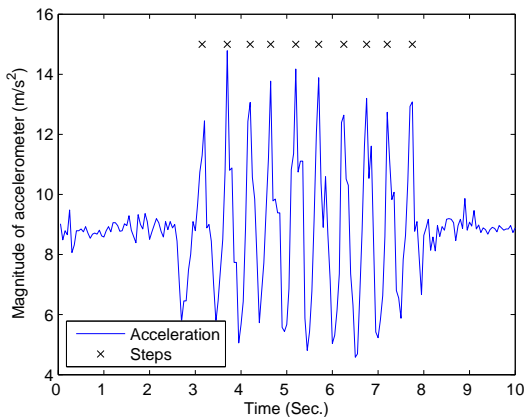

Fig. 6: 3D stress-free floor plan.

Fig. 7: The acceleration pattern for 10 steps.

5.2 Pre-processing

As user movements are usually arbitrary and ruleless, walking paths might be intersectant and accordingly the fingerprints might be overlapped. Hence data pre-processing is necessary to merge similar fingerprints, which means they are likely from the same (or very close) locations in the floor plan.

Generally, for two fingerprints $f_i = (s_1, s_2, \ldots, s_m)$ and $f_j = (t_1, t_2, \ldots, t_m)$, define RSS difference (dissimilarity) δ between f_i, f_j as follows:

$$\delta_{ij} = \parallel f_i - f_j \parallel_1 = \sum_{k=1}^{m} |s_k - t_k| \tag{1}$$

For f_i and f_j, if their dissimilarity δ_{ij} is smaller than a predefined threshold ϵ, then they are merged as a same point in the fingerprint space to be generated. Otherwise, if $\delta_{ij} > \epsilon$, f_i and f_j are treated as two different points. The determination of ϵ is based on the fingerprint samples collected at a given location (when phones are not moving). Several other works like [37], [39] adopt the similar solution as well. Hence, ϵ represents the average maximum dissimilarity of fingerprints from a distinct location, which is then feasible for merging fingerprints from the same (or close) locations and distinguishing fingerprints from different locations. In practice, the calculation of ϵ can be automatically finished by exploiting fingerprint measurements from stationary users, who can be detected via inspecting the inertial sensor data from their mobile phones.

Moreover, the raw data from accelerometer readings are pre-processed to obtain walking distance measurements. Theoretically the distance traveled can be calculated by integrating acceleration twice with respect to time. However due to the presence of noise in accelerometer readings, error accumulates rapidly and can reach up to 100 meters after one minute of operation [38].

To avoid accumulation of measurement errors, we adopt the individual step counts as the metric of walking distance instead, like a pedometer. Figure 7 shows the magnitude of acceleration during walking for ten steps. We employ a local variance threshold method [13] to detect the number of steps. The method is based on filtering the magnitude of acceleration followed by applying a threshold on the variance of acceleration over a sliding window. Step counting is accurate and in our experiments the measured steps are almost exactly what they actually are.

We understand that stride lengths vary from person to person. Previous solutions like [37] assume a fixed stride length of a person according to his weight and height, and achieve accurate results. In our solution, the variation of stride length can be efficiently alleviated through the fact that non-metric MDS can tolerate measurement errors gracefully, due to its over-determined nature [31], [32]. In addition, recently, robust trajectory estimation schemes have also been proposed for crowdsourcing-based applications, which can be incorporated in LiFSto alleviate the negative influence of crowdsourced abnormal user trajectories from multiple users [43].

5.3 Fingerprint Space Construction

To construct an accurate and informative fingerprint space, adequate fingerprints and their distance measurements are required. In our experiments, the operating phase of LiFS starts when the number of collected fingerprints reaches 10 times of the number of the sample locations in the construction of stress-free floor plans. Another possible way is to assign the first several days of LiFS' pilot run for training because
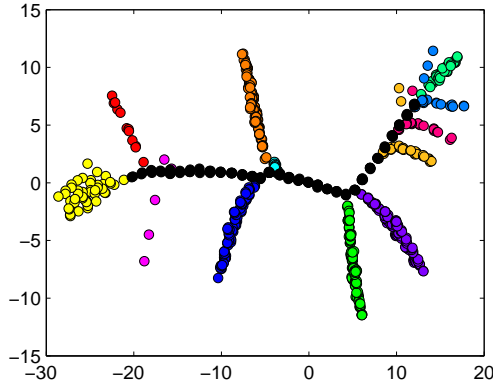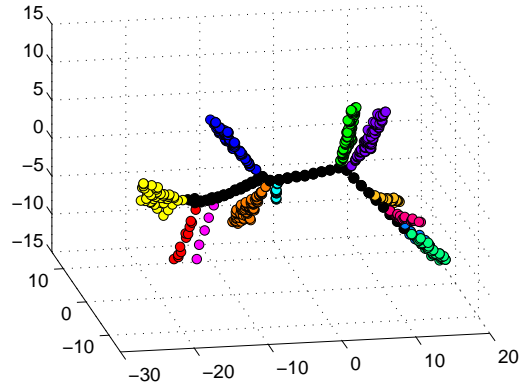
Fig. 8: 2D fingerprint space.

Fig. 9: 3D fingerprint space.

routine activities exhibit certain repetitiveness day after day. Actually the running of operating phase does not mean the end of data collection. It is reasonable to select a less conservative starting point and refine the current fingerprint database uninterruptedly in operating phase according to newly coming data.

If no user path passes through a pair of fingerprints f_i and f_j, the direct measurement of the distance d'_{ij} is unavailable. However, all user paths constitute a network of fingerprints in which f_i and f_j are connected via more than one user paths. Hence, the value of d'_{ij} can be approximated as the length of the shortest path between f_i and f_j by passing several user path segments. Note that some measured values of d'_{ij} can also be updated under this intuition. For example, if $d'_{ij} > d'_{ik} + d'_{kj}$ for some k, then d'_{ij} is updated to $d'_{ik} + d'_{kj}$. Such updates can eliminate the negative distance estimates caused by the circuitous paths or the adverse (to LiFS) user habit of pacing back and forth.

We adopt the Floyd-Warshall algorithm [10] to compute all-pair shortest paths of fingerprints. It takes $O(n^3)$ running time and n is the number of fingerprints. For convenience, we still use D' to denote the distance matrix after the above-mentioned refinements on the original D'. So far D' is dense and meaningful.

