

On Plug-and-Play Indoor Navigation with On-the-Fly Map

Paper ID #93

ABSTRACT

Existing studies on indoor navigation often require such a pre-deployment as the floor map, localization system and/or additional (customized) hardwares, making them prohibitive when the situation deviates from these requirements. We observe that when there are sufficient participants, the collected WiFi signatures by participants can serve as the fingerprints (referred to as *location fingerprints*) of their unknown locations. By computing the relative positions of these location fingerprints we can connect them to form a global map. Such a map can reflect the topology of the underlying walkable space and thus holds the potential of offering a navigation path for any intended users. Based on this observation, we design PIN, a Plug-and-play Indoor Navigation system via online crowdsourcing by participants with commercial off-the-shelf (COTS) mobile devices. Specifically, each participant uploads the sensory data (including WiFi information, compass readings, barometer data, etc.) with the built-in sensors of the device. The server generates a global map through a series of operations including local map generation and map stitch, edge computation and level-change detection. On top of on-the-fly map, a navigation path is computed and displayed on the user's device. We implement the prototype of PIN and our experiments show that PIN can quickly generate a navigation path with a guaranteed successful rate.

General Terms

Design, Algorithms, Performance

Keywords

Indoor navigation, MDS, Crowdsourcing

1. INTRODUCTION

With the increasing of indoors social, business and personal activities, people have a growing demand for indoor navigation services. Such a service holds the potential for significantly saving the wandering time reaching their destinations (e.g., meeting their friends, arriving at the shops of interests, etc.). As well known, GPS does not work well in indoor environments, leading to indoor navigation as the last-mile problem. Even

though it has been well studied for decades, the progress for widely deploying indoor navigation services is still quite low; perhaps the reasons behind this dilemma mainly stem from the fact that current navigation systems, either landmark-based or location-based, have their own shortcomings.

The landmark-based navigation paradigm has been witnessed for decades where direction boards can be found almost everywhere in urban areas, indoors or outdoors, and these direction boards provide an important guide for turn-by-turn navigation. That is, when a navigation user observe a direction board, s/he can determine the walking direction for next movement until finding another direction board or reaching the destination. In particular, the landmark-based indoor navigation systems, which exploit the direction boards or other special devices [7] deployed indoors as landmarks, have been active for a long time in such human populated scenarios as shopping malls, train stations, airports, and conference hotels. For example, assume that Alice is on the way of visiting her friend Bob at Room 6061 of a grand hotel unfamiliar with Alice. After arriving at the hotel and taking an elevator from the Lobby to Level 6, Alice observes that there are direction instructions on the wall, as shown in Fig. 1 (a), which successfully guide her movement to the destination after a right turn and two left turns. Clearly, the navigation success in such scenarios relies on accurate instructions directing the next movement, and interestingly, *the accurate localization system is no longer necessary for the landmark-based navigation* since Alice will not get lost as long as she follows the instructions, no matter how far away from the landmarks. The undesirabilities, however, are that 1) it requires user interference for reading direction instructions, 2) it incurs a significant time delay since, without the global picture of landmarks, users will take a while to determine the walking direction when encountering a new landmark, and 3) it requires densely pre-deployed direction boards or beacons, and is thus not applicable for scenarios without sufficient landmarks. For instance, in a typical shopping mall, it is generally impractical to densely deploy landmarks

covering all junctions for intended navigation users.

Thanks to the development of mobile communication technologies and the deep penetration of smart devices, recently a new paradigm based on the floor map and localization system attracts evergrowing attentions from researchers and practitioner. Let's take Fig. 1 (b) as an example. With the pre-deployed floor map (either manually or in a crowdsourcing way [3,10,13]) and pre-deployed radio maps, Alice can be located via previous schemes [5,6,8,33,39], e.g., by matching the WiFi fingerprint against the fingerprint database, and a navigation path to the destination (e.g., Room 6061) as an input is thus computed on the floor map. The progress is tracked by periodically updating the Alice's location until she reaches Room 6061. Even though this new paradigm has no disadvantages of the landmark-based navigation system, the requirements on the pre-deployment of costly floor map and radio maps, the unsatisfied localization accuracy due to unstable WiFi signals over time, together with its suffering from device and usage diversity [27], still hinder it from being widely deployed in practice.

A question naturally arises: "can we design an indoor navigation system such that the disadvantages of the above-mentioned two paradigms are avoided, while the navigation success is guaranteed?" In this paper, we strive to answer the above question by combining the key ideas of the landmark-based navigation and the location-based navigation. Before digging into the details of our system design, let's take a look at the key insights of these two navigation paradigms. The most essential part in navigation is to correctly prompt the change of walking directions (e.g., turning left/right, going upstairs/downstairs, etc.) at the right turning points, and if no navigation instructions are offered, the users only need to keep walking straight. For the landmark-based navigation, a navigation user will not change the walking direction until s/he finds an instruction asking for changing; once the user follows the landmarks s/he will not miss the destination. For the localization-based navigation, the discrete fingerprints can be regarded as a sample of the indoor walkable space, and the fingerprint database with items of the form $\langle \text{location}_i, \text{fingerprint}_i \rangle$ maps the fingerprinted point cloud to the underlying floor map, such that a user is periodically located based on fingerprint matching. However, as these fingerprints are not self-connected into a graph, without the floor map the localization-based navigation systems have no capabilities for computing the navigation path at all, let alone providing accurate instructions on when and where to change the walking directions. Moreover, as both of these two paradigms require the pre-deployments (e.g., direction boards, floor maps or radio maps), they are not feasible for plug-and-play indoor navigation where no such

pre-deployments are available.

We put together all these key insights to solve the problem of plug-and-play indoor navigation without any pre-deployment. The core idea is to treat the active users, walking indoors with mobile devices for instantly sensing WiFi signals from the surrounding WiFi APs, as a discrete sample of the underlying walkable space. These WiFi signals can serve as fingerprints of the corresponding locations (i.e., there is a one-to-one correspondence between location fingerprints and user locations). The relative positions between users (e.g., User A and User B walk on different but adjacent corridors), if can be computed with the location fingerprints, can be used to derive a weighted and directed graph. In this graph, each user (or location fingerprint) corresponds to a point, and two users nearby form an edge with a real-valued length and a direction. Please see Fig. 1 (c), where the left sub-figure depicts the users distributed along the corridors between Alice and Bob, and the right sub-figure illustrates the corresponding map where the edges between adjacent users/points are indicated by the red lines. Each edge is assigned a length and direction, and the shortest path between Alice and Bob is then computed on the on-the-fly map. The dashed lines indicate walking directions with the assist of the generated map.

We implement this idea and develop PIN, an on-the-fly map based plug-and-play indoor navigation system by offering accurate *turn-by-turn instructions*. PIN has two key components: active users with smart devices and a cloud server. The active users walking in the indoor space of interests upload the data sensed within a few seconds (including WiFi RSS lists, barometer, gyroscope and compass readings) to the cloud server; the server then computes a distance matrix between pairwise location fingerprints based on the proposed metric called *WiFi similarity*, and generate an on-the-fly map via a series of processes such as two-stage clustering and local map generation, local map stitch and refinement, and the computation of edge length and direction, and so on. On top of the derived map, a navigation path, which consists of weighted and directed edges (including turning edges which require the turning actions), from each user to the destination is generated. Specifically, to compute the between-user walkable distance (and thus the edge length), we propose a novel metric τ to measure the WiFi similarity with the sensed WiFi information by users, and then build a linear model between τ and walkable distance. With this model, the walkable distance (thus the edge length) between pairwise users can be estimated based on the WiFi information uploaded by the users. Also, during navigation, we estimate the stride length based on WiFi similarity and compute the displacement by counting walking steps, and thus navigate users with turn-by-turn instructions



Figure 1: An illustrative example. (a) A scenario for turn-by-turn navigation; (b) A typical indoor navigation based on floor map and localization system; (c) A new dimension of indoor navigation. The active users in an indoor environment (left) can be used for on-the-fly map generation (right) where each point corresponds to the location where a user uploads the sensory data, and the dashed line indicates the walking direction.

to their destinations, according to the computed navigation paths.

One previous work close to ours is NearMe [17]. NearMe estimates the distance between two users based on the similarity of WiFi signatures, i.e., the number of common APs and a rank-based correlation coefficient, say spearman ρ . The primal goal of NearMe, however, is simply to find the nearest persons or devices, instead of guiding a user to reach them. The root mean square (RMS) error of NearMe in the testing experiments is about 14 meters, which is not accurate enough for the corridor-to-corridor indoor navigation, and it is inapplicable in areas with densely deployed WiFi APs. In addition, even with the estimated distance of high accuracy, it still faces many challenges, e.g., to correctly embed the location fingerprints without incurring incorrect flips [19]¹, to properly determine the relationship between location fingerprints (including edge computation and level-change detection) for the ease of plug-and-play navigation, etc.

In reality there are many scenarios requiring the plug-and-play indoor navigation where no pre-deployment is available, and importantly absolute locations of users are not demanded. For instance, when Alice is on the way to visit Bob, Bob is attending a technical session or an important meeting and can not answer the phone [7]. Bob can simply turn on the system such that PIN can compute a navigation path and guide Alice to where Bob is. Other intended scenarios include finding the lost child carrying a device such as a smart watch, locating the missing bag with a smart device inside, quickly finding the nearby promotion products issued by shop owners, or returning to the user's car equipped with smart devices and parked in an unfamiliar parking lot,

to name a few. In these scenarios, the targets (e.g., the busy Bob, the missing child, etc.) cannot proactively tell the current location such that existing navigation systems can not work. The system PIN, however, can compute a path based on the sensory data crowdsourced by the target and other occupants, hereon navigate the user to the destination, i.e., the target's location. Since for each user, in the generated map the starting point to the destination is known, there is no need for user lock-on by adopting localization schemes for determining the accurate location of the user; at the same time, the user can be tracked simply by computing the walking distance, and thus no accurate localization is involved during navigation.

In summary, our major contributions are summarized as follows.

