PiLoc: a Self-Calibrating Participatory Indoor Localization System

Chengwen Luo
School of Computing
National University of Singapore
Singapore
chluo@comp.nus.edu.sg

Hande Hong
School of Computing
National University of Singapore
Singapore
honghand@comp.nus.edu.sg

Mun Choon Chan
School of Computing
National University of Singapore
Singapore
chanmc@comp.nus.edu.sg

***Abstract*—While location is one of the most important context information in mobile and ubiquitous computing, large-scale deployment of indoor localization system remains elusive.**

In this work, we propose PiLoc, an indoor localization system that utilizes opportunistically sensed data contributed by users. Our system does not require manual calibration, prior knowledge and infrastructure support. The key novelty of PiLoc is that it merges walking segments annotated with displacement and signal strength information from users to derive a map of walking paths annotated with radio signal strengths.

We evaluate PiLoc over 4 different indoor areas. Evaluation shows that our system can achieve an average localization error of 1.5m.

***Index Terms*—Indoor Localization, Participatory, Smartphones, Floor Plan**

I. Introduction

Location is one of the most important context information in mobile and ubiquitous computing. Recently, wireless indoor localization has attracted extensive research efforts [1]–[11] due to both the need to support indoor location-based services, and the fact that GPS does not work well indoors. However, despite significant progress, an indoor localization system that can be easily deployed on a large scale remains a challenge.

One important bottleneck that hinders large-scale deployment of existing indoor localization systems is labor-intensive site survey and system maintenance. Many of these systems involve a dedicated offline calibration stage that builds a radio map to aid localization. This calibration stage involves manual association of a location to be localized with its corresponding radio fingerprints. Furthermore, this radio map needs to be periodically updated to reflect changes in the environment. The calibration and maintenance effort required make these systems tedious and difficult to deploy on a large scale.

Recently, researchers have attempted to address this problem. For example, LiFS [1] records the inter-distances between fingerprints and uses multidimensional scaling to assign fingerprints to physical locations with minimum human intervention. Zee [3] exploits the inertial sensing capabilities of smartphones and infers location according to the constraints imposed by the walking trajectory of the user and the floor plan. There are also approaches that try to minimize the fingerprint collection effort by utilizing the signal propagation models [12], [13]. However, all these approaches [1], [3], [12], [13] require knowledge of the floor plan. UnLoc [2] attempts to reduce the need for floorplan by requiring the identification of sufficient landmarks.

All these recent research developments have shown that calibration effort can be reduced with either prior knowledge of accurate floor plans of the places being measured or the availability of sufficient number of reliable landmarks. However, an accurate floor plan or sufficient reliable landmarks may not always be easily available. In this work, we attempt to answer the following question: *can we design an indoor localization system that infers the floor plan and automatically calibrates itself without landmarks?*

We propose *PiLoc*, an indoor localization system that calibrates itself through user generated data. PiLoc is based on the following observations. First, sensor-enhanced smartphones are becoming increasingly pervasive. Second, smartphone can record a user's movement (distance and direction) together with names of APs within range and the associated signal strengths. Finally, it is possible to merge many walking segments annotated with displacement and signal strength information from users to derive a map of walking paths annotated with radio signal strengths. The last observation is central to the design of PiLoc.

By utilizing opportunistic sensing data contributed by users, PiLoc requires no prior knowledge about any building or any user intervention in both the calibration and maintenance stage. It adopts a novel trajectory matching and floor plan construction algorithm to automatically cluster, filter and merge all user inputs to automatically construct floor plans for different indoor areas. Most importantly, radio maps required for localization are also automatically built and updated in this process. PiLoc requires no special purpose hardware, the only assumption in PiLoc is the availability of WiFi infrastructure. While PiLoc itself is a complete indoor localization system, the automatic floor plan construction algorithm of PiLoc also provides insight to solving the floor plan acquisition problem in recent research developments such as LiFS, Zee or other propagation model based approaches.

We have implemented PiLoc and evaluated it in 4 different indoor areas covering a total of $5528m^2$. PiLoc uses one or more mobile clients that track user walking trajectories. Each

978-1-4799-3146-0/14/$31.00 ©2014 IEEE

client uploads the trajectory data to the server once sufficient data has been collected. A server program, takes as input the client trajectory data, generates or updates the floor plan and radio map. The server also handles localization queries from the user and returns the user's current location based on the radio map constructed. Our experiment shows that PiLoc can efficiently generate floor plans and radio maps for different indoor areas and can achieve an average localization error of 1.5m.

The rest of the paper is organized as follows. We present related works in Section II and in Section III the overview of PiLoc. We then describe data collection method in Section IV, trajectory matching algorithm in Section V, floor plan and radio map construction in Section VI. Section VII presents our evaluation results. We finally discuss future extensions in Section VIII and conclude in Section IX.

II. Related Work

A. Device-free Indoor Localization

In device-free localization, the entity being tracked does not carry any special device and does not actively participating in the localization process. Existing device-free localization systems can generally be categorized as MIMO radar-based [4], [14] and Radio Tomographic Imaging (RTI) based [5], [6]. RF-based Device-Free Passive (DfP) localization [15] is based on the idea that the existence of an entity will affect RF signal receptions [16], [17]. While this approach requires no special hardware, the accuracy is lower and the ability to track multiple entities simultaneously is still limited.

B. Device-based Indoor Localization

Device-based indoor localization aims to locate a device or a human subject carrying a device. Generally, these approaches can be categorized into four categories based on the system requirements and the underlying techniques used: infrastructure-based, fingerprint-based, propagation model-based and SLAM-based.

Infrastructure-based. These systems rely on special-purpose infrastructures deployed to locate the target device. Early systems utilize short range infrared [18] or RFID [19] and perform localization based on proximity. Cricket [20] uses radio and acoustic transmission and exploits Time Difference of Arrival (TDoA) in the signals. Recent developments explore multiple-input, multiple-output (MIMO) techniques using commodity APs and Angle of Arrival (AoA) to provide fine-grained localization [7]. While these techniques provide centimeter-level accuracy [7], [8], [20], the need for special-purpose infrastructure and high deployment cost hinder their large-scale deployment.

Fingerprint-based. A significant portion of research works in indoor localization exploit RF signal fingerprint-based approach. The basic idea is to fingerprint each location of interest and locate the device using nearest neighbor matching. The underlying assumption of these approaches is that unique signatures can be found to fingerprint each location. Most of these works use WiFi Receive Signal Strength (RSS) as the fingerprint [21], [22]. Recent works propose other forms of fingerprints such as FM Radio [9] and physical layer information Channel Frequency Response [23]. SurroundSense [10] generalizes the concept of fingerprint and explores ambient information such as noise, light color, etc. Fingerprint-based techniques reduce the deployment cost by leveraging the existing infrastructures and can achieve meter-level accuracy. However, fingerprint-based technique suffers from high calibration cost as a labor-intensive site-survey process is typically required in the offline phase to construct the fingerprint database (radio map) for each known location. The static radio map is also vulnerable to environmental dynamics, resulting in high maintenance effort. PiLoc aims to eliminate these overheads.