Similar to constructing stress-free floor plan, using D' as an input, MDS maps all f_i into a d-dimension Euclidean space. Figure 8 and 9 demonstrate the 2D and 3D visualization of fingerprints, respectively.

6 MAPPING

If all fingerprints correspond with the sample locations in the stress-free floor plan, we are able to label each fingerprint with a real location. Such correspondence comes from the spatial similarity between stress-free floor plan and fingerprint space.

6.1 Feature Extraction

6.1.1 Corridor Recognition

Generally speaking, corridors in a building connect all other office rooms like hubs in a network. When people walk from one room to another, they need

to pass through corridors. Such characteristics in real life are reflected in both stress-free floor plan and fingerprint space, as shown in Figure 5, 6, 8, and 9. We observe that fingerprints collected at corridors reside in core positions in fingerprint space. In terms of graph centrality [24], these fingerprints have a relatively large centrality values.

In graph theory, vertex centrality can be valued by degree, betweenness, closeness, etc [24]. In our context, we adopt the betweenness centrality to identify corridor fingerprints. Conceptually, vertices that have a high probability to occur on a randomly chosen shortest path between two randomly chosen nodes have a high betweenness. Formally, in a graph $G = (V, E)$ of vertices V and edges E, the betweenness centrality of a vertex $v \in V$ is defined as

$$B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}, \qquad (2)$$

where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through a vertex v.

Our solution first recognizes the fingerprints collected in corridors in the fingerprint space. According to the distances among fingerprints, we build the *Minimum Spanning Tree* (MST) [7] T that connects all fingerprints in F, as illustrated in Figure 10. In addition, we compute the vertex betweenness for all vertices (fingerprints) in T and then distinguish fingerprints from corridors and other areas based on a betweenness watershed. The betweenness watershed value is determined by two parameters: 1) the area ratio r_c of corridors to entire floor plan (i.e., $r_c = \text{size(corridor)}/\text{size(all)}$), which is available when generating the stress-free floor plan; and 2) the largest gap of betweenness values of fingerprints. The area ratio r_c is used to cut a feasible interval of the betweenness distribution and then the watershed is finally determined by finding the largest gap of betweenness values in the feasible interval. In this sense, the betweenness watershed is an automatically determined value that depends on the specific scenario settings. In our experiment, the resulting cumulative distribution of betweenness is shown in Figure 11. Roughly, nearly 8.6% have their betweenness larger
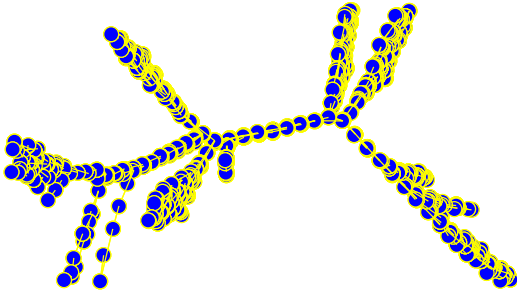
Fig. 10: MST in 3D fingerprint space.

Fig. 11: Betweenness distribution.

than 8,200; while others all less than 7,000 (shown in Figure 11). Obviously two groups of fingerprints are formed and we regard the one of larger betweenness as coming from corridors considering the structure shown in Figure 2. Let F_c denote the set of fingerprints that are estimated collected from corridors.

6.1.2 Room Recognition

Removing F_c from the fingerprint space, we observe from both Figure 8 and 9 that the remaining fingerprints form several clusters that are apparently spatially separated. To gather the fingerprints that are sufficiently close to each other, the k-means algorithm [19] (a classic clustering method) is chosen due to its computational efficiency. Thus all fingerprints in $F - F_c$ are classified into k clusters (denoted by F_{R_i}, $i = 1, 2, \ldots, k$) and in the k-means algorithm k is set to be the number of rooms in real floor plan. After clustering, all fingerprints of a same F_{R_i} are considered from the same real rooms, though we cannot tell which specific room they are from. The next subsection focuses on this mapping problem.

6.1.3 Reference Point Mapping

After characterizing corridors and rooms, we are able to establish relationships between stress-free floor plan and fingerprint space, and we think doors are the keys. Particularly, we are intended to identify the fingerprints that are collected near doors. We define $\hat{f}_i$ and $\hat{f}'_i$ as follows:

$$(\hat{f}_i, \hat{f}'_i) = \arg\min_{f \in F_{R_i}, f' \in F_c} \| f - f' \|, \qquad (3)$$

where $\| \cdot \|$ denotes the 2-Norm in the fingerprint space.

Specifically, $\hat{f}_i$ and $\hat{f}'_i$ locate as close as possible to a door in the floor plan but in opposite sides ($\hat{f}_i$ inside the room and $\hat{f}'_i$ outside the room). Let $F_D = \{\hat{f}'_i, i = 1, 2, \ldots, k\}$ denote the set of key corresponding points. Actually, the fingerprints in F_D can be organized in a chain in the MST T, as shown in Figure 10. So we present F_D in a vector form as $F_D = (f_1, f_2, \ldots, f_k)$.

While in the stress-free floor plan, let $P_D = (p_1, p_2, \ldots, p_k)$ denote the set of sample locations in the corridor that are the closest to every door. The order
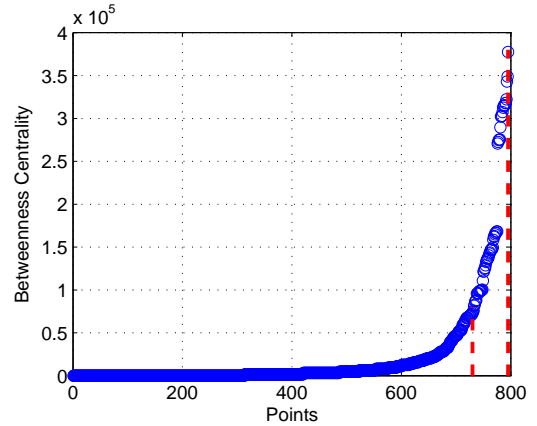
of sample locations in P_D are in accord with their appearance from one side to the other side along the corridor. There are two possible ways ($\sigma_1, \sigma_2 : F_D \to P_D$) mapping F_D to P_D:

$$\sigma_1 : f_i \mapsto p_i;$$

$$\sigma_2 : f_i \mapsto p_{k-i+1}.$$

In fact only one of σ_1 and σ_2 is the ground-truth. We use the distance constraints in both stress-free floor plan and fingerprint space to eliminate the ambiguity. We define $l = (l_1, l_2, \ldots, l_{k-1})$ and $l_i = \| p_{i+1} - p_i \|$. Similarly, l' is defined as $l' = (l'_1, l'_2, \ldots, l'_{k-1})$ and $l'_i = \| \hat{f}_{i+1} - \hat{f}_i \|$. The values of l_i and l'_i can be determined according to the distance matrix D and D', respectively. The cosine similarity of l and l', denoted by s_1, is calculated by

$$\frac{l \cdot l'}{\| l \| \| l' \|}.$$

While the similarity of l and the reverse of l', denoted by s_2, is also calculated. If $s_1 \geq s_2$, we adopt σ_1, otherwise σ_2. Without loss of generality, σ_1 is chosen in the following discussion.

Up to now, a group of fingerprints (F_D) are labeled with real locations. The relationship between F_D and P_D can be further used to map other fingerprints to real locations.

6.2 Space Transformation

In this section, we discuss how to map fingerprints (fingerprint space) to locations (stress-free floor plan). We initially try floor-level transformation and then turn to room-level transformation for better accuracy.

6.2.1 Floor-level Transformation

From the visualization of the stress-free floor plan and the fingerprint space, we observe that they are structurally similar but under trivial variations, including translation, rotation, or reflection. We use a transform matrix to solve such trivial variations.

Suppose a fingerprint $f_i \in F_D$ has its coordinate in the form of $x_i = [x_i^1\ x_i^2\ \ldots\ x_i^d]^{\mathrm{T}}$, where d is the dimension of the fingerprint space. And its corresponding location $p_i \in P_D$ has a coordinate $y_i = [y_i^1\ y_i^2\ \ldots\ y_i^d]^{\mathrm{T}}$
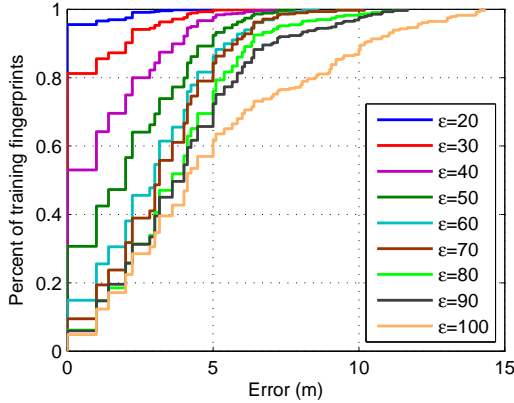
Fig. 12: CDF of location error on different ϵ.

Fig. 13: Room error rate vs. ϵ.

in the stress-free floor plan. Let A denote the $d \times d$ transformation matrix and $B = [b_1 \ b_2 \ \ldots \ b_d]^{\mathrm{T}}$. We have $k = |F_D|$ following equations

$$y_i = Ax_i + B. \tag{4}$$

We re-write the k equations as

$$H_i z = G_i, \tag{5}$$

where $H_i = [x_i^{\mathrm{T}} \ 1]$, $z = [A \ B]^{\mathrm{T}}$, and $G_i = y_i^{\mathrm{T}}$. Combining k equations as a matrix equation, we have

$$Hz = G, \tag{6}$$

where H_i and G_i are the ith row of H and G, respectively.

The least square estimation [3] of above k equations gives

$$\bar{z} = (H^{\mathrm{T}}H)^{-1}H^{\mathrm{T}}G, \tag{7}$$

which minimizes $\| G - Hz \|$.

So far, the transformation matrices A and B can be determined by $\bar{z}$; thus we are able to map any fingerprint to the stress-free floor plan with a fixed location. For a fingerprint f with the coordinate $x = [x^1 \ x^2 \ \ldots \ x^d]^{\mathrm{T}}$, the sample location that is closest to $Ax + B$ is estimated as the real location of f.

6.2.2 Room-level Transformation

From the experiment results, the unsatisfactory performance of floor-level transformation motivated us to design a fine-grained mapping solution. As previously mentioned, doors and fingerprints near doors are related, which further indicates that the rooms and the fingerprints from corresponding rooms are also related since a door belongs to only one room (i.e., the mapping from doors to rooms is injective). This fact enables room-level mapping instead of floor-level mapping.

Using MDS, the fingerprints from one room are transformed to d-dimension space. In the same way, the sample locations from the corresponding room are also mapped to d-dimension stress-free floor plan. Using doors and room corners as reference points, the fingerprints and sample locations are linked determinately by the transformation matrix above discussed. We perform the above step one room by one room
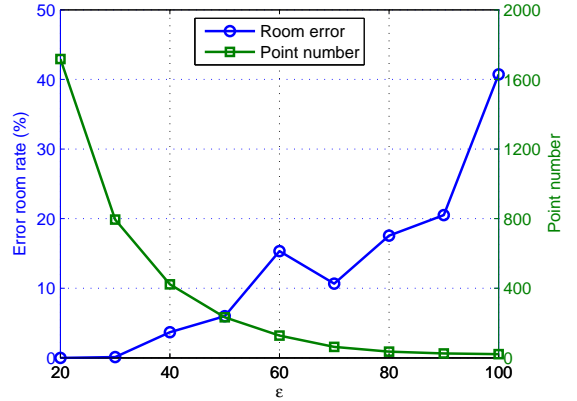
and finally achieve a full mapping for all fingerprints after multiple steps of room-level transformation.

7 LOCALIZATION

Basically, LiFS performs fingerprint database lookup to response to current location query using the nearest neighbour algorithm. User's current location is estimated as the location of which the fingerprints pre-stored in the database are the most similar to the query measurements in terms of Euclidean distance. Under this scheme, LiFS yields average localization error of 5.9 meters and maximum error of about 16 meters, as shown in Fig. 20.