- We present a new perspective of plug-and-play indoor navigation by incorporating the active users with COTS mobile devices into on-the-fly map generation, where no pre-deployment of floor maps, localization systems, or direction boards is involved.
- We study how to connect the point cloud into a graph, and develop a metric for estimating the walkable distance between users solely relying on the collected WiFi signals. Also we estimate the stride length of a navigation user by using WiFi signals.
- We design an Android-platform navigation system of PIN, and evaluate the performance via extensive experiments. The results show that the generated on-the-fly map is promising for plug-and-play indoor navigation, and the 80-percentile of between-user distance error is around 5 meters.

The rest of the paper is organized as follows. Section 2 presents the preliminary of our paper. We detail the system design in Section 3, and evaluate the system in Section 4. In Section 5 we discuss the impact of user

¹The flipping problem is very common in network localization, where there are multiple valid embeddings under the same constraints, e.g., between-object edge lengths. Hence, it is of particular importance to resolve the flip ambiguity for deriving the correct global layout of location fingerprints, which otherwise may lead to wrong turning instructions.

density, sensor desert, etc. Section 6 is devoted to the related work, and finally, Section 7 concludes the paper.

2. PRELIMINARY KNOWLEDGE

2.1 Multi-Dimensional Scaling Based Map Generation

In this paper we exploit the usage of WiFi signals for estimating Euclidean distances between pairwise location fingerprints sensed by active users, and then apply classical multi-dimensional scaling (CMDS) techniques to derive a local relative map, offering accurate navigation instructions for users. MDS is a non-linear dimension reduction technique for visualizing similarity levels of individual cases (e.g., active users) with multiple attributes, based on the between-case distance matrix D [2], where each case is assigned a coordinate in an m -dimensional space ($m = 2, 3$) such that the distances between objects are preserved as well as possible. The MDS techniques have been successfully used for computing WiFi AP location [16], user locations [31], and sensor network localization [22]. Among many MDS techniques, the CMDS technique is promising as it can yield a fast and closed-form solution: given an input of between-case dissimilarities matrix, it outputs a coordinate matrix with the configuration of minimizing the so-called *strain* defined below.

Define the inner product matrix, denoted by B , of an $n \times n$ distance matrix D as $B = -\frac{1}{2}H D H^T$ where $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, I is an n -order unit matrix and $\mathbf{1}$ is an n -dimensional vector of all ones. Then the objective of the CMDS technique is to minimize the following loss function (which is also called strain in CMDS):

$$Strain_D(x_1, x_2, \dots, x_n) = \sqrt{\frac{\sum_{ij} (b_{ij} - \langle x_i, x_j \rangle)^2}{\sum_{ij} b_{ij}^2}} \quad (1)$$

where $i, j (i, j = 1, 2, \dots, n)$ represent the indices of the i -th and the j -th object, respectively, and x_i, x_j denote the coordinates of objects i and j , respectively.

The closed-form solution of the CMDS to Equation (1) is based on the eigenvalue decomposition of matrix B . Denote by Λ, Q the diagonal matrix of the eigenvalues and the eigenvector matrix of B , respectively. The eigenvalue decomposition of B can thus be expressed as $B = Q\Lambda Q$. According to the CMDS, the output coordinates of the i -th object can be given as

$$X_i = (\sqrt{\lambda_{(1)}}q_{1i}, \sqrt{\lambda_{(2)}}q_{2i}, \dots, \sqrt{\lambda_{(m)}}q_{mi}) \quad (2)$$

where $\lambda_{(k)} (k = 1, 2, \dots, m)$ is the k -th largest eigenvalue and q_{ki} is the i -th value of the k -th eigenvector. In this paper, we first compute the coordinates of the location fingerprints for users on the same floor, and then merge the coordinate system across multiple floors to derive a global map. As such, we only need to com-

pute the top 2 largest eigenvalues and eigenvectors, and derive a 2-dimensional coordinate system.

2.2 Between-User Dissimilarity Computation with WiFi Signals

CMDS requires a dissimilarity (e.g., Euclidean distance) matrix as an input. To this end, we need to calculate the dissimilarities between location fingerprints at first. An intuitive way is to exploit the WiFi RSS distance, defined on the received WiFi signal strength (RSS) crowdsourced by active users. The WiFi RSS distance has been revealed to be proportional to the Euclidean distance between nearby users/points [41]. However, due to device diversity and usage diversity, the WiFi RSS distance varies greatly even for two pairwise users with equal distances, making it less valuable for estimating the between-user dissimilarity. If such an estimation is off, the results by CMDS will not be good.

Our key observations are: 1) when two users are very close, the received WiFi signals are more likely from the same WiFi APs and significantly different otherwise; 2) even though the WiFi signals are very unstable and suffer from device diversities, the relative orders (or, the ranks) of WiFi RSS by close users remain stable (e.g., if User 1 receives a stronger signal from WiFi AP 1 than WiFi AP2, then this relationship likely holds for nearby users too). Fig. 2 corroborates our observations where 3 pairs of close users, i.e., User 1 and User 2, User 6 and User 7, User 13 and User 14, sense more common WiFi APs as shown in Fig. 2 (b), while other pairwise users, which are separated relatively far away or due to obstacles between them (e.g., User 1 and User 6, or User 1 and User 13, etc.) as shown in Fig. 2 (c), share less common WiFi APs. In other words, there exists an obvious correlation relationship between close users in terms of the number of common WiFi APs and relative orders of WiFi RSS values from these APs [17].

Based on the key observations, we thus propose to exploit the rate of common WiFi APs and the ranks of WiFi RSS values for constructing a robust between-user WiFi similarity. Specifically, for any pairwise users v_i, v_j , let the collected WiFi RSS lists, including the BSSID (i.e., mac address) and the RSS value of each WiFi AP, be $F(i)$ and $F(j)$, respectively. That is, $F_1(i)$ (or $F_1(j)$) denotes the list of BSSID sensed by user v_i (or v_j), and $F_2(i)$ (or $F_2(j)$) is the corresponding RSS value list. Then the rate of common APs is defined as $\lambda(i, j) = \frac{|F_1(i) \cap F_1(j)|}{|F_1(i) \cup F_1(j)|}$ where $|\cdot|$ represents the cardinality of a set. Without loss of generality, we assume that $F_1^k(i)$ and $F_2^k(i) (k = 1, 2, \dots, l(i, j))$ are the RSS values from the k -th common WiFi AP by users v_i and v_j respectively, where $l(i, j) = |F_1(i) \cap F_1(j)|$ is the number of common WiFi APs. Then, For two pairs of RSS values of v_i, v_j , say $(F_2^k(i), F_2^k(j))$ and $(F_2^m(i), F_2^m(j)) (k \neq m)$, we call them *concordant* if

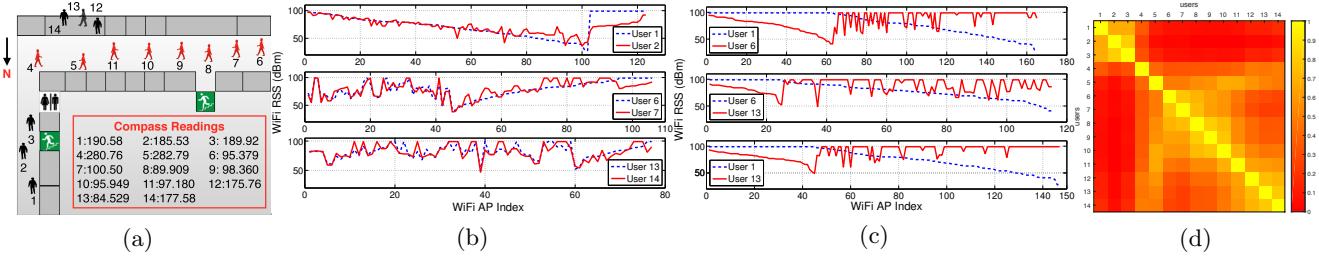


Figure 2: A motivated scenario. (a) The 14 active users with COTS mobile devices distribute along the corridors and inside a room; (b) The WiFi RSS curves on WiFi APs for three pairs of close users show that there are many common WiFi APs with roughly the same trend of RSS; (c) When users are separated far away, the number of common WiFi APs decreases, and the RSS values become less relevant; (d) The confusion matrix of similarities between the numbered users.

$(F_2^k(i) - F_2^m(i))(F_2^k(j) - F_2^m(j) + \sigma) \geq 0$ holds, and *disconcordant* if $(F_1^k(i) - F_1^m(i))(F_1^k(j) - F_1^m(j) + \sigma) < 0$ holds. Here the parameter σ (in our experiments, $\sigma = 5$ dBm) is introduced to control the effects of signal fluctuation and device diversity, as our experimental evidence shows that when two users are very close, the relative order of WiFi RSS values from the common APs may be slightly different. Let the number of concordant pairs $N_c^{ij} = \sum_{1 \leq k < m \leq l} I\{(F_2^k(i) - F_2^m(i))(F_2^k(j) - F_2^m(j) + \sigma) \geq 0\}$, and the number of dis-concordant pairs $N_d^{ij} = \sum_{1 \leq k < m \leq l} I\{(F_2^k(i) - F_2^m(i))(F_2^k(j) - F_2^m(j) + \sigma) < 0\}$. The correlation coefficient $\rho(i, j)$ between two location fingerprints is defined as follows:

$$\rho(i, j) = \frac{N_c^{ij} - N_d^{ij}}{\binom{l}{2}} \quad (3)$$

Theoretically, $N_c^{ij} + N_d^{ij} = \binom{l}{2}$, $\rho(i, j) \in [-1, 1]$, $\lambda(i, j) \in [0, 1]$. The closer two users are, the larger number of concordant pairs and thereby the larger $\rho(i, j)$ is, and vice versa. Ideally, if users v_i, v_j sense exactly the same WiFi APs with the same ranks, we have $\lambda(i, j) = 1, \rho(i, j) = 1$. We thus define the WiFi similarity between user v_i and v_j by

$$\tau_{ij} = \lambda(i, j) \times \rho(i, j) \quad (4)$$

such that $\tau_{ij} \in [-1, 1]$. However, due to the device diversity, even when two users with different devices stand very closely, the sensed WiFi APs may not always be just the same, making $\tau_{ij} < 1$ in practice. On the other hand, when users v_i, v_j are separated far away such that they sense no common WiFi APs, we define $\tau_{ij} = -1$. Fig. 2 (d) describes the between-user WiFi similarity matrix for 14 users. We clearly observe that for closer pairwise users, the WiFi similarity τ tends to be larger (with the corresponding color approaching yellow), showing that τ is a good indicator of measuring proximity between users.