Propagation Model-based. In trying to reduce the calibration effort, some researchers propose the signal propagation model-based technique to estimate the RSS value at a given location based on the theoretic model instead of manually tagging [12], [13], [24]. One popular model is log-distance path loss (LDPL) [12], which estimates RSS value based on the propagation distances. RADAR [21] also provides a model-based approach to estimate the RSS value based on the AP locations and floor plans. EZ [12] further improves it and only need to measure the signal strength at a few locations. Compared to the fingerprint-based techniques, model-based techniques typically reduces calibration effort at the cost of reduced accuracy. For most of these systems, AP locations or accurate floor plans need to be given.

SLAM-based. Simultaneous Localization and Mapping (SLAM) techniques have been extensively studied by researchers in the robotic community. SLAM relies on landmark detection by camera, laser or other ranging sensors and accurate controlled movement of robots. Several works have been proposed to leverage the idea of SLAM by combing WiFi and IMU sensors on smartphones. Zee [3] exploits dead-reckoning and infers location according to the constraints imposed by the floor plan. However, it requires an accurate floor plan which is normally not available in practice. UnLoc [2] further exploits dead-reckoning and learn indoor landmarks that exist in the environment to aid localization and requires at least one ground truth location of the landmark. LiFS [1] exploits Multidimensional Scaling (MDS) to match fingerprints with actual location using walking step information. These works successfully reduce effort in generating the radio maps provided that accurate indoor floor plans are given. Kim [25] proposes an autonomous fingerprinting method but requires the strong assumption that the initial location and direction of the user is known a priori. Recently, Walkie-Markie [11] proposes an algorithm to map pathways using WiFi-Marks.

PiLoc is different from other approaches in that it requires no prior knowledge of the environment (e.g., floor plans) or landmarks and combines element of SLAM and fingerprinting

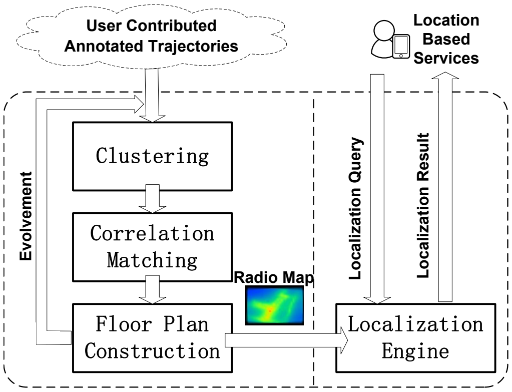


Fig. 1: Overview of PiLoc

to collect, cluster, match, and merge user contributed data to generate and update floor plans and radio maps for localization.

III. PILOC OVERVIEW

The PiLoc architecture is shown in Figure 1. PiLoc exploits crowdsourcing to trace user walking trajectories using Inertial Measurement Unit (IMU) sensors equipped in the smartphones. IMU collects angular velocity and linear acceleration data which are utilized as inputs to the system.

To enable localization, we require that one or more users carrying smartphones with the data collection application enabled walk on various parts of the indoor area to be localized and upload the annotated walking trajectories collected. An annotated walking trajectory consists of discrete walking steps, which further consist of displacement vectors (distance and direction) and the WiFi fingerprints associates with the steps. There is no restriction on the walking patterns and each walking trajectory can cover any part of the area. The limitation is that we can only localize areas that are covered by at least one walking trajectory and localization accuracy improves with more trajectories. These user contributed walking trajectories are used as inputs to construct or update the floor plan of the area covered by user movements.

The key challenge in PiLoc is how to combine these user generated trajectories into a floor plan suitable for localization. There are three main components. First, a clustering algorithm that uses AP signal strength and movement vector to separate these walking trajectories into disjoint sets that cover different indoor environments. The second component takes the disjoint segments generated in the previous component and finds segments that match again based on movement vectors and AP signals. The matching is based on path and radio signal similarity measures between two different trajectories segments within the same cluster. Finally, in the third component, we merge multiple trajectories to build floor plans. In the following sections, we present details of these three components.

IV. DATA COLLECTION

A. *Fingerprint Collection*

Data collection does not have to be performed specifically for localization purposes. Instead, users equipped with smartphones walk around the targeted indoor environment as part of their daily activities. PiLoc opportunistically collects users' walking trajectories $T = \{\tau_i, i = 1, 2, ..., m\}$. Each walking trajectory τ_i is determined by two stationary points detected by the phone accelerometer. $\tau_i = \{s_1, s_2, ..., s_n\}$, in which s_i is a discrete walking step detected by the linear accelerations from the corresponding phone accelerometer input. Besides stride length and heading direction, WiFi RSS fingerprints are collected as well between each two consecutive steps and automatically associate with each step recorded. The heading direction of each step is obtained by converting the linear acceleration from the phone coordinate to the world coordinate. Therefore, each step $s_i = \{ID_i, x_i, y_i, f_i\}$ consists of four elements, global step identifier ID_i, horizontal displacement x_i, vertical displacement y_i and (radio) fingerprints f_i. 2D displacements x_i and y_i are calculated based on the headings (angle relative to the earth North) and stride lengths to identify the relative physical 2D position of the current step with respect to the first step s_1 in the same trajectory. Fingerprints $f_i = \{r_1, r_2, ..., r_k\}$ represents the WiFi RSS measured at step i, where r_j is the received signal strength of AP_j detected.

After collecting sufficient walking trajectories marked with corresponding fingerprints, PiLoc is able to construct floor plans and radio maps for the covered area. The speed of data collection is capped by typical human walking speed. If we consider an indoor area with 100m of walk way and an average walking speed of 4km/h, we can over 1km in 15min or the entire walkway of 100m 10 times.

B. *Inertial Sensing*

Dead-reckoning with smartphones has been exploited in several previous works [1]–[3], [11], [26]. One significant challenge with dead-reckoning is the accumulated error over time. Therefore, it can only be used to track user for a short period of time or error needs to be corrected frequently. This problem makes it very challenging to align and merge different user traces, especially to construct floor plans. This is also one major challenge in PiLoc. Several works have been done to improve the accuracy of dead-reckoning with arbitrary phone placements [3], [27], [28]. Walking steps can be efficiently detected using a threshold based sliding window algorithm [29]. In our experience, step detection is very accurate and most of the time we can detect exact step counts even after several hundred steps. Headings angles can be inferred by combining linear acceleration, compass and gyroscope readings [27]. However, stride length varies for different users. In order to take into account this variation, we adopt the assumption from [3] that stride length follows gaussian distribution and use the default stride length with an additional 15% gaussian noise.