The causes of insufficient accuracy lie in two folds: 1) the fingerprint databased generated by automatic crowdsourcing is naturally less accurate than that constructed by manual efforts. 2) the fingerprints from distant locations could be similar and thus difficult to distinguish due to complex indoor environments, which is referred to as fingerprint ambiguity in the literature [40]. While the former is inevitable to crowdsourced site survey, we propose a continuous trajectory matching scheme to reduce the localization error caused by the fingerprint ambiguity for mobile users. The idea is similar to other sequence matching approaches for GSM localization [25], [45].

We record successive RSS measurements instead of only single report when a user is moving. Such continuous fingerprints represent the moving trajectory of a mobile user. When performing the localization, we search for a series of locations $L = \{L_1, L_2, \cdots, L_k\}$ that minimize the integral distance as follows:

$$arg \min_{f'_{L_i} \in F} \sum_{i=1}^{k} ||f_i - f'_{L_i}||, \tag{8}$$

where f_i is the ith RSS vector in the trajectory measurements, and f'_{L_i} indicates the fingerprint of location L_i stored in fingerprint database F. When the matched trajectory is determined, the latest location, L_k, is returned as the current estimation.

Finding the best-fitted trajectory, however, incurs exponential computation cost. To reduce the search space, we leverage the dead-reckoned distance between successive measurements and consider only the
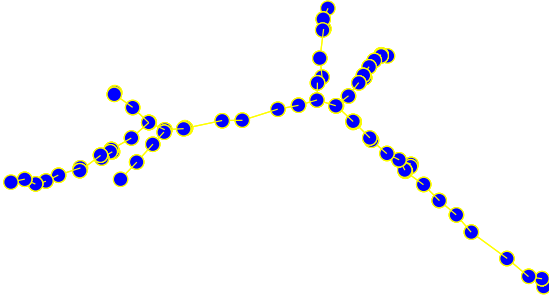
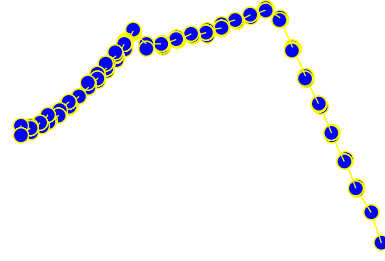Fig. 14: MST of corridor points extracted using betweenness centrality.

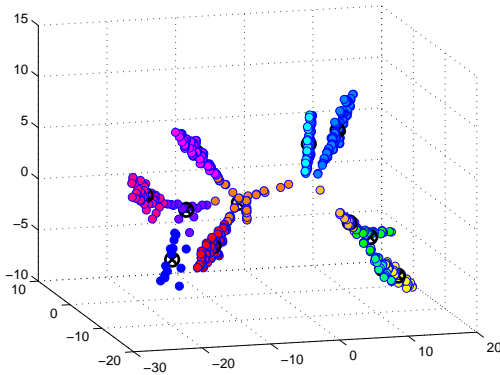Fig. 15: MST of corridor points.

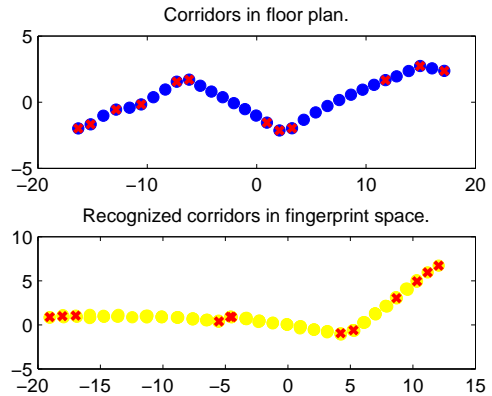Fig. 16: Clustering results by K-Means.

Fig. 17: Floor plan corridor vs. Recognized corridor. Points marked with 'X' are reference points.

candidate locations satisfying the distance constraints. By doing this, the number of candidate locations are largely reduced. Assume that the the number of candidate locations of the ith measurement in the trajectory is reduced from n to a limited value, denoted as c_i, then the computational complexity of the trajectory matching decreases from $O(n^k)$ to $O(n \cdot c^{k-1})$, where k is the record number of the trajectory and $c = \max\{c_i, i = 2, \cdots, k\}$. Typically, c is a small number (usually no more than 10) given the distance constraints. In addition, a small value k (e.g., 2 to 4) is demonstrated to generate considerable accuracy improvement. Hence in practice, the complexity can be approximately treated as $O(n)$ since the factor c^{k-1} is equivalent to a constant. In other words, as demonstrated in the experiments, the trajectory matching scheme would not cause heavy computational cost, yet can significantly improves accuracy. Since we have conducted inertial sensing to count footsteps as distance estimation in fingerprint space (as expounded in Section 5, the trajectory matching scheme that rely on successive RSS measurements and continuous dead-reckoning does not incur too much extra cost. The performance benefits are provided in Section 8.

8 EXPERIMENTS

8.1 Experiment Design

We develop the prototype of LiFS on the increasingly popular Android OS and on two Google Nexus S

phones which support WiFi and accelerometer sensors. We conduct the experiment on one floor of a typical office building covering 1600m^2, with the length of 70m and width of 23m. As shown in Figure 2, the building contains 16 office rooms, of which 5 are large rooms of 142m^2, 7 are small ones with different sizes and the other 4 are inaccessible. Totally $m = 26$ APs are installed, of which 15 are with known locations and are denoted in Figure 2.

We sample the experiment floor plan approximately every 4m^2 (2m×2m grid) and obtain 292 sample locations over all accessible areas. Although this density is not a consistently best solution for all situations (including corridors, office rooms, meeting rooms, etc.,) however, it provides reasonable positioning accuracy for general office buildings. Afterwards, we conduct MDS and the results in 2D and 3D are depicted in Figure 5 and 6, respectively.

The experiment lasts five hours by 4 volunteers. Each volunteer holds a mobile phone in hand and walk through areas of interests. LiFS records the accelerometer readings to count walking distances and picks up RSS values along the paths. Fingerprints are recorded every 4∼5 steps during moving, which corresponds 2∼3m under normal walking styles. Accelerometers work in two different frequencies: when detecting movements, they record sensory data with short intervals; otherwise a relatively long interval is adopted. WiFi is only scanned when the users are detected to be moving at a frequency of about 2Hz.