Note that the WiFi similarity τ is not a dissimilarity metric (e.g., $\tau_{ii} = 1$ instead of $\tau_{ii} = 0$). To make the

CMDs technique applicable for on-the-fly map generation, we further define a τ -induced distance d^τ . Specifically, for user v_i and v_j , the distance d_{ij}^τ between them is defined as

$$d_{ij}^\tau = \sqrt{\tau_{ii} + \tau_{jj} - 2\tau_{ij}} = \sqrt{2 - 2\tau_{ij}} \quad (5)$$

which satisfies $d_{ii}^\tau = 0, d_{ij}^\tau = d_{ji}^\tau$. With this new metric, we can obtain a distance matrix $D^\tau = (d_{ij}^\tau)_{n \times n}$, and then apply the CMDs technique for generating a relative coordinate system (i.e., on-the-fly map) of the location fingerprints. The τ -induced distance is actually not a true distance metric in the walkable space, and thus we have to compute the walkable distance between location fingerprints along corridors for tracking the progress of navigation users. This will be addressed in next section.

3. SYSTEM DESIGN

3.1 An Overview

When a user with a COTS mobile device walks naturally in an indoor space of interests (e.g., a giant shopping mall) and requests the service of on-the-fly map generation for navigation, s/he turns on the system PIN and becomes an active user. The device begins to collect the compass, gyroscope and barometer readings, and WiFi signals from surrounding WiFi APs for a few seconds, and then uploads the sensed data to the cloud server only for once. Upon receiving crowd-sourced sensory data, the server conducts a series of operations, namely, on-the-fly map generation (including between-user WiFi similarity computation, local map generation, local map stitch, edge computation, and level-change detection) and on-the-fly map based indoor navigation (including navigation path computation, progress tracking and deviation detection). Note that in our system, the client (i.e., the user's mobile device) only takes charge of sensing ambient information and displaying navigation instructions, and all computation burdens are shifted to the cloud server. Fig. 3

depicts the system diagram.

As a concrete example of map generation, assume that there are 52 active users walking along the walkable space as shown in Fig. 4(a)². With the uploaded sensory data from users, the server immediately computes the WiFi similarity τ between pairwise users (see Fig. 4(b)) according to Equation (3). One problem here is that, as shown in Fig. 4(c), in the complex environment the CMDS technique cannot correctly recover the topology of location fingerprints [22], when directly using the τ -induced between-user distance matrix of these 52 location prints (according to Equation (5)). To tame this challenge, PIN conducts the following steps. It applies the compass readings and WiFi-similarity to divide the location fingerprints into sub-clusters (see Fig. 4(d)), within each of which a local coordinate system is derived. The local systems represent the relative positions of the location fingerprints (detailed in Section 3.2). Two local systems are then stitched together by identifying whether two sub-clusters are adjacent and thereon determining two adjacent location fingerprints (i.e., turning points) corresponding to the neighboring sub-clusters, and eventually a global relative map is generated (Section 3.3). We then connect two adjacent points of the same sub-cluster and two turning points belonging to neighboring sub-clusters to form an edge, and compute the length for each edge based on the relationship between WiFi similarity and walkable distance as illustrated in Fig. 4(f), and the edge directions based on the compass readings (Section 3.4). In addition, to offer the cross-level navigation in multi-floor buildings, we exploit the barometer readings for level-change detection such that the difference of floor level between neighboring users can be computed (Section 3.5).

With the derived global relative map, the server computes for the user a navigation path to the destination (e.g., the location fingerprint sensed by a friend of the navigation user), and tracks the navigation progress via dead-reckoning based on accelerometer readings. In particular, once the navigation process starts, we periodically collect the WiFi signals and estimate the stride length by dividing the displacement (according to the function of the walkable distance over WiFi similarity) and step count. The WiFi sensing process stops for saving energy when the computed stride length converges. Afterwards, the server computes the displacement based on the estimated stride length and step count, and provide instructions on turning or floor level

²In fact, here we just let one participant walk along the corridors, and the participant manually uploads the WiFi signals and compass readings at 52 different locations during walking, just like 52 active users walking and sensing the 52 location fingerprints. In what follows, we may interchangeably use the terms of user, location fingerprint and point.

change. When a deviation from the course is detected, the server alarms the user immediately and helps him/her back to the course.

3.2 Local Map Generation

For each active user naturally walking in the indoor space of interests, the carried COTS mobile device senses for a few seconds the WiFi RSS from surrounding WiFi APs, barometer, gyroscope and compass readings, and uploads the whole packet to the cloud server. The server then computes WiFi similarity τ and τ -induced distance d^τ between users according to Equation (3) and Equation (5) in Section 2, and applies the CMDS technique for on-the-fly map generation. The active users distributing along the indoor walkable space form a dummy graph with an arbitrarily shaped network. However, the MDS-based techniques suffer from irregular shapes [22] of the dummy graph (as shown in Fig. 4(b)). This is because 1) the Euclidean space does not match with the walkable space in complex indoor environments³, and 2) two users walking on different corridors may have a WiFi similarity of -1, no matter how far away they are. We observe that on the same straight corridor or regularly shaped room, the Euclidean space correlates well with the walkable space, and the Euclidean distance (and thus the walkable distance) between users is nearly proportional to their WiFi similarity. As such, we propose a two-stage clustering process to divide the map into sub-maps with logic meanings (e.g., corridor segments, rooms) for local coordinate computation, and then stitch them into one global map (for simplicity, we assume that the users walk on the same floor, and the case for cross-level users will be discussed later).

To this end, we first divide the users into clusters according to the compass readings. As on the same corridor, the users may walk in opposite directions (i.e., the difference of compass readings is around 180 degree), we subtract 180 degrees from the compass readings over 180 degrees. As a result, the revised compass readings fall within [0,180]. Initially, each user is a cluster. Then, two clusters are merged if the maximal difference of pairwise compass readings is smaller than a given threshold (e.g., 30 degrees similar to [27]). Such a merging process repeats until there is no change of membership. Clearly, within each cluster, the users have similar compass readings with the fluctuation no greater than 30 degrees. This implies that users on parallel corridors belong to the same cluster. To differentiate the users on different parallel corridors, we further divide the users within a cluster into sub-clusters, each of which corresponds to a logic unit (e.g., corridor, room), we exploit the WiFi similarity between users. Specifically, for any

³That is, two users geographically close may not be directly reachable as they might walk along different corridors separated by obstacles (e.g., rooms, walls, etc.).

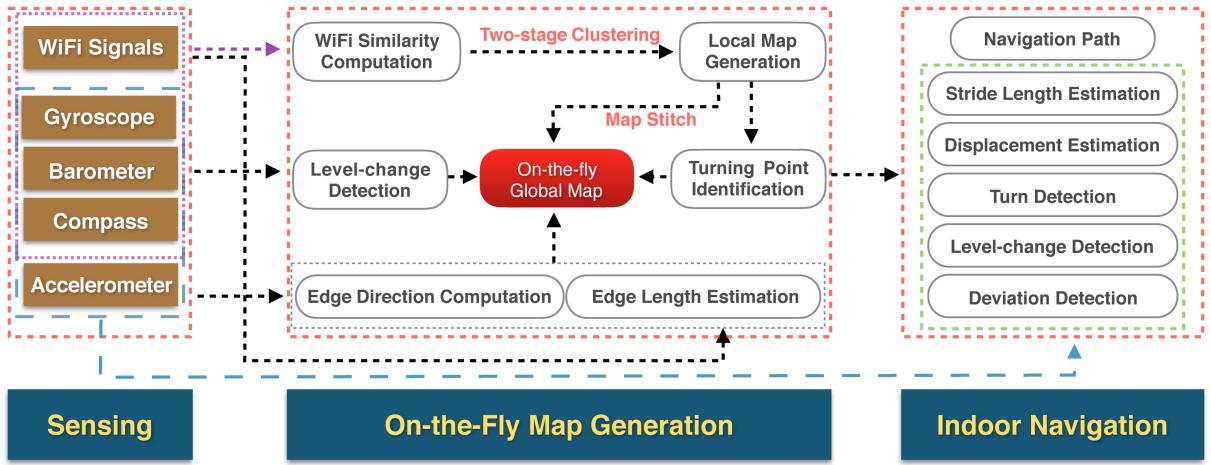


Figure 3: The diagram of PIN system.