As we will show later, error in dead-reckoning is corrected in PiLoc by combining data from many trajectories in the

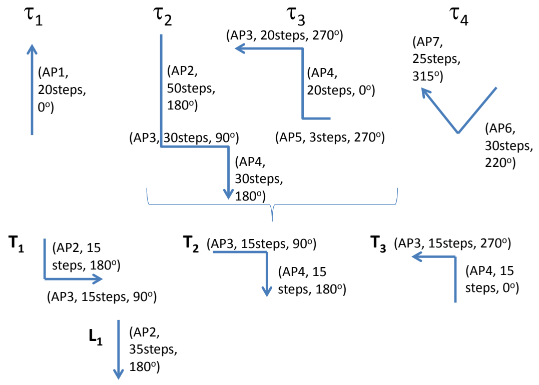


Fig. 2: Examples of Trajectories and Clustering

merging process. In addition, outliers in the data will be filtered out in PiLoc's merging and filtering process if these data do not match well with other data collected.

V. Trajectory Matching

A. Clustering

As data collected from different users cover different parts of different locations, it is necessary to perform an initial level of data clustering to group data into smaller, related groups based on approximate locations and movement vectors.

To illustrate the procedure, consider as input, the four trajectories τ_1, τ_2, τ_3 and τ_4, as shown in Figure 2.

1) AP Clustering: The goal of signal clustering is to divide all trajectories into geographically separated clusters. Each walking trajectory covers a particular indoor environment, and this clustering finds non-overlapping clusters based on the AP information. Given n trajectories inputs from all participating users, the AP clustering finds a clustering with l clusters $C = \{c_1, c_2, ..., c_l\}$, such that:

$$\forall i \forall j \; APSet(c_i) \cap APSet(c_j) = \emptyset, \; 1 \leq i \neq j \leq l \qquad (1)$$

in which $APSet(c_i)$ returns the set of all APs which appear in at least one of the fingerprints in the trajectories of cluster c_i. AP clustering therefore separates trajectories collected in different indoor environments which have different sets of APs into different clusters. As an example, the four trajectories shown in Figure 2 will be separated into three clusters. The APs in each of the three clusters are $\{\tau_1\}$, $\{\tau_2,\ \tau_3\}$ and $\{\tau_4\}$. The corresponding set of APs are $\{$AP1$\}$, $\{$AP2, AP3, AP4, AP5$\}$ and $\{$AP6, AP7$\}$ respectively. As an illustration of the overall effect, as shown in Figure 3, traces collected in 3 buildings are separated into 3 different clusters after AP clustering. Instead of relying on the fluctuating signal strength, AP clustering only detects the existence of APs and provides a more reliable clustering. Though AP clustering only provides floor level granularity, the light-weighted clustering is still an important technique to efficiently categorize the big trajectory data once the system is deployed at scale.

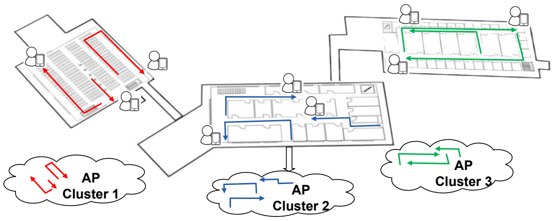


Fig. 3: AP Clustering

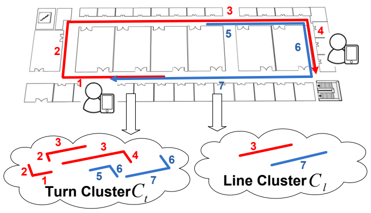


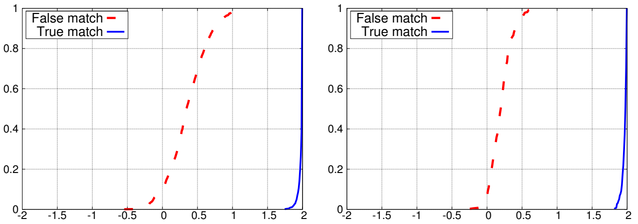
Fig. 4: Path Segment Clustering

2) Path Segment Clustering: Within the same AP cluster, we further divide a single trajectory into disjoint path segments. While path segment can take any form in general, in this work, we consider only two kinds of path segments, namely: *turns* and *long straight lines*. Walking along a straight path or making corner turns are natural walking patterns in an indoor environment. A given trajectory $\tau = \{s_1, s_2, ..., s_n\}$, can be broken into disjoint path segments (consisting of either turns and straight lines) $S = \{s_p, s_{p+1}, ..., s_q\}$ where $1 \leq p < q \leq n$. In dividing the trajectory, we first extract turns with minimum 5 and maximum 15 steps before and after the turning. After that, straight line paths containing at least more than 30 steps are extracted. As an example, consider the cluster consists of τ_2 and τ_3 shown in Figure 2. Only 3 turns, T_1, T_2 and T_3 are extracted. The fourth corner is not considered since the path before the turn is too short (less than 5 steps). Similarly, there is only one straight line segment (where AP2 is recorded). All other straight path segments are too short after the turn segments are removed.

We extract these segments from each trajectory and build second-level clusters $C' = \{c_t, c_l\}$ for each AP cluster in C based on path segments, where c_t is the cluster for turns and c_l is the cluster for long straight line segments. After second level clustering, each cluster c_t and c_l contains segments of same path shape from the same indoor environment. Each segment S in c_t or c_l becomes the basic unit for trajectory matching in the next step. The overall effect is shown in Figure 4.

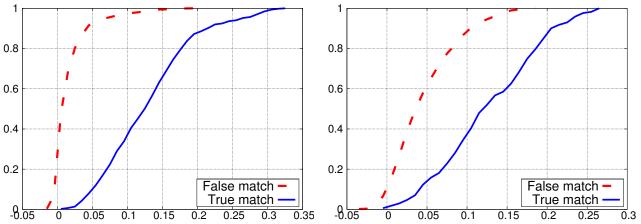
B. Matching

A key difference between PiLoc and prior work is that instead of using WiFi signal or ambient information as landmarks, we utilize movement displacement (distance and direction) as well as the associated signal to match different segments. We found that these parameters can provide high discriminative power for both dead-reckoning error correction and trajectory matching.