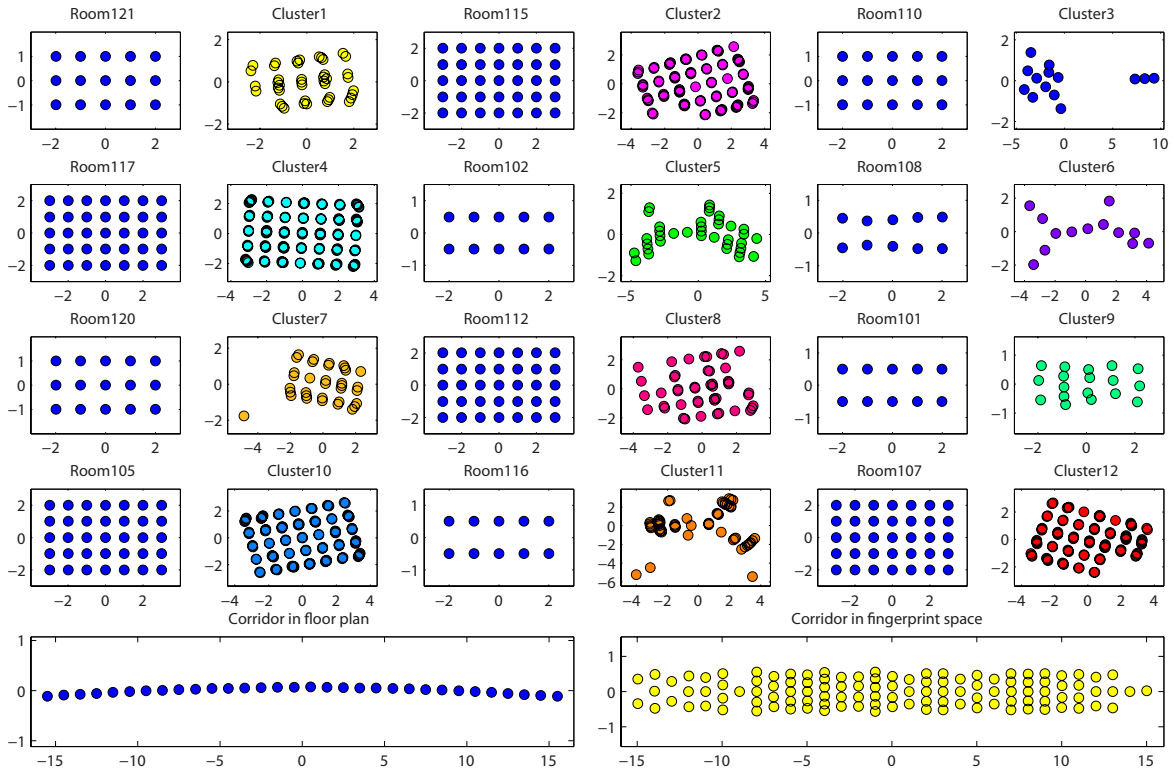Totally 600 user traces along with 16,498 fingerprint

Fig. 18: Fingerprints clusters vs. floor plan rooms.

records are collected. These traces cover most of the areas of the experimental field. The small and large rooms are covered by at least 5 and 10 paths, respectively. In addition, the corridor is covered by more than 500 paths. Different paths vary not only in the areas they covered but also in lengths. The raw data are preprocessed and refined according to the RSS values and stability over time. After that, we select a half of these data for training and use the rest in operating phase.

8.2 Performance Evaluation

8.2.1 Fingerprint Space Generation

Before generating the fingerprint space, we obtain fingerprint points (i.e., points in the fingerprint space) and their pairwise distances from raw sensory data. Each point has a set of fingerprints. Fingerprints are distinguished by their RSS dissimilarities. Fingerprints with similar RSS features are attached to the same fingerprint points while fingerprints with large dissimilarities are sticked to different points. As user traces may be overlapped in the floor plan, fingerprints collected from different traces may be attached to the same point in the fingerprint space. In addition, fingerprints from a same sample location may be bounded to different points due to the RSS fluctuation. Hence, the threshold value of ϵ can affect the fingerprint space generation a lot.

To obtain an appropriate ϵ for generating fingerprint space, we first collect a series of fingerprints from the same location and calculate the maximum dissimilarity of these fingerprints. This procedure is repeated over a set of distinct locations (randomly selected), which results in a set of dissimilarity values. Finally we set the threshold epsilon as the mean value of these dissimilarity values, which is supposed to represent the average maximum dissimilarity of fingerprints from one distinct location.

Location error and *room error* defined as follows are used to examine the effects of ϵ.

$$Location\_Error = ||L(f) - L'(f)||, \qquad (9)$$

$$Room\_Error = \frac{1}{N} \sum_{f \in F} I(R(f) \neq R'(f)), \qquad (10)$$

where f is a fingerprint, $L(f)$ $(R(f))$ and $L'(f)$ $(R'(f))$ represent the ground truth location (room) in floor plan and in fingerprint space respectively, N is the number of fingerprints, F is the set of fingerprints, and I is an indicative function. For each fingerprint, its ground truth location (room) in fingerprint space is determined as the labeled-location of those predominant fingerprints with the same location (room) label.

We plot the cumulative distribution (CDF) of location error in Figure 12. The impact of ϵ on room error and the number of points are illustrated in Figure 13. As from the results, location error and room error both increase when ϵ changes from 10 to 100, while the number of points decreases from about 1,600 to 1. Too small or large values of ϵ deteriorate the performance as fingerprints will be wrongly clustered. We choose $\epsilon = 30$ for further experiments, since 80% of fingerprints are accurate when $\epsilon = 30$.

To obtain walking distances of fingerprints, we first evaluate the step counts estimation using the
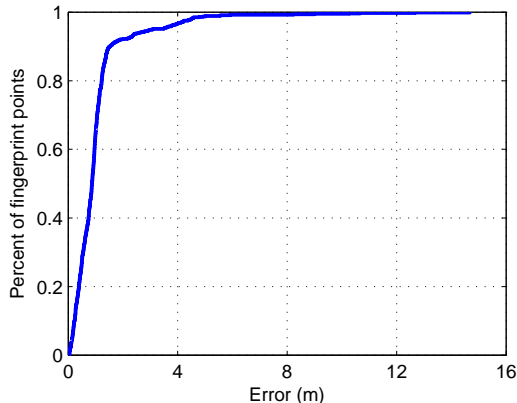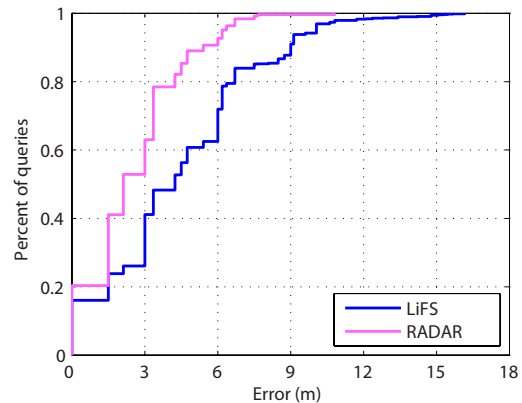
Fig. 19: CDF of mapping error.