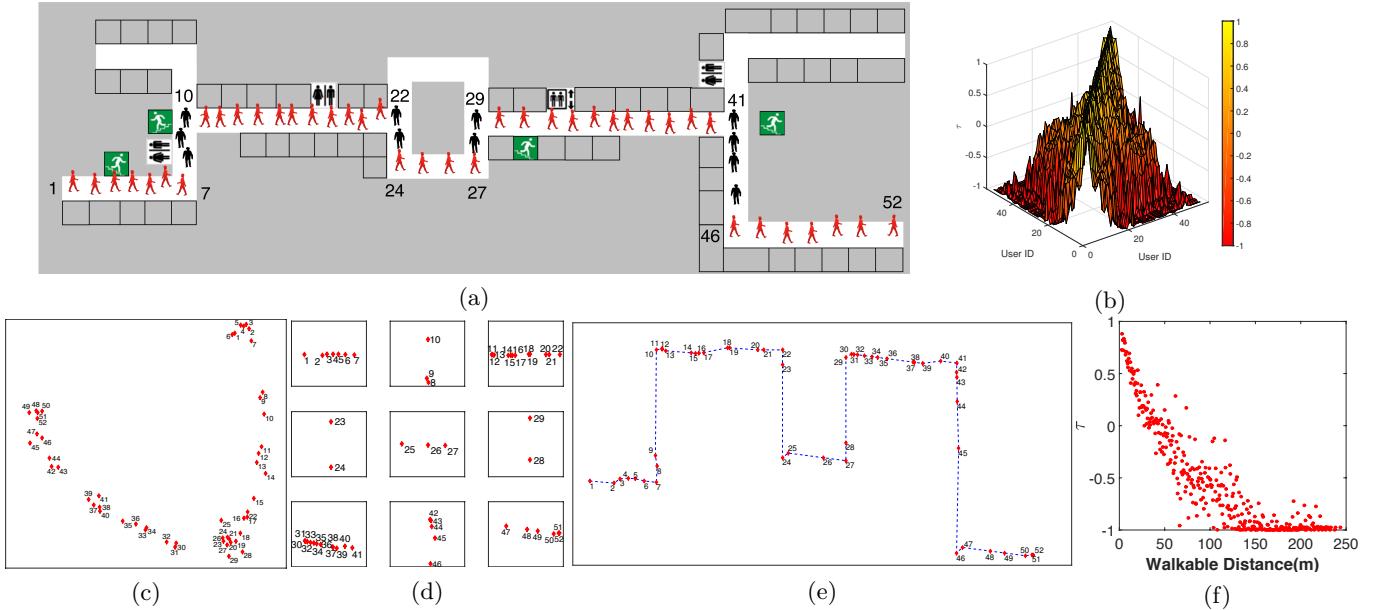


Figure 4: An illustrative example. (a) The 52 active users (marked by the numbers) walk along the corridors in a campus building and upload the sensory data to the server; (b) The between-user WiFi similarity surface where only nearby users have a WiFi similarity larger than 0; (c) The CMDS yields an incorrect user layout if directly using the τ -induced distance matrix based on the WiFi similarity given in (b); (d) The 9 sub-clusters after two-stage clustering process; (e) The global map after stitch; (f) The relationship between walkable distance and WiFi similarity.

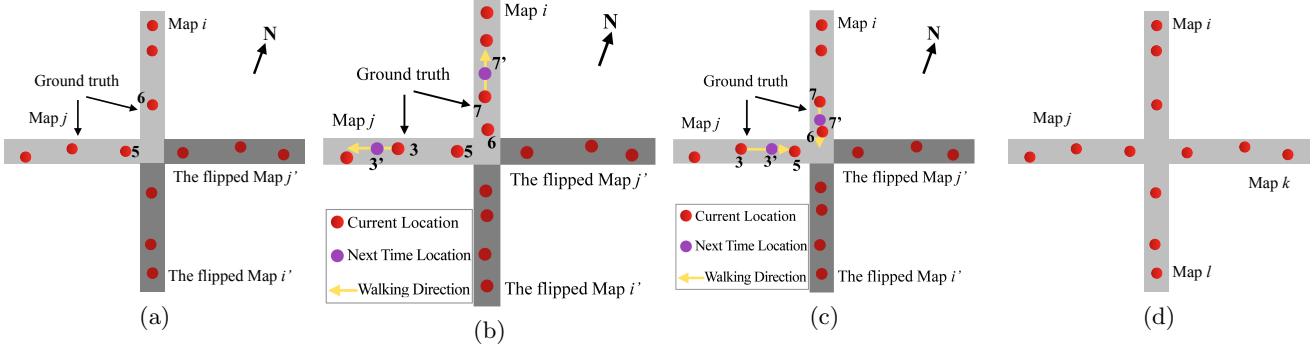


Figure 5: The flipping problem. (a) Map *i* and Map *j* have an incorrect flip symmetric to the corner of the L-shaped corridor; (b) (c) The solution for the flipping problem; (d) The cross-shaped corridors.

two users within a cluster, if their WiFi similarity is larger than a given threshold δ_τ ⁴, then these two users fall in one sub-cluster, otherwise they belong to different sub-clusters. Eventually, a cluster is divided into multiple sub-clusters, and within each sub-cluster, the WiFi similarity between pairwise location fingerprints is larger than δ_τ . Note that a user may collect significantly different compass readings as compared with those by other users on the same corridor, since the compass suffers from the ferromagnetic interference and is thus not very reliable. As a result, this user may become an *orphan* after two-stage clustering process. To clean up the orphan, we simply let it belong to a sub-cluster if the WiFi similarity between the orphan and a user in the sub-cluster is the maximal as compared with other sub-clusters. This solution also works when there are stationary users.

For any sub-cluster, say $C_{sub}(j)$ with n_j users, we compute the distance d^T between pairwise location fingerprints, and apply the CMDS technique for constructing a relative local map $M_{sub}(j)$. Note that the traditional techniques for spectral decomposition of a matrix $B_{n \times n}$ has a complexity of $O(n^3)$. Since here only the top 2 largest eigenvalues and the corresponding eigenvectors are used for local coordinate system generation, we apply the power method (a.k.a power iteration) with $O(n^2)$ complexity [22] on the distance matrix D^j , considering the scalability issue. The power method of the matrix [9] is designed for extracting the dominant eigenvalue (i.e., the largest eigenvalue) and the corresponding eigenvector. We thus repeat the power method twice to derive the top 2 largest eigenvalues and eigenvectors. Fig. 4 (d) gives the nine sub-clusters after the two-stage clustering process where each sub-cluster corresponds to a corridor (segment).

3.3 Local Map Stitch

⁴In our experiments, we set it to be 0.2 since the preliminary evidences show that the users with WiFi similarity larger than 0.2 are generally on the same corridor.

Now we stitch these local maps to form a global map based on the adjacency relationship between pairwise sub-clusters. For two maps $M_{sub}(i)$ and $M_{sub}(j)$, we compute the WiFi similarity between them as

$$\tau_{sub}(i, j) = \max_{i_1 \in M_{sub}(i), j_1 \in M_{sub}(j)} \tau_{i_1 j_1} \quad (6)$$

If $\tau_{sub}(i, j) > 0$ ⁵, these two local maps are identified as neighbors, and the points in the two maps, which correspond to the pair of users with WiFi similarity $\tau_{sub}(i, j)$, are called as *turning points*. For instance, in Fig. 4(d), the sub-cluster formed by users numbered from 1 to 7 is a neighbor of the sub-cluster formed by User 8, User 9 and User 10. User 7 and User 8 are two users with the maximal WiFi similarity, and accordingly these two points are identified as turning points.

After identifying the turning points, next we will stitch pairwise neighboring sub-clusters. This is non-trivial since we have no anchor nodes and the floor map is not handy. Even though the crowdsourcing based approaches have been proposed, e.g., in [10, 13], they requires efforts from participants and are costly to keep the floor map up-to-date. As such, there exists an incorrect flip. Please see Fig. 5 for some intuitive. With the compass readings and adjacency relationship, the relative position of Map *i* and Map *j* is not uniquely determined, and there is a flipped map for each map. If one turning point, say Point 5, is detected to make a left turn based on the gyroscope data within a few seconds, then we can infer that Map *j* is on the left side of Map *i*. If no turning action is detected, then we use the compass readings and WiFi information at time $t_0 = 0$ and $t_1 = 1$ (i.e., a few seconds later). As an example of an L-shaped turn, for the turning point of Point 5 in Map *i* and Point 6 in Map *j*, we select a point (e.g., Point 3) on Map *i* and a point (e.g., Point 7) on Map *j*, such that Point 3 (or Point 7) is several meters away (e.g., the WiFi similarity between them is less than 0.8 ac-

⁵If two corridors are not adjacent, their WiFi similarity tends to be less than 0 as there are obstacles between them, which is verified by our experiments as in Fig. 2.

cording to the linear function of walkable distance over WiFi similarity) from Point 5 (or Point 6). As shown in Fig. 5 (b), if $\tau_{35} < \tau_{3'5}$, showing that User 3 is walking away from the previous location of User 7, we have the compass readings of User 3 as $A'(3) = A(3)$ and if $\tau_{76} < \tau_{7'6}$, $A'(7) = A(7)$. On the contrary, as shown in Fig. 5 (c), if $\tau_{35} > \tau_{3'5}$, $A'(3) = 360 - A(3)$, and if $\tau_{76} > \tau_{7'6}$, $A'(7) = 360 - A(7)$. If we regard the intersection of Map i and Map j as the center of the compass, according to the revised compass readings $A'(3)$ and $A'(7)$, we can easily deduce the relative position of Map i and Map j . For instance, if $A'(7) = 0$, $A'(3) = 90$, then Map j is on the left side of Map i . In the similar way, we can determine the relative position of Maps associated with a T-shaped intersection or a cross-section as in Fig. 5 (d). Fig. 4 (e) shows the merged map of 52 users.

3.4 Edge Computation

The generated global map consists of a point cloud to be connected into a graph for indoor navigation. To that end, we should determine whether two points form an edge such that the active users can go directly from one end point to the other.

As each point is aware of its sub-cluster, for each non-turning point v_i , we regard a point v_j in the same sub-cluster of v_i as a neighbor of v_i such that τ_{ij} is the maximal as compared with other points in the sub-cluster. For a turning point v_i , its paired turning point in the neighboring sub-cluster is also a neighbor of v_i . When the users in a sub-cluster is not densely distributed, such a strategy may result in a disconnected graph even in the same sub-cluster. In this case, we further connect these isolated components to form a connected graph, such that for any point in a sub-cluster, there is a path to any other points in the same sub-cluster. Two points which are directly connected form an edge.

After determining edges between pairwise location fingerprints, we compute the walkable distance between them. From Fig. 4 (f) we observe that when the users are nearby (e.g., on the same short corridor), there is a strong linear relationship between the WiFi similarity and walkable distance. As such, we can build a linear model of walkable distance $d_{ij} = f(\tau_{ij})$ on between-user WiFi similarity τ_{ij} , and then estimate the edge length according to the WiFi fingerprints collected by the two users. Note that the linear model varies with the indoor environment where the sensory data are collected. Please see Fig. 6 for some intuitive. With the training data collected in the office building in Fig. 6 (a) we have $d_{ij} = -29.501\tau_{ij} + 28.353$, and $d_{ij} = -30.592\tau_{ij} + 30.289$ based on the training data of the shopping mall in Fig. 6 (b). However, from the CDF of ranging errors (i.e., the difference between the estimated edge lengths and the true distance between pairwise users) in different envi-

ronments with different models, we find that the environment has little effect on the ranging error, and about 80% of ranging errors are less than 5 meters by applying both models in the two tested scenarios.

The edge direction can be inferred according to the compass readings. As the compass readings vary even for users walking in the same corridor, to avoid yielding a *squiggly* graph where the edge directions for users in the same straight corridor are different, we define the directions of edges on the same sub-cluster as the median compass readings of the users on the sub-cluster. The angle of the turning edge connecting two turning points is defined as the difference of the edge directions in the two neighboring sub-clusters.

3.5 Level-change Detection

So far we have established a global relative map of a 2-dimensional space based on the crowdsourced data from the active users. Note that the users may be at different floor levels. Fortunately, to offer a cross-level navigation service, we only need to determine the level-change information from the starting point (i.e., the navigation user) to the destination (i.e., the targeted person); the specific floor level of each user/point is not necessary (and also difficult) to be identified. Both accelerometer and barometer data can be used for detecting whether a user walks on the same floor or not, but barometer is often more robust when detecting vertical activities as the barometer readings are independent of the orientation of devices and user activities (e.g., stationary or motion) [23]. It is commonly accepted that, 1) absolute pressure value is an unreliable indicator for floor-level information, as it suffers from such factors as device diversity, room temperature, weather patterns and humidity, etc., and 2) pressure difference is useful only for barometer readings by the same device. For instance, from Fig. 7(a), we observe that even on the same floor, the absolute pressure values by Huawei Mate 8 are always 1.5 mBar higher than those by Mi Note, and the pressure difference between the barometer data by Huawei Mate 8 and Mi Note is far higher than changing one floor by the same device, which is often below 0.7 mBar as the floor heights typically range from 2.5 to 6 meters. In other words, we can not judge whether two users are on the same floor simply based on the pressure difference of their barometer data.

To address these challenges, we exploit the barometer data sequence $\{p_1, p_2, \dots, p_n\}$ uploaded by users walking for a few seconds (e.g., 3 seconds), as shown in Figs. 7(a)-(c). Clearly we can see that the pressure difference $\Delta(p) = p_n - p_1$ by the same device varies greatly among different walking patterns, such as remaining on the same floor to changing floor by stairs, escalators and elevators. In general, for the case of floor unchanged, $\Delta(p)$ is the smallest, and the $\Delta(p)$

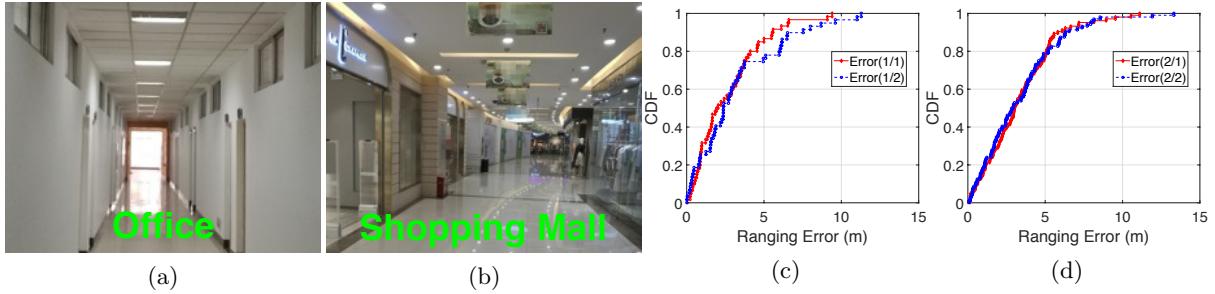


Figure 6: The ranging error distribution on different scenarios with different models. (a) Scenario 1: an office building; (b) Scenario 2: a giant shopping mall; (c) The cdf of ranging errors for testing data collected in Scenario 1 based on training data in Scenario 1 (i.e., Error(1/1)) and Scenario 2 (i.e., Error(1/2)), respectively; (d) The cdf of ranging errors for testing data of Scenario 2 based on models with training data in Scenario 1 (i.e., Error(2/1)) and Scenario 2 (i.e., Error(2/2)), respectively.

for taking elevators is the largest. This shows that it is possible to detect a user's walking pattern by setting a threshold $\delta(p)$ on the pressure difference of the sequence. Generally speaking, the barometer readings via taking elevators/escalators are significantly different with those when people walking in the same floor. However, we find from Fig. 7 (d) that the threshold-based approach does not always succeed in differentiating whether a user walks via stairs or remains on the same floor. For example, if $\delta(p) = -0.05$, a user may go upstairs or stay on the same floor. We thus refer to this approach as *Naive Approach*. We observe that when users go down or up via stairs, elevators or escalators, the barometer readings are monotonically increasing or decreasing (i.e., there is a monotonic trend), while the barometer readings change randomly without any trend when users walk on the same floor level. As such, to statistically infer the floor change information, we further apply a simple nonparametric testing approach, named *Mann Kendall Trend Test* (or simply M-K test) [25], on the barometer data. Specifically, we state the following claim for the null and alternative hypotheses of a two-sided test:

H_0 : The barometer readings have no trends $\leftrightarrow H_1$: The barometer readings follow a monotonic trend.

The statistic for M-K test is given as

$$S = \sum_{k=1}^{n-1} \sum_{j=k+1}^n \text{sign}(p_j - p_k) \quad (7)$$

where

$$\text{sign}(p_j - p_k) = \begin{cases} 1, & p_j > p_k \\ 0, & p_j = p_k \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

The variance of S is given as

$$\text{Var}(S) = \frac{n(n-1)(2n+5) - \sum_{k=1}^c t_p(t_p-1)(2t_p+5)}{18} \quad (9)$$

where c denotes the number of tied groups with equal values and $t_p (p = 1, 2, \dots, c)$ is the number of observations in the p -th group⁶. A large and positive (or small and negative) S often indicates that there is an increasing (or a decreasing) trend, and otherwise there is no monotonic trend. And its normalized statistic Z_{MK} can be derived as follows.

$$Z_{MK} = \begin{cases} \frac{S-1}{\sqrt{\text{Var}(S)}}, & S > 0 \\ 0, & S = 0 \\ \frac{S+1}{\sqrt{\text{Var}(S)}}, & \text{otherwise} \end{cases} \quad (10)$$

When the null hypothesis H_0 holds, which implies that the user remains on the same floor, the statistic Z_{MK} should follow the standard normal distribution $N(0, 1)$. Given a positive constant $\alpha \in (0, 1)$, the so-called *significant level*, and an observation z of Z_{MK} , if the p-value $p(|Z_{MK}| < |z|) < \alpha$ (e.g., $\alpha = 0.05$), we will reject H_0 with at most Type I error rate of α , and thus infer that the barometer readings have a monotonically increasing or decreasing trend, depending on $z > 0$ or $z < 0$. Or, given a threshold $z_{\frac{\alpha}{2}}$, the $100(1-\alpha)$ th percentile of $N(0, 1)$, if $|z| > z_{\frac{\alpha}{2}}$ (e.g., $z_{0.025} = 1.945$), we reject H_0 at the significant level of α such that there is a monotonic trend of barometer data: if $z > z_{\frac{\alpha}{2}}$, there is a monotonically increasing trend, and if $z < -z_{\frac{\alpha}{2}}$, there is a monotonically decreasing trend. Otherwise, we do not reject the null hypothesis, indicating the user remains on the same floor. Table 1 gives some results

⁶For example, if a data sequence is '1, 2, 2, 3, 4, 4, 4, 5, 5, 6', then there are ($c = 3$) tied groups, namely, '2, 2', '4, 4, 4', and '5, 5', such that $t_1 = 2, t_2 = 3, t_3 = 2$.

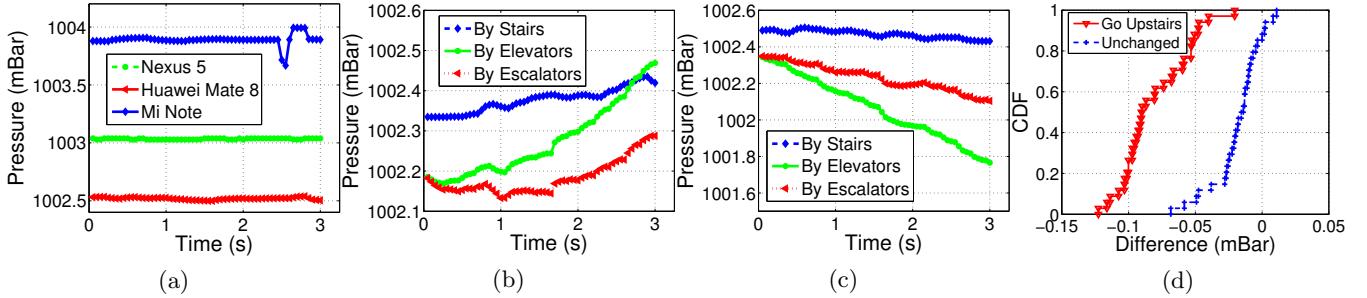


Figure 7: The curves of barometer readings over time (a) by three different devices on the same floor, (b) by taking different tools for going upstairs or (c) downstairs with Huawei Mate 8; (d) the CDF of pressure difference for going upstairs and staying on the same floor (i.e., the floor level is unchanged).

Table 1: The M-K test results on multiple barometer readings by Huawei Mate8.

Floor level info	S	Var(S)	Z_{MK}	p-value	Result
Remaining at Floor 12	30	14223.33	0.2	0.80788	Accept H_0
Remaining at Floor 13	116	14246.67	1	0.33531	Accept H_0
Going Up / Elevator	591	14060.33	5	$6.5012e^{-7}$	Reject H_0
Going Down / Elevator	-316	14092.673	-2.7	0.00797	Reject H_0
Going Up / Escalator	917	14144.33	7.7	$1.3323e^{-14}$	Reject H_0
Going Down / Escalator	400	14237.33	3.3	0.000826	Reject H_0

of M-K test on different scenarios.

By incorporating the Naive Approach and the trend test, we can determine whether a user walks on the same floor, or go up or down via stairs, elevators or escalators. Specifically, when there is a trend and $|\Delta(p)| < |\delta(p)|$, the floor level remains unchanged; otherwise, the user goes up or down, depending on the sign of $\Delta(p)$. In our experiments, we empirically set the threshold $\delta(p)$ for taking stairs, escalators and elevators to be 0.04, 0.1, and 0.2, respectively. We call the user changing the floor level as a *cross-level user*. With these cross-level users, we can easily determine whether a user has a higher or lower floor level than other users. Specifically, for two neighboring users v_i and v_j (i.e., connected by an edge), if they and their neighbors are all not cross-level users, then we infer that v_i and v_j are on the same floor. If v_i and v_j are not cross-level users, but they have at least one neighboring cross-level user v_k with decreasing (or increasing) trend of barometer data, such that τ_{ik} increases while τ_{jk} decreases, then v_i is one higher (or lower) floor than v_j . The maps corresponding to different floors are connected via the cross-level users, and eventually, we can derive a multi-floor on-the-fly map with between-user floor level difference, which is helpful for navigating users across different floors. To show the effectiveness of our approach, we conduct experiments and compare the performance of Naive Approach and our approach on going upstairs and remaining unchanged. Figs. 8(a) and (b) depict the detection accu-

racies of going upstairs, remaining unchanged and the overall accuracy, under different threshold values. The barometer readings are sensed and collected for 3 seconds. It can be seen that for any threshold value $\delta(p)$, our approach is more accurate than Naive Approach. For instance, if $\delta(p) = 0.04$, our approach can detect the case for remaining unchanged with 100% accuracy, higher than Naive Approach of 84% accuracy, while the accuracies of detecting going upstairs are comparable, both over 96%. Figs. 8(c) and (d) illustrate the comparison results when the participant walks on the same floor and the sensing duration is 1 second and 2 seconds, respectively. Clearly we observe that our approach outperforms Naive Approach for all threshold values. In addition, from Figs. 8(a), (b) and (c) we observe that the overall accuracy for sensing duration of 1 second and 2 seconds is lower than the accuracy for sensing duration of 3 seconds.

3.6 Plug-and-Play Indoor Navigation with On-the-Fly map

Having the generated on-the-fly map, each user is aware of its current point in the map. Given a destination point (e.g., corresponding to Alice's friend Bob) on the map, PIN computes the shortest path from the user to the destination, which consists of weighted and directed edges, a set of turning points and/or cross-level points (these two kinds of points are referred to as *critical points*) and walkable distance between adjacent crit-

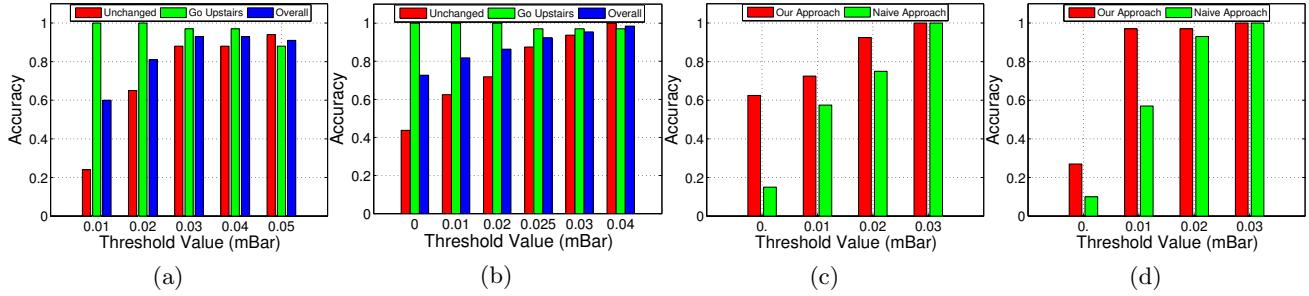


Figure 8: Detection accuracy of floor level change when going upstairs or remaining the same floor. (a) The naive approach only and (b) our approach combining the naive approach with the trend test. The sensing duration for the barometer is 3 seconds; (c) (d) The comparison results of our approach and naive approach on detecting users without changing floor level where the sensing durations are one second and two seconds, respectively.

ical points. Based on the navigation path, PIN displays the turn-by-turn instructions on the user's smartphone to the destination, and apply dead-reckoning method for user tracking.

To compute the user's displacement for dead-reckoning, the traditional ways are to multiply the step count with stride length which however requires user input (i.e., height, weight), or apply the double integral on accelerometer readings which suffers from accumulated errors. Another way is to let the user periodically upload the WiFi information, and then compute the WiFi similarity between two consecutive moments for estimating the displacement. However, this is an energy-expensive solution of periodically sensing WiFi information [27]. Instead, in our system, we propose an energy-efficient approach while avoiding any input from users. Specifically, when a user requests the navigation service, the device will upload the WiFi information immediately, and then after the user walks for a few (e.g., 10) steps, the device will once again upload the WiFi information. The displacement for walking every 10 steps is computed based on WiFi similarity of these two WiFi fingerprints. By averaging the displacement over the step count we can estimate the user's stride length. As there can be fluctuation of WiFi signals, the estimated displacement may not be very accurate. We thus repeat such a process until the stride length converges, i.e., the difference of two consecutive stride lengths is smaller than a given threshold (e.g., 0.1 meter).

With the estimated stride length, PIN tracks the navigation progress by simply counting the walking steps, and prompt for users the instructions on turning or level-change (e.g., going up/down for 2 floors) based on the computed distance to the next critical point on the navigation path. With the walking displacement and instructions, PIN detects whether the user deviates from the path. Specifically, if the user reaches a point nearest to a cross-level point on the path, PIN uses the barometer readings to determine whether the

user is correctly going downstairs or upstairs. If the user reaches a turning point, after walking a distance equal to the turning edge length, the gyroscope readings are used for detecting whether there is a correct turning action. If the user is detected to veer off the course, PIN alerts him/her back to the course until s/he reaches the destination.

4. PERFORMANCE EVALUATION

In this section, we will present the evaluation of PIN system. To run our navigation system, we rent the cloud server of a world-famous cloud server provider, with the basic configuration of 1-core 2GB, 2Mbps, and storage size of 50G. We conduct extensive experiments in a 3-story campus building where the testing area of each floor is around $2400m^2$ and there are enclosed corridors on Floor 2 and Floor 3. To generate the map, we have 18 volunteers with Android smartphones at hand. On each floor, a part of volunteers first upload the sensory data (including WiFi signals, barometer, gyroscope and compass readings sensed for 3 seconds) while walking from the initial locations along a given corridor, and then go to another corridor for the sensing and uploading task. Doing so we can generate, though offline, a map for each floor, just like there are sufficient users⁷. To merge the maps of the three floors, we let 10 volunteers go up or down via stairs or elevators and records the WiFi information and the barometer data for 3 seconds. PIN determines the floor-level change information and merge the maps based on the identified cross-level point generated by the 10 participants. For the performance evaluation of the map based navigation, we randomly select 100 pairs of source-destination in the map, and accordingly compute the navigation path with turn-by-

⁷We do not hire enough participants with Android smartphones to walk along the major corridors in the complex 3-story building for map generation. However, this can actually show that our system do not require the user to upload the sensory data simultaneously.

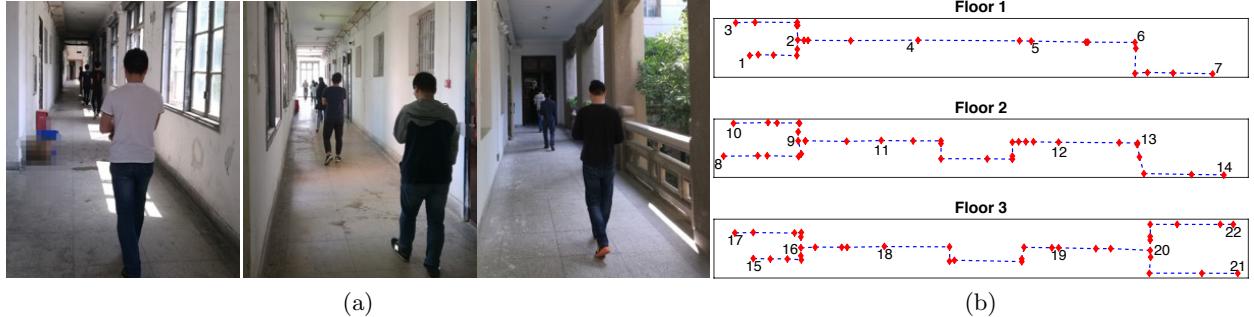


Figure 9: The evaluation scenario in a three-story campus building. (a) There are 18 participants walking naturally along the corridors. Within each floor, we let each participant upload sensory data first on one corridor and then go to another corridor for sensing and uploading. We also let some participants go up or down by elevators or stairs for floor-level change detection; (b) The generated map where the numbered points correspond to an elevator or stairs.

turn instructions. The navigation progress is periodically tracked by computing displacement based on the estimated stride length, and detecting deviation from the course according to the gyroscope and barometer readings. Fig. 9 shows the experiment settings and the generated map for the 3-story campus building.

4.1 The Performance of On-the-fly Map Generation

In this paper, the on-the-fly map is of crucial importance to the plug-and-play indoor navigation. We thus first evaluate the performance of map generation, including the effect of user density and distribution, the robustness of WiFi similarity, turning point detection accuracy, edge direction accuracy, and between-user distance error.

The density and distribution of users walking indoors affects greatly the accuracy of the generated map, and more users widely distributed along the walkable space will offer a more accurate map. However, since our system aims at providing turn-by-turn instructions, it works as long as there are users such that the generated map covers the corridors along the way to the destination and thus is capable of offering a connected navigation path. We randomly select the location fingerprints in Fig. 4, and Fig. 10 illustrates the generated maps for four different distribution of active users. We observe that as long as there are at least two users walking on each corridor, the generated maps will correctly reflect the shape of the underlying walkable space. Also we notice that if there is no user near a true turning point, the generated map will incur an inaccurate turning point, e.g., User 6 in Fig. 10(d) is several meters away from the corridor corner, but overall the shape has no difference with other three maps of Figs. 10(a) to (c). That is, they all can provide accurate turn-by-turn instructions, though the provided instructions based on the map in Fig. 10(d) may be not very accurate.

To show the robustness of WiFi similarity, one month after the map in Fig. 4(e) is derived, we let a participant upload the sensed WiFi information at the same locations where the previous 52 users collected sensory data, respectively. We then compute the WiFi similarity between each current WiFi signature and all previous 52 WiFi fingerprints. The current location is thus inferred based on the principle of the maximal WiFi similarity. That is, if the WiFi similarity is with previous fingerprints $i (= 1, 2, \dots, 52)$, then we infer that current user is at the location of previous user i . The results are shown in Fig. 11. We can see that 36 current locations are inferred accurately, and the 80-percentile distance error (i.e., the distance between the true location and the estimated location) is less than 5 meters, which is often sufficient for turn-by-turn indoor navigation. This also implies that when a new user requires the navigation service, we can easily associate the user with a point (or location fingerprint) in the generated map with the maximal WiFi similarity between the location fingerprint and the new WiFi signals, without any need for map reconstruction.

Fig. 12 describes the accuracy of turning point identification between two neighboring corridors. It is defined as the ratio of the number of accurately identified turning points to the number of all turning points on a navigation path. Due to the fluctuation of WiFi signals, it is possible that two points of pairwise neighboring corridors, even though they are nearest to a corridor corner or intersection, may not be identified as the turning points as their WiFi similarity is not the maximal among other pairwise points. However, our approach can still identify the turning points with high accuracy, where around 80% of navigation paths has an accuracy over 95%. At the same time, for the mistakenly identified turning points, we find that they are not very far from the ground truth.

Fig. 13 presents the CDF of turning angle errors. As

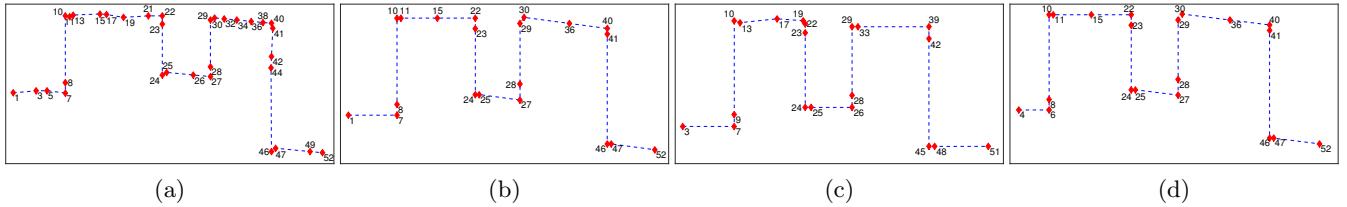


Figure 10: The effect of user densities on map generation.

we only compute the difference of compass readings at the turning points, the turning angle does not suffer from error accumulation, which is verified by our experiments where the 80-percentile error is 20 degrees and the median error is 15 degrees around. This can guarantee the accuracy of the turning instructions.

Note that we apply the regression model about WiFi similarity to estimate the distance between location fingerprints. Fig. 14 shows the CDF of ranging errors (i.e., the error of the true distance and the estimated distance between pairwise location fingerprints). Our approach can achieve a high accuracy, with the 80-percentile error of 5 meters and the maximal error of 12 meters around. In particular, we find that the large ranging error is mainly due to the fact that two location fingerprints are separated far away, and for close fingerprints, their estimated distance is very close to the true distance. This explains why PIN tries to connect close location fingerprints into an edge during map generation.

4.2 The Performance of Plug-and-Play Indoor Navigation

Next we show how our indoor navigation system performs in terms of displacement error, deviation detection delay, and lead time at several checkpoints.

An important function of navigation is to track the navigation users, such that when users deviate from the course, the system should make a prompt alert. PIN system first exploits the WiFi similarity (e.g., sensed at time t_0 and time t_1 after a few steps) to estimate the stride length. Fig. 15 shows that after a few steps, the stride length curves of two users with the true stride length being 50cm per step (50cm/s for short) and 60cm per step (60cm/s for short) respectively. The estimated stride lengths after walking 15 steps are 53cm/s and 64/s respectively. With the estimated stride length, we keep tracking the navigation user by dead-reckoning, and the tracking accuracy (i.e., the displacement error) is shown in Fig. 16. As expected, our approach can yield a promising result, with the maximal error being 1.5 meters.

Fig. 17 illustrates the CDF of deviation detection delay during navigation. According to the generated map and the computed navigation path, the users are instructed to make a left/right turn or go up/down. Once there is a deviation, e.g., the barometer readings indi-

cate that the user is going down (or straight) while s/he is supposed to go up (or make a left/right turn), PIN will alert the user to back to the course. With the gyroscope and barometer data, the deviation can be detected in time, with the median delay of 3 seconds.

Finally we consider the navigation performance in terms of the timeliness of navigation instructions, e.g., on making a left/right turn or going up/down. Fig. 18 plots the lead time of instructions at some checkpoints (corresponding to the crossing and staircases in the tested building) as indicated by the numbers in Fig. 9(b). As there can be multiple checkpoints on a navigation path, here of our primary concern is the lead time at the first check point. This is because once a navigation user passes by the first checkpoint, the coming progress can be easily tracked according to the walking displacement and the distance between adjacent turning points after each checkpoint. In this sense the checkpoint can be called as a *reset point* as it can help reduce the dead reckoning error. We find from the figure that 80-percent of instructions is offered at least 5 seconds (from Fig. 9(b)) we observe that there is only about 20-percent of lead times is less than 5 seconds) ahead of the action, and the median is 10 seconds. Of course the lead time is closely related to distance from the starting location to the checkpoint, and the closer the shorter of lead time. That is why some navigation instructions are provided less than 5 seconds before the action. But overall PIN can timely prompt the action at these checkpoints.

5. DISCUSSIONS

5.1 The Impact of User Density and User Distribution

To make PIN work properly, the active users should spread widely in the indoor space to ensure a connected global map, violating this will lead to the failed navigation. Even though more users contribute to generate a more accurate map, it does not mean that PIN only works in the user populated scenarios. As verified by our experiments in Section 4.1, if there are at least two users at each corridor segment along the way to the destination, the system PIN can still work as well. If the user distribution is skewed instead of being uniform, i.e., some corridors have many users while some other corridors have no users, the derived map may be

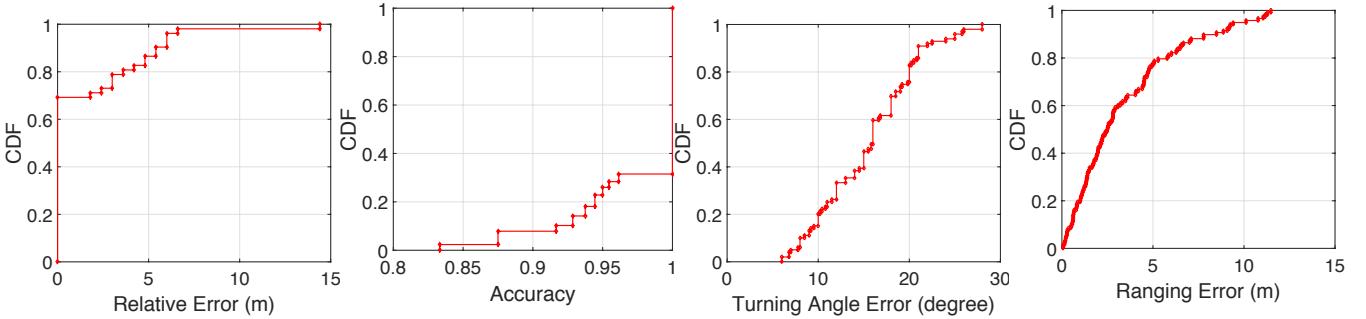


Figure 11: The robustness of WiFi similarity.

Figure 12: The accuracy of turning point identification.

Figure 13: The CDF of turning angle errors.

Figure 14: The CDF of between-user distance error.

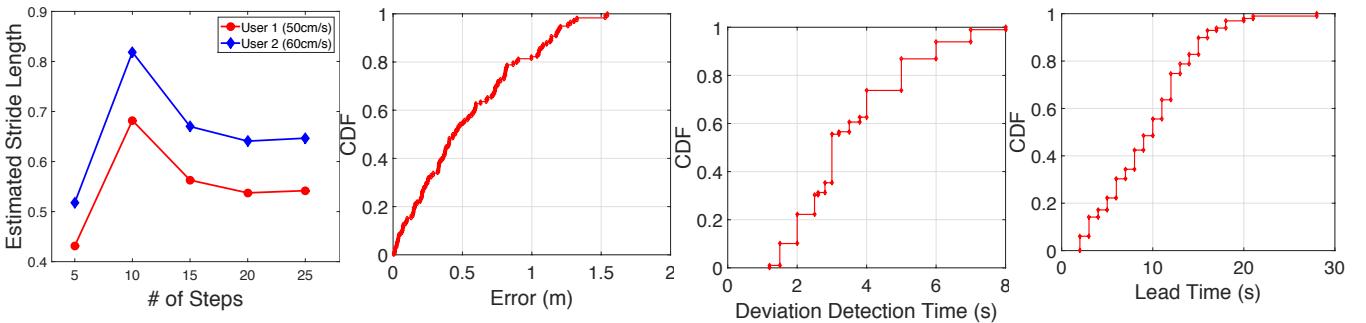


Figure 15: The stride length estimation for two users.

Figure 16: The stride length based displacement error.

Figure 17: The deviation detection delay.

Figure 18: The lead time of instructions at checkpoints.

disconnected as there exists at least pairwise corridors which can not be connected. To tame this problem, PIN can delay the map generation by letting the users keep walking for a while (e.g., 10 seconds) and periodically upload sensory data to the server. This way, we can collect multiple packages (instead of one package as usual) from one single user, where each package corresponds to a location fingerprint (i.e., a point in the map), until the derived graph is connected to guarantee a path between the user and the destination. We leave such a problem of delay-tolerant plug-and-play indoor navigation for future work.

5.2 The Performance in “Sensor Deserts”

The performance of PIN highly depends on the WiFi signals uploaded by the active users. The WiFi signals are used to determine the relationship between users, and instantly derives a global map for plug-and-play navigation. Even though it is rather uncommon that there has none of WiFi APs indoors, occasionally there can be areas uncovered by deployed stable WiFi APs, e.g., in underground parking lots. In such cases, if there are many users with shared hotspots from the smartphones, we may also derive a global map for navigation, though the time-delay mechanism may also be introduced.

5.3 The General Case of Active Users

So far we have assumed that the active users walk naturally along the walkable space, and the compass readings can reflect the walking direction. In practice the real situation may deviate from this assumption. For example, an active user may stand still (e.g., in conversation with friends or having a telephone call) such that the compass readings can be arbitrary ranging from 0 to 360 degrees. Also when the mobile devices are put in bags, instead of being held in hands, the compass readings do not accurately indicate the walking directions of users. To address these issues, we can exploit the accelerometer data to infer whether a user is walking or stationary. For a stationary user, the compass readings are meaningless, and s/he will be an orphan after the two-stage clustering process in Section 3.2. We can affiliate such an orphan with the sub-cluster such that there exists a user having the maximal WiFi similarity with the orphan. If the users are walking, we can obtain the device orientation according to the existing work, e.g., [42], and then determine the walking direction based on the orientation and compass readings.

6 RELATED WORK

With the growing demand for location-based services and the deep penetration of smart devices with built-in

sensors, indoor navigation have attracted a lot of attention from both academia and industry. Roughly speaking, there can be three categories: landmark-based navigation, floor map and location system based navigation, and guider-follower based navigation.

6.1 Landmark-based Indoor Navigation

Often in a giant indoor space (e.g., a big hotel, a convention and exhibition center with activities, etc.) people can find direction boards at the intended junctions to give hints for next movement toward the destination. Under the help of the direction boards, people will go straight after determining the direction for next movement, until they find another direction board asking them to change the direction. We also refer to such an approach as turn-by-turn navigation. This approach is simple, and works when users have no devices at hand. It, however, requires efforts from the users to read the direction boards, and most importantly, is impractical for many indoor environments without the deployment of direction boards. Nowadays, in many indoor spaces (e.g., airports, shopping malls) there are many beacons or landmarks deployed for indoor navigation [7,29]. The beacons and landmarks are used to locate the users and direct the users toward the destination, where the floor map is also used for inferring the current location of navigation users.

6.2 Floor Map and Location System based Indoor Navigation

Currently, most of the indoor navigation systems rely on the floor map and localization system. A handful of studies have been proposed on indoor localization have been proposed, including the WiFi fingerprinting based approach (e.g., [1, 4, 5, 11, 30, 34, 35, 38, 39, 40, 42]), the built-in Inertial Measurement Unit (IMU) sensors based approach (e.g., [12, 20, 24]), visible light based approach (e.g., [14, 15, 18, 21, 26, 32, 36, 36, 37]), and so on. These indoor localization systems rely on the floor map, which are used to locate the users and provide a global reference frame (and a global picture) from the current location to the destination. Given that the floor map is costly to obtain, many researchers [3,13] propose to construct the floor plan in a crowdsourcing fashion, e.g., by using the camera and the built-in inertia sensors. Clearly, these systems require extensive efforts from the participants. Also the derived floor maps are often coarse-grained and difficult to keep up-to-date, and current localization systems still provide unsatisfactory results, making this kind of systems hardly to be widely deployed indoors.

6.3 Guider-Follower Based Indoor Navigation

The guider-follower based navigation [28,41] is a new paradigm without relying on the floor map. This ap-

proach first builds reference paths by some participants, and when a user asks for navigation service, the system downloads the reference path with the same starting location and destination. Zheng *et. al* [41] present Travani, a vision-guided navigation system which collects pathway images, WiFi fingerprints and IMU sensors, to generate a reference trace. The navigation users are then guided by comparing current sensor data with the reference trace. In FollowMe [28], a reference trace is generated only based on a leader's record of sensory data along a specific trip and her/his walking patterns, with no need for taking images or WiFi AP information. However, the major drawbacks are that this paradigm requires pre-deployed reference paths and at the same time, the navigation user should have the same starting location and destination with one of the reference traces.

7. CONCLUSIONS

In this paper, we proposed a plug-and-play indoor navigation system called PIN under the help of online crowdsourcing by active users with commercial off-the-shelf (COTS) mobile devices. The core idea is to treat the active users carrying mobile devices as a sample of the underlying indoor space, on top of which a weighted and directed graph is generated where each user corresponds to a point, and two users nearby form an edge with a real-valued length and a direction. As such, each active user uploads the sensed real-time WiFi information from the surrounding WiFi APs with the built-in sensors of the device. The server generates a global map through a series of operations including robust distance computation, local map generation and map stitch, level-change detection, etc. On top of on-the-fly map, a navigation path is computed and displayed on the user's device. We implement the prototype of PIN and the experiments show that our system can quickly generate a navigation path with a guaranteed successful rate, desirable lead time and deviation detection delay.

8. REFERENCES

- [1] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2000.
- [2] I. Borg and P. Groenen. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.
- [3] S. Chen, M. Li, K. Ren, and C. Qiao. Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos. In *Proceedings of IEEE ICDCS*, 2015.
- [4] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *Proceedings of the 10th International Conference*

- on Mobile Systems, Applications, and Services*, 2012.
- [5] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, 2010.
 - [6] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman. Indoor location sensing using geo-magnetism. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, 2011.
 - [7] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see bob?: Human localization using mobile phones. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, 2010.
 - [8] I. Constandache, R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2010.
 - [9] J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
 - [10] R. Gao, M. Zhao, T. Ye, F. Ye, G. Luo, Y. Wang, K. Bian, T. Wang, and X. Li. Multi-story indoor floor plan reconstruction via mobile crowdsensing. *IEEE Transactions on Mobile Computing*, 15(6):1427–1442, 2016.
 - [11] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, 2004.
 - [12] R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys Tutorials*, 15(3):1281–1293, 2013.
 - [13] Y. He, J. Liang, and Y. Liu. Pervasive floorplan generation based on only inertial sensing: Feasibility, design, and implementation. *IEEE Journal on Selected Areas in Communications*, 35(4):1–9, 2017.
 - [14] P. Hu, L. Li, C. Peng, G. Shen, and F. Zhao. Pharos: Enable physical analytics through visible light based indoor localization. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, 2013.
 - [15] A. Jimenez, F. Zampella, and F. Seco. Light-matching: A new signal of opportunity for pedestrian indoor navigation. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013.
 - [16] J. Koo and H. Cha. Unsupervised locating of wifi access points using smartphones. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1341–1353, 2012.
 - [17] J. Krumm and K. Hinckley. The nearme wireless proximity server. In *Proceedings of UBICOMP*, 2004.
 - [18] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
 - [19] S. Lederer, Y. Wang, and J. Gao. Connectivity-based localization of large-scale sensor networks with complex shape. *Acm Transactions on Sensor Networks*, 5(4):1–32, 2009.
 - [20] 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 *Proceedings of the ACM Conference on Ubiquitous Computing*, 2012.
 - [21] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A visible light based positioning system. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, 2014.
 - [22] W. Liu, D. Wang, H. Jiang, W. Liu, and C. Wang. An approximate convex decomposition protocol for wireless sensor network localization in arbitrary-shaped fields. *IEEE Transactions on Parallel and Distributed Systems*, 26(12):3264–3274, 2015.
 - [23] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal. Barometric phone sensors: More hype than hope! In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2014.
 - [24] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, 2012.
 - [25] D. Rasch and J. D. Gibbons. Nonparametric statistical inference, 2nd. ed. statistics: Textbooks and monographs vol. 65. marcel dekker, inc., new york and basel. *Biometrical Journal*, 28(8):936–936, 1986.
 - [26] N. Ravi and L. Iftode. Fiatlux: Fingerprinting rooms using light intensity. In *Proceedings of the Fifth International Conference on Pervasive Computing*, 2007.
 - [27] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: Indoor pathway

- mapping made easy. In *Proceedings of USENIX NSDI*, 2013.
- [28] Y. Shu, K. G. Shin, T. He, and J. Chen. Last-mile navigation using smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015.
- [29] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, 2012.
- [30] Y. Wen, X. Tian, X. Wang, and S. Lu. Fundamental limits of rss fingerprinting based indoor localization. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2015.
- [31] C. Wu, Z. Yang, and Y. Liu. Smartphones based crowdsourcing for indoor localization. *IEEE Transactions on Mobile Computing*, 14(2):444–457, 2015.
- [32] B. Xie, G. Tan, and T. He. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015.
- [33] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lv. Maloc: A practical magnetic fingerprinting approach to indoor localization using smartphones. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014.
- [34] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, 2013.
- [35] Q. Xu, A. Gerber, Z. M. Mao, and J. Pang. Acculoc: Practical localization of performance measurements in 3g networks. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, 2011.
- [36] Q. Xu, R. Zheng, and S. Hranilovic. Idyll: Indoor localization using inertial and light sensors on smartphones. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015.
- [37] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 2015.
- [38] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, 2012.
- [39] Z. Yang, Z. Zhou, and Y. Liu. From rssi to csi: Indoor localization via channel response. *ACM Computing Surveys*, 46(2):25:1–25:32, 2013.
- [40] M. Youssef and A. Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, 2005.
- [41] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao. Travi-navi: Self-deployable indoor navigation system. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
- [42] P. Zhou, M. Li, and G. Shen. Use it free: Instantly knowing your phone attitude. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.