(a) $3000m^2$ Office Floor (b) $120m^2$ Research Lab

Fig. 5: CDF of Path Correlation



(a) $3000m^2$ Office Floor (b) $120m^2$ Research Lab

Fig. 6: CDF of Signal Correlation

1) Path Correlation: Similar to the clustering component, the trajectory matching algorithm follows a 2-phase scheme. The first phase is based on a simple but effective idea. When people walk along the same segment (turns or straight lines), the evolution of the two trajectories in a 2D plane should be highly correlated. The path correlation correction can be measured as:

$$Corr_{path} = Corr_x(S_1, S_2) + Corr_y(S_1, S_2) \tag{2}$$

For two path segments from the same cluster c_t or c_l, $S_1 = \{s_1, s_2, ..., s_n\}$ and $S_2 = \{s'_1, s'_2, ..., s'_n\}$ with same number of steps n, the pearson correlation can be computed as:

$$Corr_x(S_1, S_2) = \frac{E[(X_1 - \mu_{X_1})(X_2 - \mu_{X_2})]}{\sigma_{X_1}\sigma_{X_2}} \tag{3}$$

where $X_1 = \{x_1, x_2, ..., x_n\}$ and $X_2 = \{x'_1, x'_2, ..., x'_n\}$ are sequence of horizontal displacements of steps of S_1 and S_2 respectively. Similarly, $Corr_y$ is the correlation of the vertical displacements of steps of S_1 and S_2. These displacements can be computed given the step distance and direction of movement. $Corr_{path}$ therefore measures the similarity between two walking paths in the 2D plane.

Figure 5 shows the CDF of the path correlations for traces collected from both a large indoor floor level covering 3000 m^2 and a research lab covering only 120 m^2. Since one can walk along the same path in two directions, we compute the $Corr_{path}$ in both directions as well and take the maximum of the two as the final path correlation. In both environments more than 90% of path correlations for correct matches (paths with same evolution trend in 2D plane) have value greater than 1.90 (maximum 2). The path correlation is much lower for incorrect matches, with 90% less than 0.75.

2) Signal Correlation: Path correlation alone is not sufficient for obtaining accurate matches. When path segments are collected from parallel corridors in the same building, these segments may have high path correlations. Another feature exploited in PiLoc is changes in RSS signal along the walking path. It is observed that there are specific trends in which RSS signal changes along the same path way. This change is due to the signal propagation and other environmental obstacles. The pattern in which RSS signal changes provide another useful hint to determine matching segments.

One question on using these signal measurements is the stability of their trend with respect to the changes in phone model and time. Figure 7 shows the stability of WiFi signal trends on the same path across 3 different phone models (Samsung Galaxy S3, S4, and Galaxy Nexus). The trends are plotted with smoothed curves and are stable across different phone models for both APs. The variation is also relatively stable at different period of the day. As shown in Figure 8, the RSS trends collected for the same walk path from morning (9am), afternoon (1pm) and in the night (10pm) are also similar. Another observation is that the similarity for APs with higher RSS value tends to be higher than those with lower RSS values. As shown in Figure 7 and Figure 8, the trend detected for AP1 is more stable than that of AP2. With these observations, we use signal correlation as a metric to further measure the similarity between two path segments S_1 and S_2:

$$Corr_{signal} = \sum_i \omega_i \cdot Corr(R_1^i, R_2^i) \cdot I(R_1^i, R_2^i) \tag{4}$$

where $R_1^i = \{r_1, r_2, ..., r_n\}$ and $R_2^i = \{r'_1, r'_2, ..., r'_n\}$ are sequence of RSS values of AP_i observed in S_1 and S_2 respectively. ω_i is the weight for AP_i and we set $\omega_i = \frac{2}{|\mu_{R_1^i} + \mu_{R_2^i}|}$. As signal strength values are given in negative values (measured in dBm), APs with larger average RSS value will have more weight. $Corr(R_1^i, R_2^i)$ is the pearson correlation of two RSS sequences for AP_i. $I(R_1^i, R_2^i)$ is an indicator function used to decide if an AP_i should be included in the computation.

$$I(R_1^i, R_2^i) = \begin{cases} 1, & |\mu_{R_1^i} - \mu_{R_2^i}| < \sigma_{RSS} \quad \text{(5a)} \\ 0, & otherwise \quad \text{(5b)} \end{cases}$$

where σ_{RSS} is the maximum acceptable difference between the two mean RSS values of two path segments. The current value for σ_{RSS} is set to 5dBm, which has been observe to work well for different environments. Similar to the path correlation computation, as movement can occur in both directions on the same path, we calculate the correlation for both forward and

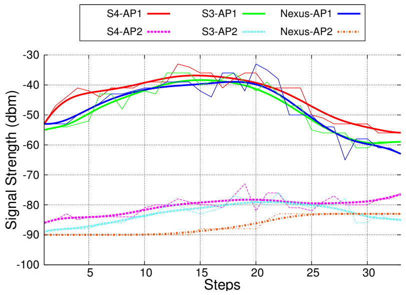


Fig. 7: Stability of Signal Trends (Phone Varing)

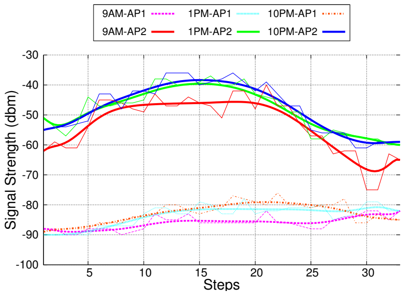


Fig. 8: Stability of Signal Trends (Time Varing)

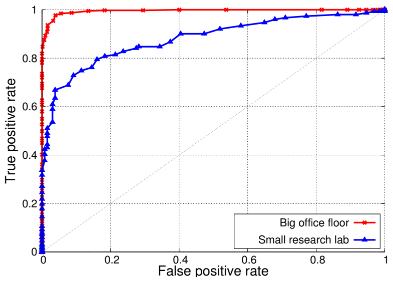


Fig. 9: ROC Curve of Final Matching

reverse directions for each pair of segments and the maximum correlation is used.

Note that not all APs are included in the computation. First, we exclude APs that appear only in one segment and not in the other. Second, we also remove APs that appear in less than 10 steps in either of the two segments. In summary, for the signal correlation computation, we only consider APs that appear often enough in both segments and where their average signal strengths are similar.

In general, the $Corr_{signal}$ increases as two trajectory segments have more common APs and the trends of APs are similar. Figure 6 shows the signal correlation distribution in both the 3000 m^2 office floor and 120 m^2 research lab. In both environments more than 42% of signal correlations for correct matches (same paths) have value greater 0.15. The signal correlation is much lower for incorrect matches, with 98% less than 0.15.