Fig. 20: CDF of localization error.

local variance threshold method. Paths with different lengths (from 5 to 200 footsteps) are designed for testing. Experiment results show an error rate of 2% in the number of steps. To further validate the performance under relatively long traces in different scenarios, we collect accelerometer data from different users by letting them walk a 300-step walking with their mobile phones in hand, in pockets, or in bags and perform the step counting algorithm. The results show that errors of 90% of all testing cases are limited within 5 steps, which is fairly accurate and suffices the requirements of distance estimation for LiFS. Although different users have various step sizes which result in different distances of the same number of steps, it will be shown later that MDS has outstanding performance in tolerance to measurement errors. The accumulative error of long paths brings about unobvious performance drop as only path segments (inter-fingerprint distances) are used by MDS and the distance of far-away points are calculated by aggregating many paths.

Totally, 795 points are generated for fingerprint space when $\epsilon = 30$. First we assign the pairwise distances of these points with their measured walking distances and thus we get a connected network. By performing the Floyd-Warshall algorithm on the network, we obtain all the pairwise distances of 795 points. Finally, we conduct 2D and 3D MDS on these points and the results are shown in Figure 8 and 9, where each color denotes one room (or the corridor) in the floor plan. As seen from the figures, real floor plan structure is well reflected by MDS under constraints of walking distances.

8.2.2 Mapping Performance

We build the MST of the fingerprint points (Figure 10) to calculate the betweenness centrality of each point. We sort all points by betweenness centrality in Figure 11 and select those points with higher betweenness than the watershed value (8,000 in our experiments). All selected points are estimated from the corridor recognition. As illustrated in Figure 14, most of the candidate corridor points are correctly extracted. However, some room points are also mixed among them and on the other hand, some true corridor fingerprints are not included. Hence, we refine the

corridor recognition by iteratively performing MST and sifting low betweenness points until the MST of the remaining points form a single line, i.e., each point has at most one parent and one child in the MST. The final corridor points are depicted in Figure 15.

The rest of fingerprint points, most of which are actually collected from rooms, are then clustered into 12 clusters (equal to the room number) using k-Means. The clustering results are shown in Figure 16, where each different color indicates a cluster. The figure shows that most rooms can be recognized correctly while only a small portion of corridor points are mixed.

For each cluster, we identify a point in the corridors that has the shortest distance to all the points in a cluster as the reference point. The 12 reference points for 12 clusters are shown in Figure 17. Some clusters may take the same point as its reference point, which is caused by the clustering errors. The reference points in the floor plan which link rooms to corridors are also presented in Figure 17. The reference point sequences of floor plan and fingerprint space are $l = \{2.37, 3.36, 9.40, 1.18, 1.16, 8.07, 1.17, 3.82, 2.51, 2.55, 1.25\}$ and $l' = \{0.33, 2.12, 12.98, 1, 31, 1, 31, 10.17, 1.24, 10.17, 1.24, 5.99, 3.69, 1.18, 0\}$ respectively. Let l'' be the reverse of l'. The cosine similarities of l and l' and l and l'' are $s_1 = 0.97$ and $s_2 = 0.67$, respectively. Since $s_1 > s_2$, l' is adopted.

Up to now, the corresponding relationship of clusters to rooms is achieved. We then conduct the room-level transformation below. To understand the cluster-room mapping, we plot the 2D MDS results of each cluster and each room in Figure 18. The mapping relations of 12 clusters and their corresponding rooms are also illustrated in Figure 18. As seen from Figure 18, the stress-free rooms are the same as in the floor plan while the 2D fingerprint points especially those from small rooms are a bit rambling. This is because the points are from multiple rooms and the measured distances are of errors.

We then map the points in each cluster to sample locations in its corresponding room by choosing the nearest neighbor for each point. As shown in Figure 19, the mapping results are satisfactory as the mapping error of up to 96% points is lower than 4 meters

and the average error of is only 1.33 meters.

8.2.3 Localization Error

Two metrics are designed for localization performance: location error and room error. Location error is defined as the Euclidean distance from the estimated location to the ground truth one. Room error means the error rate of fingerprints that are estimated to be in incorrect rooms. As the final outputs of LiFS, the RSS noises and mapping errors are simultaneously taken into account. We emulate 8,249 queries using real data on LiFS, and integrate all the localization results, as shown in Figure 20. Each query contains a fingerprint and LiFS returns an estimated location.

We also implement RADAR [2], a famous and classical fingerprint-based localization system, and compare its performance with LiFS on the same experiment data. The reasons we choose RADAR as our performance reference are that LiFS adopts the standard algorithm of RADAR for fingerprint matching and our main purpose of this part of experiments is to show the location accuracy losses due to crowdsourced site survey, rather than how LiFS outperforms RADAR. As shown in Figure 20, the average localization error of LiFS is 5.88 meters, which is larger than RADAR (3.42 meters). The performance of LiFS is comparable to the state-of-the-art model-based approaches (larger than 5 meters) reported in [35] and outperforms EZ (larger than 7 meters) [6]. As shown in Figure 20, localization error of 80% of fingerprints is under 9 meters while about 60% is under 6 meters. Some location errors are caused by the symmetric structure of rooms, but they are relatively small and will not contribute to room error. This accuracy is fairly reasonable, though not much impressive, as LiFS needs no site survey and no specific infrastructure.

While the basic performance of LiFS is as expected to be no better than the classical RADAR since the automatic generated radio map is inevitably less accurate than the manually constructed one, the accuracy of the enhanced trajectory matching scheme for mobile users is surprisingly promising. We examine the effect of the trajectory matching scheme by letting a user walk for a certain duration and then stop at a specific spot. The RSS and inertial sensor data along the user's moving are record and fed to the localization server while the spot which the user stopped at is marked as the ground-truth location of this query. We collect multiple query trajectories from different users and integrate the localization results in Fig. 21. As portrayed in Fig. 21, the average localization error is about 2 meters and the maximum error is bound within 8 meters, reduced by more than 60% and 50% respectively compared to the basic localization scheme. The results not only outperform the RADAR but also are comparable to several recent localization schemes [18], [29]. While providing good accuracy, the computing time is no significantly larger than the nearest neighbor algorithm, since the accuracy improvement can be gained by short trajectories of several successive records. We also analyse the room
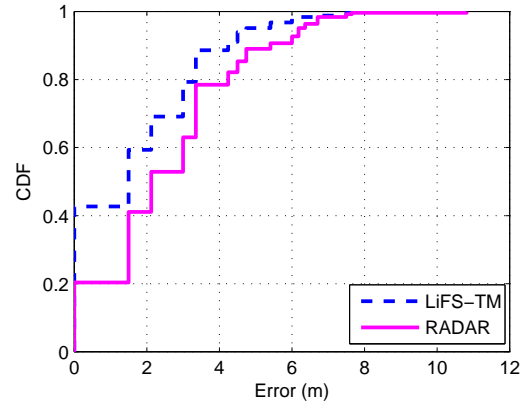

Fig. 21: Localization error of trajectory matching scheme (Denoted as LiFS-TM)

error of all queries on LiFS and find that the room error rate is only 10.91%. As shown in Table 1, LiFS requires less sensors (only accelerometer) while produces comparable accuracy compare to a number of other localization methods. Such encouraging results show the benefits of mobility assisted localization and demonstrate the feasibility of LiFS in practical applications.

9 DISCUSSION AND FUTURE WORK

9.1 Global Reference Point and Multiple Floors

Global reference points include the last reported GPS location [6], AP's location [20], similar surrounding sound signature [1], feature-distinct public area, etc. Though we do not use global reference point in this design, they can be integrated into LiFS, resulting in a more robust mapping solution, while LiFS still works in case of deficient global information. Global reference points are also the key in case of symmetric floor plans or multi-floor buildings. As the crowdsourced data can be collected from multiple floors in practice, part of our future work is to incorporate automatic floor identification by leveraging global reference point as long with existing floor localization methods such as skyloc [36]. Inspired by recent works, the motion states of users walking across different floors using elevators, escalators, or stairs can be characterized by inertial sensors with advanced detection algorithms [15], [33], [37]. These motion states can further be incorporated into the LiFS systems to determine floors.

9.2 Building Types

Our experiment field is one floor of an academic building. The corridor in the middle connects all other office rooms that lie on both sides of the corridor. According to such layout, we try to distinguish corridors and rooms based on user traces. This solution fits a majority of office buildings but may fail in large open environments, such as hall, atrium, gymnasium, or museum, in which users' movements are difficult to characterize. We envision that the recognition of

TABLE 1. Comparison with other localization systems.

	Methodology	Sensors	Floor plan	Device diversity	Experimental Areas	Reported Median accuracy	Reported 80%ile accuracy
EZ [6]	Ranging	GPS	No	Yes	Whole building (Office building)	2m (small area) 7m (large area	3.3m (small area) 10m (large area)
Zee [29]	Fingerprinting	A/G/C*	Yes	No	Only hallways (Office building)	7m (using Horus)	13m (using Horus)
Unloc [37]	Fingerprinting	GPS/A/M/G/C	No	No	Whole building (Office& Mall)	1.7m	3.5m
LiFS	Fingerprinting	A	Yes	No	Whole building (Office building)	4.5m (LiFS) 1.5m (LiFS-TM)	7m (LiFS) 3.5m(LiFS-TM)

* A - Accelerometer, G - Gyroscope, C - Compass, M - Magnetometer

different functions of areas helps to regionalize spaces and model users' movements, as our future work.

9.3 Device Heterogeneity

Device heterogeneity is a long-standing common challenge of all WiFi fingerprint based indoor localization techniques. Even under identical wireless environment, users with different devices would observe different wireless signals (and thus wireless fingerprints) due to the hardware diversity. The problem become even more prominent in crowdsourcing-based applications where a large amount of users with different devices participate in data collection. In our work, benefiting from user mobility information, LiFS is supposed to tolerate different hardware devices better than traditional methods because mobility provides a new dimension of knowledge as an effective supplement to wireless fingerprint itself. However, the issue is not yet dedicatedly considered in current prototype implementation, which leaves as our future work.

9.4 Floor Plan Construction From User Data

Floor plan plays an essential role in many indoor pervasive and mobile applications, but its collection and on-site calibration are inconvenient and usually prohibitively costly for (indoor) map providers. Based on the ideas of human-centric sensing and crowd-sourcing, it becomes possible to generate floor plans automatically. The movement records from a large amount of contributing users can be used to depict the interior layout of a building. Further, the functions of a specified area (such as offices, corridor, elevator, and stairs) can be identified from rich sensor hints according to user behaviors. Such higher level semantics enable niche targeting location-based services.

9.5 Social Significance of Mobility Observation

Automatic floor plan construction also assists the research of human mobility and social behavior, in which data collection is hard and the collected data are less well labeled. Our system provides not only sufficient mobility information, but also corresponding meaningful comments. For example, through the traces collected from one user in a certain duration, we can investigate the lengths of stays at his office and in the corridor, the total walking distance, as well as the average speeds in the office and corridor respectively, and etc.

10 CONCLUSION

By utilizing the spatial relation of RSS fingerprints, we are able to create fingerprint space in which fingerprints are distributed according to their mutual distances in real world. On this basis, we design and implement LiFS, an indoor localization system based on off-the-shelf WiFi infrastructure and mobile phones. The preliminary experiment results show that LiFS achieves low human cost, rapid system deployment, and competitive location accuracy. This work sets up a novel perspective to cut off human intervention of indoor localization approaches. Our ongoing research focuses on making LiFS feasible and pervasive to various applied environments and buildings.