3) Final Matching: PiLoc combines the discriminative power of both path and signal correlations in the final matching to achieve an accurate match. For each pair of segments in the cluster c_t or c_l, we first align them to have same step numbers, and the turning point is used to align turns segments. In this way, PiLoc does not require that the starting points and ending points of the path segments in the matching process to be the same.We use path correlation threshold σ_{path} and signal correlation threshold σ_{signal} to find matching pairs.

In order to evaluate the accuracy of our matching algorithm, we have to obtain the ground truth on how the different segments match through manual tagging. Figure 9 shows the receiver operating characteristic (ROC) curve for both large office floor and small research lab. Both curves show high performances of matching with large area under the curve. A good operating point can be chosen using the $y = x$ line. This operating point provides a guide for choosing the appropriate thresholds for the path and signal correlation values to use for matching.

VI. Floor Plan Construction

A. Algorithm

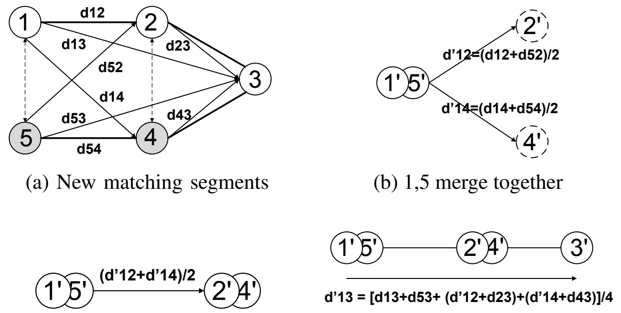
In PiLoc, the inaccuracy in the IMU and WiFi signal strength measurement places challenges in merging trajectories

Algorithm 1: **Floor Plan Construction Algorithm**

```
Input: Matching result $\mathcal{M}$, Trajectories set $T$ of 1 cluster $c$
Output: Updated displacement matrix $M_d$
Initialized displacement matrix $M_d$;
for each matching segment pair $(S_i, S_j)$ in $\mathcal{M}$ do
    // Collocate and determine displacements of
    // matching steps
    Set of collocated steps, $S_{merge}$, is initially empty;
    for each matching step pair $(s_m, s_n)$ in $(S_i, S_j)$ do
        Place $s_m$, $s_n$ into a single location;
        New displacement of $s_m$ and $s_n$ are average
        displacements of $s_m$ and $s_n$ to all points in $S_{merge}$;
        $S_{merge} = S_{merge} \bigcup s_n \bigcup s_m$;
    end
    for each step $p$ in $T$ or but not in $S_{merge}$ do
        Displacement of $p$ = average displacements of $p$ to all
        points in $S_{merge}$;
    end
    Update displacement matrix $M_d$ based on all new
    displacements calculated;
end
return $M_d$;
```

from different users. PiLoc addresses this challenge by merging and filtering all users inputs in the floor plan construction algorithm. The trajectory matching algorithm discussed in the previous section generates matching pairs for all segments from the same indoor environment. The output of the matching algorithm $\mathcal{M} = \{(S_1, S_2), ..., (S_i, S_j)\}$ contains pairs of matched path segments and these matching pairs are used as inputs to the algorithm.

Initialization. PiLoc merges and generates floor plans for all trajectories T collected in the same indoor environment, i.e., in the same AP cluster c as discussed in Section V-A1. In the initialization phase, PiLoc builds a displacement matrix M_d. Given two steps with global ID i and j, each belonging to one of the two matching segment pairs, the entry $M_d[i][j]$ gives the 2D displacement (x, y) between the positions indicated by the two steps as $(x_j - x_i, y_j - y_i)$. The displacement between two steps can only be measured if there are common matching path segments that can relate them. The displacement is "undefined" if the steps are from two different trajectories with no relationship.



(a) New matching segments

(b) 1,5 merge together

(c) New displacements calculated for 2 and 4

(d) New displacement calculated for 3

Fig. 10: Example of Motion Vector Merging. d_{ij} denotes the current displacement and d'_{ij} denotes the new displacement.

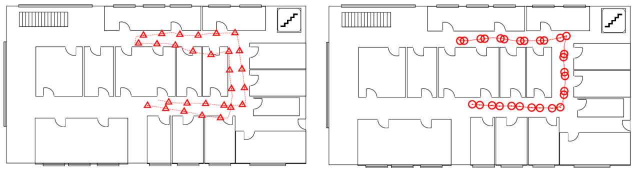
Iteration. In the iteration phase, each matching segment pair (S_i, S_j) is taken into account to update the displacement matrix. Recall that matching segments have the same number of steps. For each pair of matching steps (s_m, s_n), we "move" the starting position of these steps so that they start at the same point. Then, we compute the new displacements by finding the average displacements of these steps to those steps whose new displacements have been determined. The detail algorithm is shown in the Algorithm 18.

As an illustration, consider Figure 10. The trajectory consists of five steps $\{1, 2, 3, 4, 5\}$. $S_1 = \{1, 2\}$ and $S_2 = \{5, 4\}$ are the only pair of matching segments in this example. The algorithm first computes the starting (relative) position of the first matching steps. Figure 10(a) shows the original displacements of the points in the trajectory.

In Figure 10(b), the starting point of the first pair of matching steps $\{1, 5\}$ are considered to be at the same location (shown as $1'$ and $5'$ in the figure). In order to calculate the new displacements for the next pair of matching steps $\{2, 4\}$, which is again assumed to be collocated, the new displacements d'_{12} and d'_{14} are computed as $\frac{d_{12}+d_{52}+d_{14}+d_{54}}{4}$, as shown in Figure 10(c).

After the new displacements for all matching steps in this segments have been computed, the displacement of all the other steps are updated. As shown in Figure 10(d), the displacement d'_{13} is determined by averaging the displacements to all four matched steps.

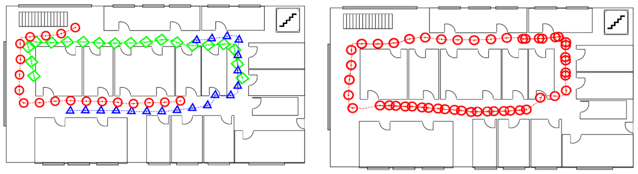
Since the matching pair can either be from the same trajectory or different trajectories, the floor plan construction algorithm works for both intra-graph merging and inter-graphs merging. As shown in Figure 11, the trajectory is refined internally and merged with itself using the algorithm. The error cumulated in dead-reckoning is corrected using data within the same trajectory. Figure 12 shows the merging of different trajectories collected from the same floor. Note that since each step carries fingerprint data in the floor plan constructed, it naturally can serve as the radio map to handle localization queries and decide the current user location in the map. Since the merging algorithm works for all geographically separated clusters, floor plans and radio maps are generated for all different indoor environments covered by the participating users.



(a) Before

(b) After

Fig. 11: Intra Trajectory Merging



(a) Before

(b) After

Fig. 12: Inter Trajectory Merging

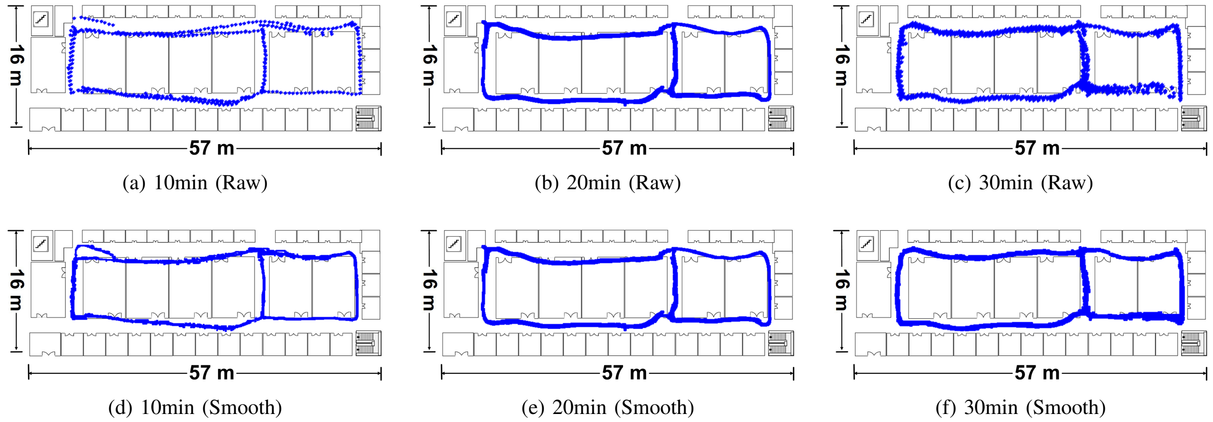
The maps generated are relative maps, i.e., the locations in the map are not associated with the absolute location yet. To map the floor plan to the real locations in the indoor environment, PiLoc only requires at least one point to be associated with a GPS coordinate. This point becomes a global reference point and all the locations of the rest points in the maps can be fixed.

B. Floor Plan Filtering

Filtering is required to remove the noisy samples and trajectories in the floor plan construction process. Trajectories that have no matching segments are first filtered out after the matching process. Therefore the outlier trajectories will not be reflected in the final results. To further smooth the constructed floor plans, we adopt a grid-based filtering scheme. The generated floor plans are divided into $1 \times 1m^2$ grids. We observed that most grids that contain correct walking trajectories have more steps than the average number of steps over all grids in the floor plans generated by the trajectory merging algorithm. In the final floor plan constructed, all grids with number of steps less than the averaged will be removed. To smooth the floor plan constructed, morphological operators dilation and erosion [30] are used, and the extracted contour from the erosion result are used as the smoothed walking paths.

C. Floor Plan Evolution

To reflect the environmental changes and new user inputs, the floor plan generated needs to be periodically updated. One important feature of PiLoc is the floor plans will keep evolving with continuous incoming of user inputs. And the evolvement is also fully automatic. In PiLoc, the floor plan is updated every 10 minutes to handle the new user input. All new data will be clustered into the existing clusters or new clusters (e.g.,



(a) 10min (Raw)

(b) 20min (Raw)

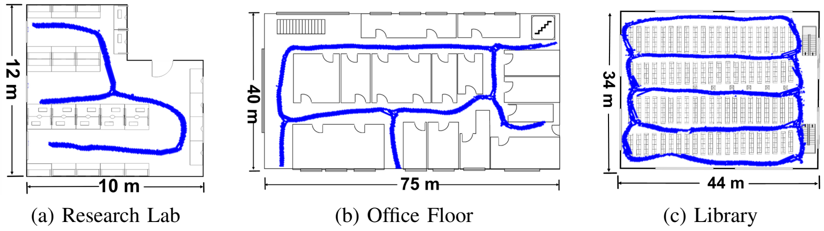
(c) 30min (Raw)

(d) 10min (Smooth)

(e) 20min (Smooth)

(f) 30min (Smooth)

Fig. 13: Floor Plan Evolving



(a) Research Lab

(b) Office Floor

(c) Library

Fig. 14: Floor Plan Construction For Varies Indoor Environment

new floors) may be generated. As shown in Figure 13, the floor plan is updated every 10 minutes to generate an evolving indoor map. The radio maps are also updated in the same process to keep an up-to-date localization database.

D. *PiLoc Localization*

PiLoc adopts fingerprint-based approach for indoor localization. The radio maps is automatically built and updated by merging user contributed walking data. In this way, PiLoc is able to handle localization queries and return the user location using the radio map and input fingerprints. Previous works such as RADAR [21] utilizes the fingerprint database by using the nearest neighbors from the query point to the reference points in the database as the similarity metric.

Such an approach works relatively well for indoor areas with sparse AP deployments (In RADAR, only three APs are presented). However, during our data collection we observe that many indoor environments have very dense AP deployments (More than 100 in one floor). Nearest neighbor matching works poorly in the dense AP environment. This is because at each location smartphones can observe a long list of remote APs with RSS ranging from -80dbm to -90dbm. The RSS fluctuations of large number of these remote APs overwhelm the small set of nearby APs in calculating the similarity. However, nearby APs are more important in deciding the current location of the user since high RSS values only covers a small area for each AP. Based on this observation, PiLoc uses the simple but more effective *weighted maximum similarity* as the metric:

$$WMS = \sum_{i=1}^{n} \omega_i' \cdot \frac{1}{max\{|r_i - r_i'|, 1\}} \tag{6}$$

where n is the total number of APs, $\omega_i' = 1/|\mu_i|$ is the weight of ith AP and is inverse to the absolute of its mean value. Therefore nearby APs with higher average RSS value will have higher weight. r_i is the input RSS of AP_i and r_i' is from the radio map. WMS will get higher value if the input point and reference have more common APs and the RSS difference for nearby APs are smaller. The location will be determined by the maximum WMS matching in the radio map. The PiLoc localization provides better accuracy than the conventional approach especially in the dense AP environment. The evaluation will be discussed in the next section.

VII. Evaluation

A. *Implementation*

PiLoc has both client and server components. The client performs two functions: *data collection* and *issues localization query*. For data collection, the client runs an Android smartphone service in the background to opportunistically collect walking trajectories and radio fingerprints. For localization, the client issues queries to the server to localize the phone. The server collects user uploaded trajectory and fingerprint data. It uses the data collected to construct and update the

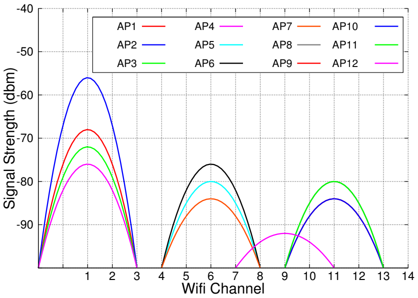


Fig. 15: WiFi Signal Graph

floor plans periodically for all indoor environments it has data for. For each localization query, the server first finds out the correct radio map to use based on the AP clustering result. Then, the weighted maximum similarity match is used to find the best matching location of the phone.

B. Data

The experiment data is collected over a one month period from 4 different areas which cover 5528 m^2 areas in total. The layouts are shown in Figure 13 and 14. The sizes of these 4 different floors range from 120 m^2 to 3000 m^2. The smallest area of 120 m^2 involves the inside of a research lab with lots of partitions which pose a special challenge due to its very short turns and walk ways. Three different phone models are used: Google Galaxy Nexus, Samsung S3 and Samsung S4. All phones run Android OS. An average of 35 APs are detected in each of the 4 areas. In total, 600 user trajectories are recorded, which contains 90,000 steps and each step is associated with direction as well as WiFi fingerprint. In terms of time, these data corresponds to 700min of data collection. If collection is done by say 5 users in parallel, data collection time will be reduced accordingly to 140min.

To increase the fingerprint sampling rate, we only scan Channel 1 (2412MHz), 6 (2437MHz) and, 11 (2462MHz). These channels are non-overlapping with the commonly deployed 802.11 b/g/n [31] network. As shown in Figure 15, these 3 channels cover most of the deployed APs in the environment we measured. In our scan, we also include one channel (5240MHz) from the less commonly deployed 802.11a network. By reducing the number of channels scanned and improving the efficiency of the code, we significantly increase the sampling rate. On the average, three radio fingerprints can be collected per step, compared to using the Android WifiManager which can only collect one sample every three steps. The average fingerprint per step is computed by combing all fingerprints collected between two consecutive steps.

C. Performance

1) Evaluation Metrics: We evaluate the overall performance of PiLoc by looking at the quality of floor plan constructed and localization accuracy. Two metrics are used in the measurement:

- **Step Mapping Error (SME).** The floor plan constructed maps steps of walking trajectories into the real floor plan. The step mapping error measures how accurate the trajectories fit the real floor plan. Since fingerprints are associated with each step, less step mapping error results in higher fingerprint mapping accuracy, which directly affects the localization accuracy. The SME is defined as:
$$SME = \|L(s) - L(s')\| \tag{7}$$
where $L(s)$, $L(s')$ are mapped location of the step and the ground truth location of the step respectively. A smaller SME reflected better matching of the constructed floor plan to the real one. To establish the ground truth, locations where each step is taken in the reference floor plan are manually tagged. Since each step has a globally unique identifier, the location of one particular step in the constructed floor plan can be obtained by querying the ID, and SMEs are measured by calculating the differences between the estimated step locations and their respective ground truth locations.
- **Localization Error (LLE).** LLE measures how well the location given the localization server matches the ground truth location of the phone.
$$LLE = \|L(p) - L(p')\| \tag{8}$$
where $L(p)$ is the estimated location and $L(p')$ is the real location of the phone. The smaller the euclidean distance, the better the localization quality.

2) Floor Plan Construction: To measure SME, each step associated with fingerprints is assigned a global ID. We tagged the ground truth localization for each collected step and measure the SME in the constructed floor plan. We plot the CDF for both the mid-size (900 m^2) office floor and 120 m^2 research lab.

Figure 16 and 17 shows three different CDF curves for the office floor and research lab. Each CDF curve corresponds to different time period of data collection, ranging from 10min to 30min. For the mid size office area shown in Figure 16, PiLoc achieves average SME of 1.65m, 1.47m and 1.27m for 10min, 20min and 30min of data collection respectively.

For the research lab, PiLoc achieves average SME of 0.54m, 0.6m and 0.46m for 10min, 20min and 30min of data collection respectively. Surprisingly, the accuracy for the research lab is better, probably because the step counting mechanism used incurs much less error for short distances.

3) Localization: Localization evaluation is performed for the large office floor (3000 m^2) and research lab. As shown in Figure 18, PiLoc achieves an average LLE of 1.37m for the research lab, with 80% of the errors less than 2.3m.

System	Average LLE	Effort
RADAR [21]	2∼ 5m	Site survey
Horus [22]	∼ 1m	Site survey
Zee [3]	1∼ 3m	Floor plan
UnLoc [2]	1∼ 2m	Door localization
LiFS [1]	3∼ 7m	Floor plan, time-consuming
PiLoc	1∼ 3m	Self-calibrate

TABLE I: Listing of related localization systems

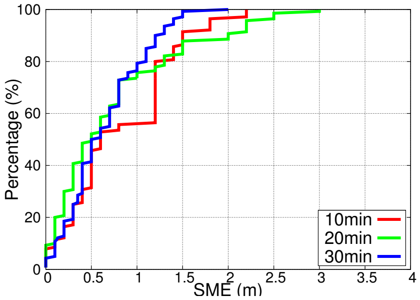


Fig. 16: CDF of SME ($900m^2$ Office Floor)

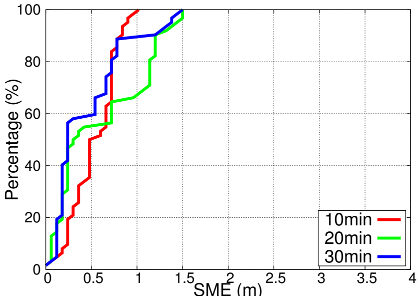


Fig. 17: CDF of SME (120 m^2 Research Lab)

For the large office floor, the average LLE is 1.58m with 80% of the errors less than 3m.

Table I provides a brief summary and qualitative comparison between PiLoc and other localization systems. As the evaluations are performed in different settings, the localization errors listed (obtained from the respective papers) can only provide a high level guide on the relative performances of the various systems. Even though PiLoc does not require manual calibration and landmarks, it could achieve localization accuracy that are comparable with the other localization schemes.

VIII. Discussions

A. Applications

Indoor localization plays a very important role in many real world applications. For example, location-based services and location-based advertisements nowadays have gained increasing popularity. However, deploying and maintaining current indoor localization schemes requires too much effort and that hinders the development of location-based applications. By opportunistically collecting walking trajectories from causal users whose roles are not dedicated to localization, the localization system can be easily built and updated with PiLoc. For example, the movement of security guards or even any other users can contribute traces for constructing the indoor floor plan of any given indoor environment. PiLoc enables an efficient way to leverage the daily human movements for localization and has the potential to be deployed in a large scale.

B. Limitations

PiLoc currently extracts turn segments and line segments for matching. Extending to more complicated layouts containing curve shapes requires extracting additional curve segments. In PiLoc, path correlation and signal correlation are used for trajectory matching to construct pathway floor plans. In open spaces where people may not walk along distinct walkways, path correlation and signal correlation may fail to differentiate intersecting or parallel aisles that are not separated by sufficiently large distances. This is one limitation of PiLoc. However, in practice walking paths inside buildings are often separated by walls or other obstacles. This will result in differences in signal correlations that can be distinguished by PiLoc.

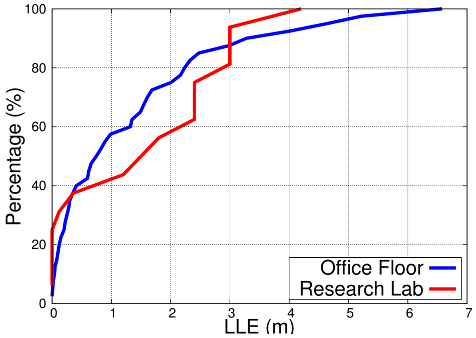


Fig. 18: CDF of LLE

C. Extensions

1) Diverse Floor Plans: In PiLoc, path segments are extracted and clustered for efficient matching. These path segments reflect the physical layouts of the floor plans. Though most indoor floor plans have rectangular layouts, there are indoor layouts that may contain curved walking paths. While a curved walking path may be captured as a series of straight lines and turns, the inaccuracy introduced can be substantial. Hence, to achieve a higher accuracy for these types of floor plans, we may have to include additional types of walking paths. Conceptually, adding additional path segment shapes in PiLoc is straight-forward, though the actual process of extracting these new shapes may be much more complex. Nevertheless, once the new paths are extracted, there is no change in the rest of the algorithms. The current architecture is thus highly extensible to diverse floor plans.

2) Multiple Floors: Currently, PiLoc constructs floor plans for different buildings and different floors. Different floor plans will be constructed for each AP cluster. However, these floor plans are independent to each other due to lack of links between each other. A more complete floor plan for the indoor environment would need to support 3D construction. In PiLoc, even though different floors in one building have AP overlaps and belong to one AP cluster, they don't have matching pairs hence multiple floor plans will be returned for all different floors. To link these floor plans to extend PiLoc from 2D to 3D, additional $z\text{-}axis$ information is required. As barometers are becoming more and more common in current smartphones, acquiring $z\text{-}axis$ (sea level) information would become easier. However, barometers readings are known to be sensitive to environmental changes, especially in the indoor environment (e.g., air pressure change due to air conditioners). How to robustly incorporate z-$axis$ into the system is a challenge and a possible future extension to PiLoc.

3) Multiple Fingerprints: PiLoc utilizes WiFi fingerprints for localization. However, WiFi fingerprints are not tightly bound to our systems. Different fingerprints such as FM radio signal [9] or even ambient noise [10] can be associated with each step and used in the localization phase. Also, to improve the performance, other fingerprints such as indoor magnetic fingerprints can also be added into the system to provide more information.

IX. Conclusion

In this paper we propose and evaluate PiLoc, an indoor localization scheme that takes user walking trajectories as input and automatically builds and updates the indoor floor plan. By incorporating radio fingerprints, indoor radio map is also automatically managed by PiLoc. PiLoc requires no human intervention and can achieve high localization accuracy with 1.5 meter error on the average. As PiLoc enables minimum user effort for calibration and maintenance, it has potential for large scale deployment.

Acknowledgement

This research was carried out at the NUS-ZJU Sensor-Enhanced Social Media (SeSaMe) Centre. It is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the Interactive Digital Media Programme Office.

References

[1] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *MobiCom*. ACM, 2012.

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

[3] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *MobiCom*. ACM, 2012.

[4] F. Adib and D. Katabi, "See through walls with wi-fi!" in *SIGCOMM*. ACM, 2013.

[5] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *TMC*, 2010.

[6] J. Wilson and N. Patwari, "See-through walls: Motion tracking using variance-based radio tomography networks," *TMC*, 2011.

[7] J. Xiong and K. Jamieson, "Arraytrack: a fine-grained indoor location system," *HotMobile*, 2012.

[8] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained indoor localization via smartphone," in *MobiSys*. ACM, 2013.

[9] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "Fm-based indoor localization," in *MobiSys*. ACM, 2012.

[10] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *MobiCom*. ACM, 2009.

[11] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: indoor pathway mapping made easy," in *NSDI*. USENIX, 2013.

[12] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *MobiCom*. ACM, 2010.

[13] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration indoor localization over ieee 802.11 wireless infrastructure," *Wireless Networks*, 2010.

[14] Y. Yang and A. E. Fathy, "See-through-wall imaging using ultra wideband short-pulse radar system," in *Antennas and Propagation Society International Symposium*. IEEE, 2005.

[15] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," in *MobiCom*. ACM, 2007.

[16] M. Seifeldin, A. Saeed, A. Kosba, A. El-Keyi, and M. Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *TMC*, 2012.

[17] C. Xu, B. Firner, R. S. Moore, Y. Zhang, W. Trappe, R. Howard, F. Zhang, and N. An, "Scpl: indoor device-free multi-subject counting and localization using radio signal strength," in *IPSN*. ACM, 2013.

[18] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *TOIS*, 1992.

[19] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: indoor location sensing using active rfid," *Wireless networks*, 2004.

[20] N. B. Priyantha, "The cricket indoor location system," Ph.D. dissertation, MIT, 2005.

[21] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM*. IEEE, 2000.

[22] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *MobiSys*. ACM, 2005.

[23] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona lisa: spot localization using phy layer information," in *MobiSys*. ACM, 2012.

[24] H. Lim, L.-C. Kung, J. Hou, and H. Luo, "Zero-configuration, robust indoor localization: Theory and experimentation," in *INFOCOM*. IEEE, 2006.

[25] Y. Kim, Y. Chon, and H. Cha, "Smartphone-based collaborative and autonomous radio fingerprinting," *TSMC*, 2012.

[26] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using gaussian process latent variable models." in *IJCAI*, 2007.

[27] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Ubicomp*. ACM, 2012.

[28] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am i facing: Inferring horizontal device orientation from an accelerometer signal," in *ISWC*. IEEE, 2009.

[29] A. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu," in *WISP*. IEEE, 2009.

[30] Erosion and dilatation. http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html. [Online]. Available: http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

[31] J. Niu, B. Lu, L. Cheng, Y. Gu, and L. Shu, "Ziloc: Energy efficient wifi fingerprint-based localization with low-power radio," in *WCNC*. IEEE, 2013.