11 ACKNOWLEDGMENTS

REFERENCES

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of ACM MobiCom*, pages 261–272, 2009.
[2] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM*, volume 2, pages 775–784, 2000.
[3] Å. Björck. *Numerical methods for least squares problems*. Number 51. Society for Industrial Mathematics, 1996.
[4] I. Borg and P. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Verlag, 2005.
[5] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *Proceedings of the ACM MobiSys 2012*, pages 169–182, 2012.
[6] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of ACM MobiCom*, pages 173–184, 2010.
[7] T. Cormen. *Introduction to algorithms*. The MIT press, 2001.
[8] J. Costa, N. Patwari, and A. Hero III. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks*, 2(1):39–64, 2006.
[9] B. Ferris, D. Fox, and N. Lawrence. Wifi-slam using gaussian process latent variable models. In *Proceedings of IJCAI*, pages 2480–2485, 2007.
[10] R. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.
[11] W. G. Griswold, P. Shanahan, S. W. Brown, R. Boyer, M. Ratto, R. B. Shapiro, and T. M. Truong. ActiveCampus: experiments in community-oriented ubiquitous computing. *Computer*, 37(10):73–81, 2004.
[12] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. Efficient, generalized indoor wifi graphslam. In *IEEE ICRA 2011*, pages 1038–1043, 2011.

[13] A. Jiménez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In *Intelligent Signal Processing, IEEE International Symposium on*, pages 37–42, 2009.

[14] J. Koo and H. Cha. Autonomous construction of a WiFi access point map using multidimensional scaling. *Pervasive Computing*, pages 115–132, 2011.

[15] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74–82, 2011.

[16] 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.

[17] H. Lim, L. C. Kung, J. C. Hou, and H. Luo. Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure. *Wireless Networks*, 16(2):405–420, 2010.

[18] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Accurate wifi based localization for smartphones using peer assistance. *IEEE Transactions on Mobile Computing*, 2013.

[19] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14, 1967.

[20] D. Madigan, E. Einahrawy, R. P. Martin, W. H. Ju, P. Krishnan, and A. S. Krishnakumar. Bayesian indoor positioning systems. In *Proceedings of IEEE INFOCOM*, pages 1217–1227, 2005.

[21] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-slam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of AAAI*, pages 593–598, 2002.

[22] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. LANDMARC: indoor location sensing using active RFID. *Wireless Networks*, 10(6):701–710, 2004.

[23] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *Proceedings of IEEE INFOCOM*, volume 3, pages 1734–1743, 2003.

[24] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, 2010.

[25] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan. Energy-efficient positioning for smartphones using cell-id sequence matching. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 293–306, New York, NY, USA, 2011. ACM.

[26] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, et al. Growing an organic indoor location system. In *Proceedings of ACM MobiSys*, pages 271–284, 2010.

[27] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of ACM Mobi-Com*, pages 32–43, 2000.

[28] T. Pulkkinen, T. Roos, and P. Myllymäki. Semi-supervised learning for wlan positioning. *Artificial Neural Networks and Machine Learning*, pages 355–362, 2011.

[29] A. Rai, R. Sen, K. K. Chintalapudi, and V. Padmanabhan. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of ACM MobiCom*, 2012.

[30] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: spot localization using phy layer information. In *Proceedings of the ACM MobiSys*, pages 183–196, 2012.

[31] Y. Shang and W. Ruml. Improved MDS-based localization. In *Proceedings of IEEE INFOCOM*, volume 4, pages 2640–2651, 2004.

[32] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from mere connectivity. In *Proceedings of ACM MobiHoc*, pages 201–212, 2003.

[33] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: indoor pathway mapping made easy. In *Proceedings of the USENIX NSDI*, pages 85–98, Berkeley, CA, USA, 2013. USENIX Association.

[34] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[35] D. Turner, S. Savage, and A. Snoeren. On the empirical performance of self-calibrating wifi location systems. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 76 –84, oct. 2011.

[36] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara. The skyloc floor localization system. In *Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on*, pages 125–134. IEEE, 2007.

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

[38] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of ACM UbiComp*, pages 114–123, 2008.

[39] C. Wu, Z. Yang, Y. Liu, and W. Xi. Will: Wireless indoor localization without site survey. *Parallel and Distributed Systems, IEEE Transactions on*, 24(4):839–848, 2013.

[40] S. Yoon, K. Lee, and I. Rhee. FM-based indoor localization via automatic fingerprint DB construction and matching. In *Proceeding of the ACM MobiSys*, pages 207–220, New York, NY, USA, 2013. ACM.

[41] M. Youssef and A. Agrawala. The horus WLAN location determination system. In *Proceedings of ACM MobiSys*, pages 205–218, 2005.

[42] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. A-grawala. Pinpoint: An asynchronous time-based location determination system. In *Proceedings of ACM MobiSys*, pages 165–176, 2006.

[43] X. Zhang, Z. Yang, C. Wu, W. Sun, and Y. Liu. Robust trajectory estimation for crowdsourcing-based mobile applications. *Parallel and Distributed Systems, IEEE Transactions on*, pages 1–1, 2013.

[44] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng. I am the antenna: accurate outdoor ap location using smartphones. In *Proceedings of the ACM MobiCom*, MobiCom '11, pages 109–120, New York, NY, USA, 2011. ACM.

[45] P. Zhou, Y. Zheng, and M. Li. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. In *Proceedings of the ACM MobiSys*, pages 379–392, 2012.

Chenshu Wu received his B.S. degree in School of Software from Tsinghua University, Beijing, P.R. China, in 2010. He is now a Ph.D. student in the Department of Computer Science and Technology, Tsinghua University. His research interests include wireless ad-hoc/sensor networks and mobile computing. He is a student member of the IEEE and the ACM.

Zheng Yang received a B.E. degree in computer science from Tsinghua University in 2006 and a Ph.D. degree in computer science from Hong Kong University of Science and Technology in 2010. He is currently an assistant professor in Tsinghua University. His main research interests include wireless ad-hoc/sensor networks and mobile computing. He is a member of the IEEE and the ACM. He is awarded the 2011 National Nature Science Award (second class).

Yunhao Liu received the BS degree in automation from Tsinghua University, China, in 1995, the MS and PhD degrees in computer science and engineering from Michigan State University, in 2003 and 2004, respectively. He is currently a ChangJiang professor at Tsinghua University. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing.