IndoorWaze: A Crowdsourcing-Based Context-Aware Indoor Navigation System

Tao Li, *Student Member, IEEE*, Dianqi Han, Yimin Chen, *Student Member, IEEE*, Rui Zhang, *Member, IEEE*, Yanchao Zhang, *Fellow, IEEE*, and Terri Hedgpeth

***Abstract*—Indoor navigation systems are very useful in large complex indoor environments such as shopping malls. Current systems focus on improving indoor localization accuracy and must be combined with an accurate labeled floor plan to provide usable indoor navigation services. Such labeled floor plans are often unavailable or involve a prohibitive cost to manually obtain. In this paper, we present IndoorWaze, a novel crowdsourcing-based context-aware indoor navigation system that can automatically generate an accurate context-aware floor plan with labeled indoor POIs for the first time in literature. IndoorWaze combines the Wi-Fi fingerprints of indoor walkers with the Wi-Fi fingerprints and POI labels provided by POI employees to produce a high-fidelity labeled floor plan. As a lightweight crowdsourcing-based system, IndoorWaze involves very little effort from indoor walkers and POI employees. We prototype IndoorWaze on Android smartphones and evaluate it in a large shopping mall. Our results show that IndoorWaze can generate a high-fidelity labeled floor plan, in which all the stores are correctly labeled and arranged, all the pathways and crossings are correctly shown, and the median estimation error for the store dimension is below 12%.**

***Index Terms*—Indoor navigation, context-aware, labeling, crowdsourcing.**

I. Introduction

There is a strong need for usable navigation services in large indoor environments such as shopping malls, hospital, and museums. For example, shoppers often spend much time looking for the stores they are interested in. Although some shopping malls provide indoor maps at a few locations, it is still not easy for shoppers to understand the maps and reach the target stores. In particular, people with poor spatial awareness, children, senior citizens, and visually impaired users are more in need of a user-friendly indoor navigation system.

We use the shopping-mall example in Fig. 1 to illustrate how a usable indoor navigation system works in our opinion. Alice is now standing beside the GAP store, but her target store is Nike. Since Alice may not know the directions in the shopping mall, it would be confusing for her if we give instructions such as “Go north” or “Go east”. Instead, a more usable instruction can be “Go towards the next Coach store and then keep walking”. The Coach store is adjacent to GAP and is easy to find. When she approaches the Apple store, we can give an instruction like “Turn right at the Apple store and keep walking”. When she approaches the Nike store, we can give the last instruction as “The Nike store is on your right”. All the instructions can be given as audio signals through the smartphone. Alice needs not to take time to understand the physical mall map or watch her phone screen while walking. Such voice instructions are particularly useful for visually impaired users to reach their target stores and also be well aware of the mall environment to gain similar shopping experience enjoyed by sighted people.

To provide the above navigation service, the system needs to keep tracking the shopper in a floor plan with context-aware information such as store names. Most existing indoor localization systems like [1], [2] focus on the localization accuracy and must be combined with a labeled floor plan with location-store mappings to provide the usable navigation service. Such labeled floor plans are often unavailable or quite difficult to obtain. For example, it would be infeasible or incur a prohibitive cost for a national indoor navigation service provider to obtain a labeled floor plan for each indoor environment it intends to cover. While rough floorplans may be available in some scenarios (e.g. shopping malls), they are often only accessible through a static PDF or JPG file with outdated store information. Without accurate dimension and up-to-date store information, they are inadequate to provide the localization and navigation services that we envision. Shen *et al.* [3] and Wang *et al.* [4] explore crowdsourcing to build an indoor pathway map which nonetheless does not contain any store label. To the best of our knowledge, automatic construction of a context-aware indoor floor plan with labeled information is still an open challenge.

In this paper, we present IndoorWaze, a crowdsourcing-based usable indoor navigation system. We use the shopping-mall example throughout the paper for convenience, but

Manuscript received March 4, 2019; revised October 25, 2019 and April 13, 2020; accepted April 13, 2020. Date of publication May 15, 2020; date of current version August 12, 2020. This work was supported in part by the U.S. National Science Foundation under Grant CNS-1933069, Grant CNS-1824355, Grant CNS-1619251, Grant CNS-1514381, Grant CNS-1933047, Grant CNS-1651954 (CAREER), Grant CNS-1718078, and Grant CNS-1700039. The associate editor coordinating the review of this article and approving it for publication was K. Zeng. *(Corresponding author: Yanchao Zhang.)*

Tao Li, Dianqi Han, and Yanchao Zhang are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: tli@asu.edu; dqhan@asu.edu; yczhang@asu.edu).

Yimin Chen is with the Department of Computer Science, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: yiminchen@vt.edu).

Rui Zhang is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716 USA (e-mail: ruizhang@udel.edu).

Terri Hedgpeth is with the Disability Resource Center, Arizona State University, Tempe, AZ 85287 USA (e-mail: terrih@asu.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.2993545

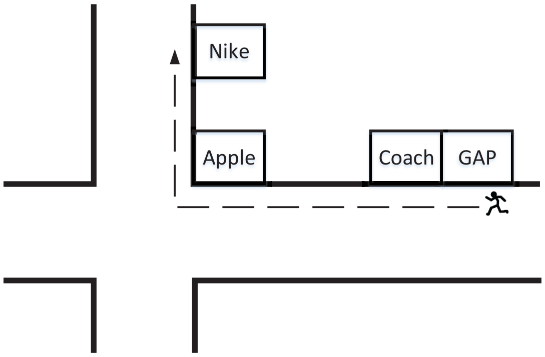


Fig. 1. An indoor navigation example.

IndoorWaze can easily apply to any large complex indoor environment where a usable indoor navigation service is needed. Core to IndoorWave is a novel crowdsourcing-based technique to automatically construct an accurate indoor floor plan with labeled stores. Our technique is motivated by the observation that shoppers often walk around in the shopping mall, while store employees mostly stay in their respective stores. The shoppers help geometrically connect the stores they pass by. When a shopper passes a particular store, the Wi-Fi fingerprints he senses there can be very similar to those measured by the store employees. In addition, the employees of each store can and are motivated to provide a store label. We can then construct a high-fidelity context-aware indoor floor plan by correlating the shoppers' Wi-Fi fingerprints with Wi-Fi fingerprints and store labels offered by store employees.

IndoorWaze is a lightweight crowdsourcing system that involves little effort from participating shoppers and store employees. The tasks of shoppers should be simple enough, as otherwise they may lack incentives to participate. In our system, the only thing shoppers need to do is to allow access to the IMU and RSS data on their smartphones. All the data are collected implicitly while they shop in their usual way. They do not need to take photos [5] or check in manually [6] in the stores. Note that such low requirements on crowdsourcing workers have been proved quite feasible and effective in Waze, Google Map, and other crowdsourcing-based traffic and navigation apps. In contrast, store employees can do slightly more work because they want shoppers to more easily find their stores. In IndoorWaze, store employees are required to collect fingerprints at a few locations near the store according to the store dimension. This one-time work can be easily done within a few minutes, which is quite acceptable.

We make the following contributions. First, we present the first crowdsourcing-based indoor navigation system that can automatically generate a context-aware indoor floor plan. Our system infers the shoppers' walking traces from the IMU sensors on their smartphones and then geometrically connects the stores by mapping them to the walking traces. By combining the walking traces from different shoppers, we can get a high-fidelity floor plan which accurately delineates the labeled stores, pathways, and turning positions. Second, we develop techniques to conquer RSS signal fluctuations which may cause large errors when inferring relative store positions. We also present techniques to extract useful walking traces from the complex data submitted by crowdsourcing shoppers. Third, we implement the system on Android smartphones and evaluate it in a large shopping mall. Our system can generate a high-fidelity labeled floor plan, in which all the stores are correctly labeled and arranged, all the pathways and crossings are correctly shown, and the median estimation error for the store dimension is below 12%.

The rest of the paper is organized as follows. Section II describes the system model and architecture. Section III illustrates the basic technique to construct a rough indoor floor plan. Section IV presents an advanced technique to combine walking traces from different shoppers to construct an accurate labeled floor plan. Section V gives some implementation details of our system. Section VI experimentally evaluates our system. Section VII outlines the related work. Section VIII concludes this paper.

II. System Model and Architecture

A. System Model

There are three entities in the IndoorWaze system: the IndoorWaze service provider, store employees, and shoppers. The service provider releases an app that store employees and shoppers can download and install to their smartphones. The shoppers and employees then register in the app and are required to allow access to the IMU sensors (accelerometer, compass, and gyroscope) in the smartphones. In addition, store employees need to provide their store names in the registration. Store employees are responsible for collecting Wi-Fi RSS fingerprints at a few locations near their store entrances and submit them to the service provider. The sampling locations are picked by the employees themselves and can be a few meters apart from each other. The larger the store, the more sampling locations needed. RSS data collection is a one-time task for employees and takes only a few minutes. The fingerprints provided by the employees of each store act as labeled samples for the store. After that, the employees work as normal. In contrast, the shoppers just walk and shop as usual with their smatphones and do not need to do anything else.

We do not consider security, privacy, and incentive problems which are associated with any crowdsourcing-based system and deserve to be explored in separate papers [7]–[10].

B. System Architecture

Fig. 2 illustrates the architecture of our system. IndoorWaze first constructs the shoppers' walking traces based on their IMU sensors. It then extracts the walking traces which are useful for later floor plan construction. Next, we compare the RSS fingerprints from shoppers and store employees to infer the stores shoppers passed in the walking traces. The system then connects and combines walking traces from different shoppers to form a labeled floor plan. IndoorWaze also recognizes pathways in the floor plan and connects stores along the two sides of each pathway to improve the connectivity of the graph representing the floor plan. When we combine all the walking traces to form a labeled floor plan, the RSS fingerprints are mapped to the floor plan to form a fingerprint

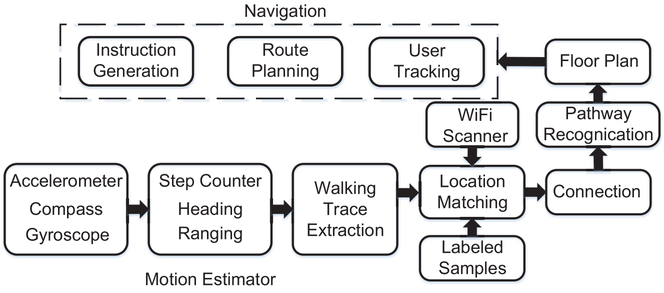


Fig. 2. IndoorWaze system architecture.

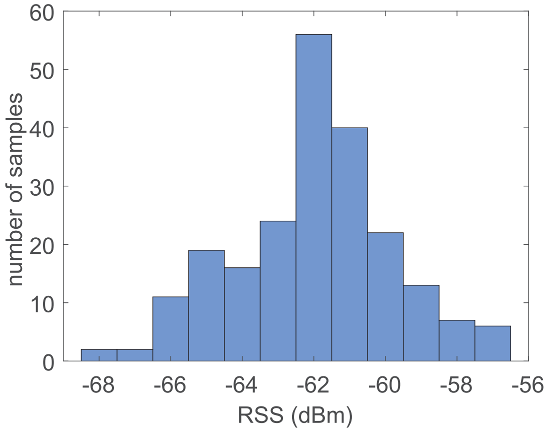


Fig. 3. Histogram of RSS values collected from an AP when the user is static for about 5 minutes.

map as well. After a labeled fingerprint map is constructed, IndoorWaze starts to accept navigation requests from users. The user inputs the target store name and submits his current RSS fingerprint readings. IndoorWaze calculates the user's realtime position using his RSS data and then finds a path to his destination. At last, the system gives audio instructions along the way to the user's destination store according to his changing RSS fingerprints and thus locations.

In Section III, we assume that the shoppers only walk along one side of a pathway for simplicity of descriptions. In Section IV-D, we relax this assumption and present methods to deal with more complex walking traces.

III. Constructing a Rough Floor Plan

In this section, we introduce how to construct a rough floor plan using the RSS fingerprint samples collected by store employees and shoppers.

We assume that there are n stores in the shopping mall. According to the dimension of each store, its employees pick a few locations along the exterior perimeter of the store to collect RSS fingerprint samples. Let $\langle S_1, S_2, \cdots, S_{n'}\rangle$ denote the n' $(n' \geq n)$ sampling positions on the floor plan. At each sampling position, the employee of each store uses the smartphone to collect a set of RSS fingerprints. Each RSS fingerprint is represented by $(rss_1, rss_2, \ldots, rss_m)$, where each rss_i is the received signal strength (RSS) for the ith Wi-Fi access point (AP) for all $1 \leq i \leq m$, and m is the number of APs in the environment.

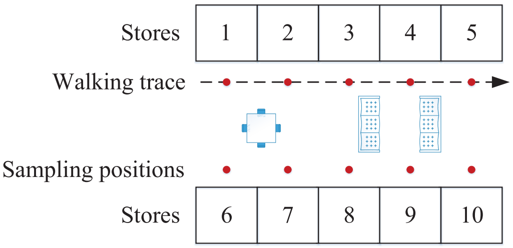


Fig. 4. An exemplary floor plan.

The RSS value at each sampling position typically exhibits fluctuation. For example, Fig. 3 shows the histogram of RSS values for a single AP at the same location during a period of five minutes. Although the RSS values fluctuates over a large range of 11 dBm, they generally follow a Gaussian distribution. As a result, we represent the RSS value at each sampling position S_j for each AP_i using a Gaussian distribution $\mathcal{G}(\mu_{i,j}, \sigma_{i,j}^2)$ fitted from the employee's RSS fingerprint samples, where $\mu_{i,j}$ and $\sigma_{i,j}$ are the mean and standard deviation, respectively. Let $x_1, \ldots, x_s$ are the RSS samples measured at sampling positon S_j from AP_i by the employee. The mean and standard deviation of the Gaussian distribution are computed as

$$\mu_{i,j} = \frac{1}{s}\sum_{k=1}^{s} x_k,$$
$$\sigma_{i,j} = \sqrt{\frac{1}{s-1}\sum_{k=1}^{s}(x_k - \mu_{i,j})^2}. \tag{1}$$

We can then estimate the store passed by a shopper using the maximum likelihood estimation. Specifically, for an RSS fingerprint $f_u = (rss_{u,1}, \ldots, rss_{u,m})$ submitted by a shopper, the likelihood of RSS value $rss_{u,i}$ generated by the Gaussian distribution $\mathcal{G}(\mu_{i,j}, \sigma_{i,j}^2)$ is given by

$$\mathcal{L}_{i,j}(rss_{u,i}) = \frac{1}{\sqrt{2\pi\sigma_{i,j}^2}} e^{-\frac{(rss_{u,i}-\mu_{i,j})^2}{2\sigma_{i,j}^2}}, \tag{2}$$

where we set $rss_{u,i}$ to -100 dbm if the user does not detect any Wi-Fi signal from AP_i for any $1 \leq i \leq m$ [11]. Considering all m RSS values, the likelihood of RSS fingerprint f_u being measured at sampling position S_j is then given by

$$\mathcal{L}_j(f_u) = \prod_{i=1}^{m} \mathcal{L}_{i,j}(rss_{u,i}), \tag{3}$$

where $\mathcal{L}_{i,j}(rss_{u,i})$ is given in Eq. (2). The sampling position passed by the shopper is then estimated as

$$j^* = \arg \max_{j \in \{1,\ldots,m\}} \mathcal{L}_j(f_u). \tag{4}$$

For an RSS fingerprint trace $\langle f_1, \ldots, f_l\rangle$ submitted by a shopper when walking in the mall, we can infer the shopper's location in realtime based on each RSS fingerprint along the trace.

Fig. 4 illustrates a simple floor plan which has two rows of stores and a pathway in the middle. Each store has a sampling

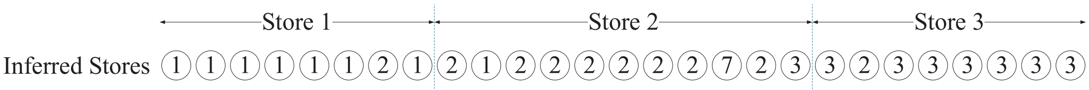


Fig. 5. The inferred stores of each fingerprint sample when the user walks from Store 1 to Store 3.

Store 1
Store 2
Store 3
Store 4
Store 5

Fig. 6. A simple graph corresponding to the walking trace in Fig. 4.

position (red dot). The shopper walks from store 1 to store 5 along the upper side of the pathway. The server calculates the shopper's realtime location by comparing the RSS fingerprints from the shoppers and employees based on likelyhood. Fig. 5 shows a sequence of inferred locations when the shopper walks, where each location is represented by an inferred store index. Normally, when the shopper passes a store, the system ought to pick the store as his current location. Due to signal fluctuations, the system may nevertheless associate the shopper's position with an adjacent store especially when the shopper is in the middle of two stores. The system may also pick the stores on the opposite side of the pathway by mistake. Our remedy comes from the observation that it takes a normal shopper a few seconds to pass a store. By contrast, the mistakes caused by instant signal fluctuations do not last long. For example, assume that w is the store dimension, v is the walking speed, and s is the RSS signal sampling rate. Then the shopper can get $\frac{w}{v} \cdot s$ RSS samples when passing the store. Common shoppers walk slowly in the shopping mall with speed v less than 1 m/s, and most stores are wider than 4 meters. If the RSS sampling rate is 1 Hz as in our prototype, the shopper can at least collect 4 RSS fingerprints which are most similar to the fingerprints collected by the store employees. So we search in Fig. 5 for the stores that appear continuously for at least four times with a sliding window. Finally, we can get a simple graph like Fig. 6.

It is possible that a store may get a wrong adjacent store from a single walking trace for many reasons. For example, the shopper walks so fast that the system cannot capture enough fingerprint samples especially when he passes a small store. In addition, a crowd of people in the mall may cause long-time signal fluctuations which lead to large localization errors. We solve these problems by considering the walking traces from different shoppers. As long as the results from most shoppers are correct, the overall system performance is satisfactory.

IV. Constructing an Accurate Floor Plan

The rough graph generated in the above section only gives the label of each store and the relative positions between adjacent stores. To find the target store with the rough labeled floor plan, the shopper has to look for stores one by one, which is still not user-friendly enough. To make the navigation system more usable, we should give more instructions as described in Section I. As a result, we need a more accurate labeled floor plan which contains the shape and dimension of each store as well as the pathway information. In this section, we combine IMU sensors and the walking traces submitted by multiple shoppers to achieve the above ambitious goal.

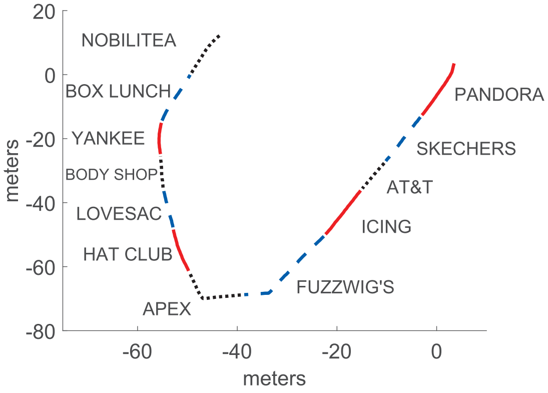


Fig. 7. A walking trace with a sequence of stores the shopper passed.

A. Dealing With a Single Walking Trace

With the help of IMU sensors (accelerometer, gyroscope, and compass), we can recover the shopper's walking trace. In particular, the accelerometer can capture the motion caused by the shopper's walking. After recognizing each step, we can count the steps to infer the shopper's walking distance. The walking direction can be extracted from the electronic compass, but the magnetic signals needed are not stable due to indoor magnetic interference. In contrast, the gyroscope can provide an accurate short-term angle estimation which is free from indoor signal interference, but it can introduce large cumulative errors in the long term. Our system thus combines gyroscope and compass to get a more accurate and stable direction estimation.

Fig. 7 illustrates the walking trace submitted by a shopper, which corresponds to part of the floor plan in a large shopping mall in Fig. 8. The shopper starts from store "PANDORA", walks around, and stops at store "NOBILITEA". The server compares the fingerprints submitted by the shopper with the data from the employees of each store using the method in Section III. Fig. 7 shows that we can correctly infer each store the shopper passed in the walking trace. In addition, we can recover the shape of the original floor plan, so we are able to give more accurate instructions like "Turn right at store APEX". By counting the steps of the shopper when passing the stores, the system can know the dimension of each store as well, which is particularly useful for visually impaired users.

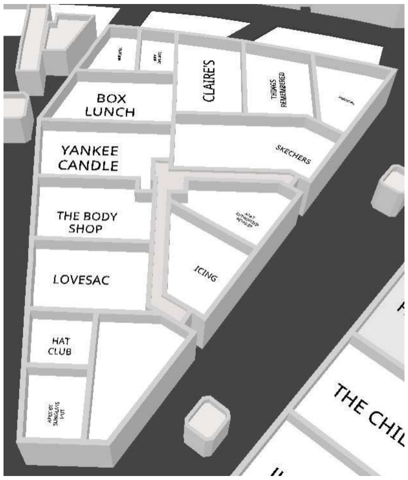


Fig. 8. The original floor plan where the walking trace in Fig. 7 was extracted.

B. Combining Walking Traces From Multiple Users

For a large shopping mall, a single shopper may only walk on part of the floor plan, so we need to connect walking traces from multiple shoppers to cover the whole floor plan. Even for the same part of the floor plan, we can combine the data from multiple users to improve performance. The sampling locations of each store provide good landmarks to combine or connect walking traces from different shoppers.

We design an iterative algorithm similar to the one in Shen *et al.* [3] to combine data from multiple shoppers to construct a more accurate floor plan. Note that the graph in Section III only shows the connectivity between different stores, while Algorithm 1 adds displacements between stores in the graph. Let G denotes the graph that represents the floor plan. In graph G, there are n' vertexes which represent the n' sampling positions in n stores. The edges between vertexes represent displacements between stores.

Let c_i be the current coordinate for vertex i and $\vec{d}_{i,j} = c_i - c_j$ the current displacement between vertexes i and j in the graph. Assume there are N edges (displacements) between vertexes in the graph. By combining the data from multiple shoppers, the system may receive $N_{i,j}$ measurements $\{\vec{r}_{i,j,k} | 1 \le k \le N_{i,j}\}$ for every displacement $\vec{d}_{i,j}$, where measurement $\vec{r}_{i,j,k}$ is extracted from the smartphone's IMU sensor readings. We first initialize all the vertex coordinates to the origin (Lines 1 to 2), so all initial displacements are also zero. We then iteratively update the vertex coordinates based on the measurements. In each iteration (Lines 4 to 18), we first average all the measurements for each current displacement as $\vec{\eta}_{i,j} = \frac{1}{N_{i,j}} \sum_{k=1}^{N_{i,j}} \vec{r}_{i,j,k}$ (Lines 12 to 14). We then sum up the adjustment vectors over all N_i neighboring vertexes as $\vec{S}_i = \sum_j^{N_i} \vec{\eta}_{i,j} - \vec{d}_{i,j}$ (Line 15). The coordinate for vertex i is updated as $c_i = c_i + \frac{1}{N_i}\vec{S}_i$, where $\frac{1}{N_i}\vec{S}_i$ is the average adjustment vector across N_i neighboring vertexes (Line 16). The iteration terminates if the average of $S = \frac{1}{n'} \sum_{i=1}^{n'} \frac{|\vec{S}_i|}{N_i}$ (Lines 17 to 18) is below a threshold α at which point the algorithm outputs the coordinates of each of the n' sampling positions (Line 19).

The running time of Algorithm 1 largely depends on how fast the algorithm converges, which further depends on how consistent the input walking traces are. For each iteration, the running time increases as the number of stores, i.e., the number of vertexes n', and the number of edges N increase. Given the same set of stores, the more walking traces from shoppers, the less uncertainty of the graph G and thus the higher accuracy of constructed floor plan, and vice versa.

Algorithm 1 Iterative Floor-Plan Construction

```
input : Graph $G = (V, E)$, coordinates $\{c_i | 1 \le i \le n'\}$,
        number of adjacent vertexes of $i$th vertex $N_i$,
        displacement measurements
        $\{\vec{r}_{i,j,k} | 1 \le k \le N_{i,j}, 1 \le j \le N_i, 1 \le i \le n'\}$,
        number of edges $N$, threshold $\alpha$.
output: $\{c_i | 1 \le i \le n'\}$.
1  forall the $i \in \{1, \dots, n'\}$ do
2      $c_i \leftarrow (0, 0)$;
3  $S \leftarrow \infty$;
4  while $S > \alpha$ do
5      $S \leftarrow 0$;
6      forall the $i \in \{1, \dots, n'\}$ do
7          $N_i \leftarrow 0$;
8          $\vec{S}_i \leftarrow (0, 0)$;
9          forall the $j \in \{i+1, \dots, n'\}$ and $e(i, j) \in E$ do
10             $N_i \leftarrow N_i + 1$;
11             $\vec{d}_{i,j} \leftarrow c_i - c_j$;
12             $\vec{\eta}_{i,j} \leftarrow (0, 0)$;
13             forall the $k \in \{1, \dots, N_{i,j}\}$ do
14                 $\vec{\eta}_{i,j} \leftarrow \vec{\eta}_{i,j} + \frac{1}{N_{i,j}} \vec{r}_{i,j,k}$;
15             $\vec{S}_i \leftarrow \vec{S}_i + \vec{\eta}_{i,j} - \vec{d}_{i,j}$;
16         $c_i \leftarrow c_i + \frac{1}{N_i} \vec{S}_i$;
17         $S \leftarrow S + \frac{1}{N_i} |\vec{S}_i|$;
18     $S \leftarrow \frac{S}{n'}$;
19 return $\{c_i | 1 \le i \le n'\}$;
```

When the shopper passes two adjacent sampling positions for a store, the system may not locate the shopper correctly especially in a large environment with a very small RSS fingerprint database. If we incorporate the adjacent sampling positions in the algorithm, the results would not be good. However, it is relatively easy to locate the shopper between two sampling positions of adjacent stores because of the large interval. As a result, we only use one sampling position for a store in the construction algorithm and explore other sampling positions as well to infer the store dimensions.

C. Pathway Recognition

Fig. 9 illustrates a representive distribution of stores in a shopping mall. The stores can be adjacent to each other like stores 1 and 2, on the opposite sides of a pathway like stores 3 and 8, or back-to-back with each other like stores 7 and 12. The adjacency relations have been inferred by our techniques in previous sections. The back-to-back stores are not directly reachable and thus do not affect the performance of IndoorWaze. The stores on the opposite sides of the pathway

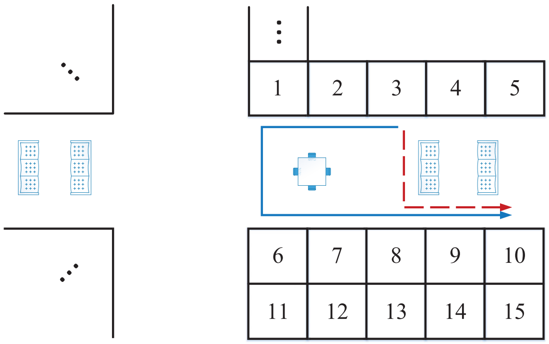


Fig. 9. An exemplary floor plan with three common geometric store relations.

are often directly reachable, but their relations cannot be captured by previous techniques. For example, stores 3 and 8 are physically directly reachable from each other, but they are not directly connected in the graph formed with our previous techniques. The disconnectivity of stores on the opposite sides of the pathway may cause problems in the navigation. For example, a shopper is now at store 3, but his destination is store 10. The system may navigate the shopper along the blue solid route in Fig. 9, because the system thinks by mistake that stores 1 and 6 are adjacent to each other, but stores 3 and 8 are not directly reachable. Actually, the shortest path from store 3 to store 10 should be the red dashed line.

From the discussion above, it is important to recognize each pathway in the graph and correctly associate the stores on its two sides. For example, we need to know there is a pathway between stores 1 and 6 but not between stores 6 and 11. We use some simple criteria to identify such relations with stores 1, 6, and 11 as an example, which are situated around the intersection of pathways. First, there exist straight (e.g., north-south in Fig. 9) walking traces between two stores. Second, there are walking traces that pass either store in other directions (e.g., west-east in Fig. 9). The walking traces under these two criteria should cross with each other. Third, the likelihood of collecting similar fingerprints along the walking trace between the two stores is relatively low. For example, the walking trace from store 6 to store 1 meets the third criterion, but the one from store 6 to store 11 does not because there is a big space gap between stores 1 and 6 but not between stores 6 and 11. Our previous techniques have identified adjacent stores along either side of a pathway. After we know the pathway between stores 1 and 6, we can correctly associate the other stores along the two sides of the pathway. Fig. 10 is the new graph after we add edges (dotted lines) to stores on the opposite sides of the pathway. Sometimes two stores on the opposite sides may not be directly reachable in practice, so we need to double-check if there is a direct walking trace between each pair of opposite stores in the original walking traces.

D. Dealing With Complex Walking Traces

Previous sections only consider the shoppers walking along one side of the pathway, but some shoppers may walk between stores on the opposite sides of the pathway. As a result, the walking traces from many shoppers become too complex to construct a floor plan. Now we present methods to extract walking traces which are useful in the floor-plan construction. We observe that the indoor layout imposes different patterns on the shoppers' walking traces. For example, many people tend to make a turn at the intersection stores (e.g., stores 1 and 6 in Fig. 9) rather than around the stores in the middle of one row (e.g., stores 2 to 4). As a result, we can count the number of shoppers who turn at each store to infer the stores at turning positions. Then we search for straight walking traces between turning stores. For example, a walking trace from store 1 to 5 is a valid trace which can be used in the floor-plan construction. By contrast, a walking trace that passes store 1, 2, 3, 8, 9, and 10 is rejected because there is a turn at store 3. Other walking traces in the original data set are not useless and can still be used to infer the connectivity between stores on the two sides of the pathway.

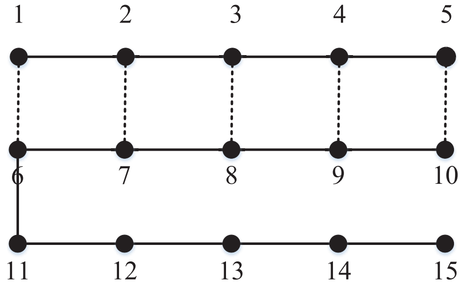


Fig. 10. A refined graph representation of the floor plan in Fig. 9.

E. Navigation

With the aforementioned techniques, the system can construct a labeled indoor map with Wi-Fi RSS fingerprints to provide usable indoor navigation services. Let us continue the previous shopping-mall example. The shopper first inputs his target store into the IndoorWaze app. The system then estimates the shopper's current location based on his current fingerprint readings and the fingerprint map. After that, the system searches in the floor plan and finds a shortest path from his current location to the target store. Based on the path and the shopper's current location, the system provides context-aware audio instructions to the shopper. The system also tracks the shopper's realtime positions using his RSS fingerprints and provides new instructions if he deviates from the path. The RSS fingerprints the shopper measures may not be stable when walking. If we infer his realtime locations based on only the most recent RSS fingerprints, the calculated positions would be back and forth because of signal fluctuations. Similar to what we have proposed previously, we can determine whether the shopper is passing a store only when he can report at least 4 continuous RSS fingerprints which are most similar to those collected by the store employees. In addition, we can combine the shopper's IMU sensors and the floor plan to detect false position estimates. For example, the shopper in Fig. 9 started from store 2 and wants to reach store 5, and he is currently at store 3. From his IMU sensors, the system knows that he walks in a straight line. If the position estimate changes from store 3 to 2, our system can immediately catch this wrong location estimate. Such navigation details are well considered in our system.

F. RSS Fingerprint Update

The RSS distribution at a sampling position may change due to indoor layout update. While the employee of nearby stores can easily notice the change and recollect the RSS fingerprint samples, such change may not be noticed by the employee of stores that are further away. It is possible to design and develop an automatic mechanism for detecting RSS fingerprint change to keep the RSS distribution up-to-date at each sampling location. Specifically, we can maintain the RSS distribution using the most recent RSS fingerprints from users at the same sampling location using a sliding window. For example, the most recent RSS distribution can be fitted from the 100 most recent ones or the ones received within the last 24 hours. If the difference between the most recent RSS distribution and the original RSS distribution exceeds certain threshold, we can update the RSS distribution at the sampling location using the most recent RSS fingerprint samples and let the app inform the nearby store's employee to collect additional samples if needed. We leave the detailed investigation of this issue as our future work.

V. System Implementation

We implemented IndoorWaze with Java on a Google Nexus 6 smartphone which has a Quad-core 2.7 GHz Krait 450 CPU, 3 GB RAM, a 5.96-inch display, and four relevant IMU sensors (magnetometer, compass, accelerometer, and gyroscope). The sampling frequencies for IMU sensors and the Wi-Fi module are 16.7 Hz and 1 Hz, respectively. Below, we briefly describe some implementation details about step detection, turn detection, and AP filtering.

A. Step Detection

The system estimates the walking distance of shoppers by counting their steps. The accelerometer can capture the motion caused by walking and recognize each step. For example, let (a_x, a_y, a_z) denote an accelerometer reading which represents the acceleration along the three axes. In each step, the amplitude $\sqrt{a_x^2 + a_y^2 + a_z^2}$ reaches a maximum when the shopper's heel strikes the ground. As a result, we just need to count the amplitude peaks in a sequence of readings to infer the number of steps.

Fig. 11(a) illustrates the amplitudes of a sequence of acceleration readings. It is difficult to count the peaks caused by each step because the shopper's random movements can introduce many peaks in the signal. We first process the data using a moving average filter to obtain Fig. 11(b) which is much smoother than Fig. 11(a). We then use a low-pass filter to remove some high-frequency components caused by random phone movements. Fig. 11(c) illustrates the signal after filtering, from which we can easily identify the peak caused by each step. To count the peaks, we have to know a rough period of the signal, which can be inferred by doing a signal autocorrelation. After that, we use a sliding window to search and count the peaks in the signal. The length of the sliding window t_w can be equal to the period of the signal. The ith data sample is a peak if it is larger than all the samples in $[t(i) - t_w/2, t(i) + t_w/2]$, where $t(i)$ is the time stamp of the ith data sample.

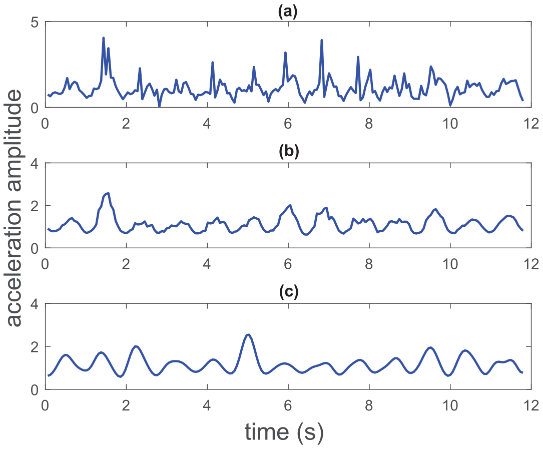


Fig. 11. Acceleration data processing for step detection.

B. Turn Detection

The compass in the smartphone works well in the outdoor environment. However, the indoor magnetic environment is so complex that the compass cannot provide reliable direction estimation. Fortunately, the gyroscope can provide accurate short-term angle changes and is not affected by indoor magnetic signals. In our system, we combine the accelerometer and gyroscope to provide stable turn detection.

The gyroscope reading (g_x, g_y, g_z) represents realtime phone angle changes in the 3-axis smartphone coordinate system. To detect the direction changes of walking users, we need to convert the gyroscope readings to those in the Earth coordinate system. Before the axis rotation, we have to know the realtime altitude of the phone. Assume that (ϕ, θ, ψ) represents the phone's altitude angles in the three axes of the Earth coordinate system. Based on the gravity sensor (i.e., the vertical component of the accelerometer), we can only infer ϕ and θ which seem not enough to calculate the exact altitude in the 3-axis coordinate system. Fortunately, the user normally walks in a two-dimensional $x - y$ plane, so what we care about is not the exact yaw angle ψ but an angle change around the z axis. Therefore, we can calculate the gyroscope data in the Earth coordinate system as $(e_x, e_y, e_z) = R_y(\theta) \cdot R_x(\phi) \cdot (g_x, g_y, g_z)$, where $R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix}$ and $R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix}$ are the rotation matrices around the x and y axes, respectively. After the axis rotation, we use e_z to measure the angle changes of the walking user.

TABLE I
WI-FI AP ANALYSIS

Frequency	Encrypted?	Has a name?
2.4 GHz: 250	Yes: 411	Yes: 281
5 GHz: 237	No: 76	No: 206

C. Filtering Access Points

The Wi-Fi environment in the shopping mall is very complex. In addition to the APs set up by the shopping mall, every store can install its own APs. Sometimes we may even find some hotspots served by mobile devices which are not stable signal sources. In the experiment area, we found 487 APs in total. If we incorporate all the APs into the experiments, the performance would be bad. We therefore consider three possible criteria to classify the APs as shown in Table I based on the information we collected. In particular, we observed that the signals from the APs with names are more stable, because these APs are often installed by shopping mall operators and stores (e.g. AT&T store) and have fixed locations. In contrast, the signals from those APs without names exhibit large fluctuation and are thus less stable. In addition, the signals of the APs classified under the other two criteria do not differ too much. We therefore only use named APs in our experiments.

VI. SYSTEM EVALUATION

IndoorWaze was evaluated in a large shopping mall in our metropolitan area. The mall covers 120,000 m^2 and consists of 213 stores. In the experiment, we mimicked both shoppers and store employees. The mall is so large that we cannot cover the whole area or all the stores. Therefore, the experiment was done in part of the shopping mall as illustrated in Fig. 12. The experiment area is about 6,000 m^2 and contains 25 stores. We chose this area because it contains all possible indoor features such as pathways and crossings.

Our evaluation mainly focuses on the construction of the labeled floor plan which is the main contribution of this paper. For completeness, we also test the room level localization and navigation based on the generated floor plan using Horus [12]. It is difficult to compare the IndoorWaze with prior work quantitatively, since different works use different data sources, provide different functions, or work in different settings. In addition, quantitative comparison will be affected by many factors and parameters in real experiment that we cannot control. Instead, we provide a qualitative comparison with prior work in Table II.

A. Evaluation of Floor Plan Construction

1) Data Collection: We collected data during the normal operating hours of the shopping mall. We first acted as the store employees to collect fingerprints along each store in the floor plan. For this purpose, we picked the number of sampling positions according to the dimension of each store. Fig. 13 illustrates a histogram of the number of sampling positions along the stores. Most stores are not very large and just need three sampling positions. We first collected Wi-Fi RSS fingerprints at every sampling position for 20 seconds. We then held the phone and walked around for about 20 minutes at a normal speed (0.8 m/s). During the 20-minute collection period, we collected 14 fingerprint traces to cover each store at least twice.

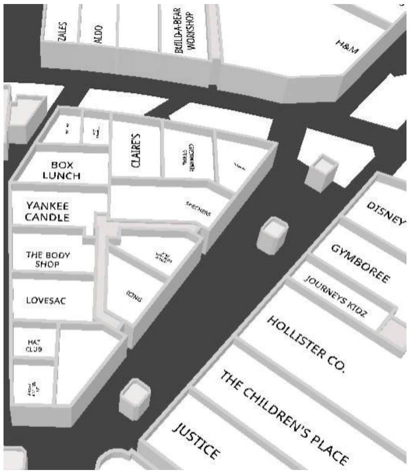


Fig. 12. The real floor plan in our experiment.

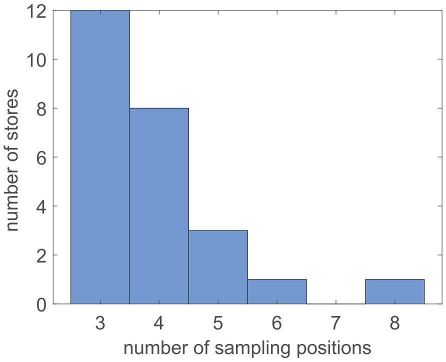


Fig. 13. Number of sampling positions for stores.

Fingerprint collection by shoppers under IndoorWaze is the same as in Walkie-markie [3] and UnLoc [4] and thus they incur similar time cost. IndoorWaze additionally requires store employees to collect fingerprint samples at sampling locations near their stores. Such cost is very acceptable in practice, as it suffices to take samples for about 20 seconds at each sampling location. While we expect that more samples can improve the localization accuracy, the benefit from additional samples diminishes as the number of samples increases. As a result, the cost of data collection under IndoorWaze is similar to that under prior solutions Walkie-markie [3] and UnLoc [4] and is significantly lower than prior floor plan construction systems that require more complex information such as image [15], video [16], and check-in [6] that require explicit participation of shoppers.

2) Visual Comparison: Fig. 14 depicts the graph representation of the floor plan generated by our system. The stores at the crossings are represented by red dots, the edges along

TABLE II
SUMMARY OF RELATED WORK

System	Method	User's task	Navigation	Floor plan	Granularity	Flexibility	labelling
Travi-Navi [5], FollowMe [13], [14]	Leader-follower	Not required	✓	–	–	Low	–
SemSense [6]	Manual check-in	Explicit	–	–	–	Low	✓
Jigsaw [15], CrowdMap [16]	Image, Video	Explicit	–	✓	Room Level	High	–
Walkie-markie [3], UnLoc [4]	RSS, IMU	Implicit	–	✓	Pathway Level	High	–
IndoorWaze	RSS, IMU from both shoppers & employees	Implicit	✓	✓	Room Level	High	✓

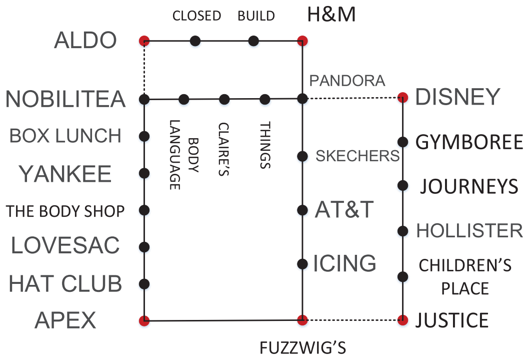


Fig. 14. Graph representation of the floor plan with IndoorWaze.

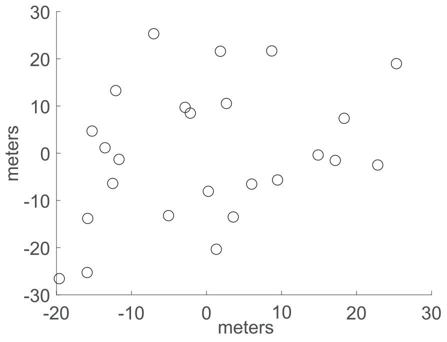


Fig. 15. Floor plan after 5 iterations.

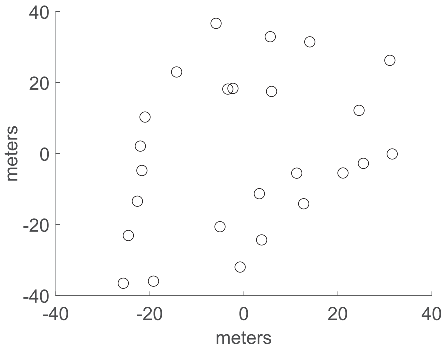


Fig. 16. Floor plan after 10 iterations.

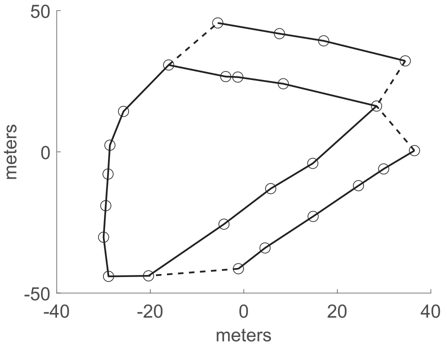


Fig. 17. Floor plan after 20 iterations.

either side of a pathway are shown by solid lines, and edges across the pathways are represented by dotted lines. Fig. 14 only shows the relative positions among adjacent stores, and the resulting floor plan is still too rough for usable indoor navigation. Fig. 15 to Fig. 17 show the generated floor plan after 5, 10, and 20 iterations, respectively. The floor plan in Fig. 17 is very similar to the ground truth in Fig. 12. It clearly identifies the pathways and differentiates the stores along two sides of the pathways. All the store labels are also correct.

3) Dimension Estimation: In addition to the relative store positions, our system can estimate the dimension of each store. Fig. 18 illustrates the errors in the store-dimension estimation, where the median error is about 2.5 steps. Actually, the estimation errors are closely related to the true store dimensions. Fig. 19 illustrates the ratio of errors to the store dimension, where the median error ratio is about 12%. By accurately estimating the store dimensions, we can infer the duration for a shopper to pass a particular store.

4) Impact of Participation: IndoorWaze is a crowdsourcing-based system, and not all the stores may participate in data collection. In the experiment, we randomly removed a few stores from the dataset and evaluated the impact on the floor-plan construction. Fig. 20 shows the floor plan after we randomly removed three stores. The floor plan is not as good as the one in Fig. 17, but it is still sufficient for highly usable indoor navigation.

B. Localization and Navigation

1) Static Localization: We evaluated the localization performance of IndoorWaze when the shopper is static. In this experiment, we stood still at the center of each store's exterior

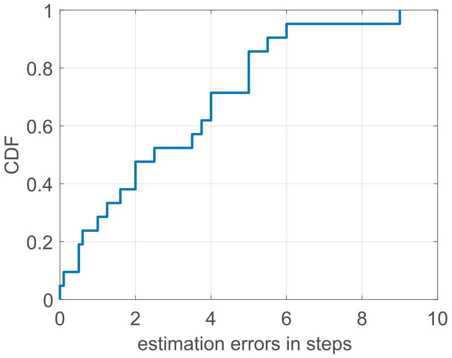


Fig. 18. Accuracy of store-dimension estimation.

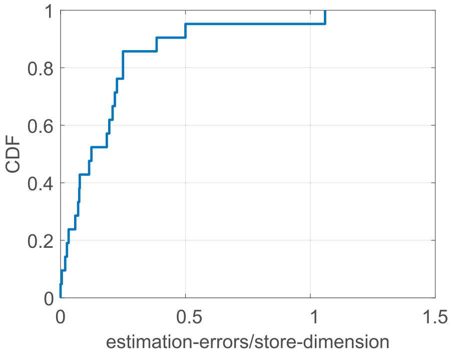


Fig. 19. Ratio-based accuracy of store-dimension estimation.

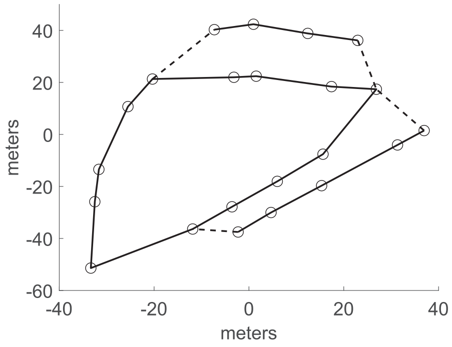


Fig. 20. Floor plan after 20 iterations with 3 non-participating stores.

perimeter for about 10 seconds. Then, we computed the average of all the collected RSS fingerprints to find the best match in the RSS fingerprint database. Our system found the correct stores for all the 25 location queries.

2) Navigation: We also tested the navigation service based on the generated floor plan. In this experiment, we randomly picked four destination stores (“BUILD”, “CLAIRE’S”, “AT&T”, and “JOURNEYS”) and a starting store “LOVESAC” in Fig. 14. We successfully reached each destination store at ease by following IndoorWaze’s audio instructions.

VII. Related Work

Most existing indoor localization/navigation systems focus on improving the localization accuracy. To provide a user-friendly navigation service, these systems need a labeled floor plan with location-store mappings. However, such context-aware labeled floor plans are often unavailable or quite prohibitive to obtain. Our work fills this gap and generates a high-fidelity labeled floor plan that can be used with most existing indoor localization/navigation system. In what follows, we describe some representative work on indoor localization systems, indoor navigation systems, and indoor floorplan construction. Table II summarizes some most germane work.

Fingerprint-based techniques are the most popular approaches for indoor localization. As probably the first work along this line, Radar [17] is a deterministic localization method that employs RSS for indoor localization. Horus [12] improves Radar by keeping a fingerprint distribution for every position in the floor plan and then finding a maximum likelihood match in the database. SurroundSense [18] introduces more indoor features (such as light and sound) in addition to RSS fingerprints.

Model-based indoor localization techniques estimate indoor locations using statistic models. A popular approach is to build a relation between RSS and signal propagation distance based on the RF propagation model (e.g., the log-distance path loss (LDPL)) [19]. The work in [20] evaluates some self-calibrating algorithms in office environments and finds that the median errors are consistently greater than 5m. In addition to LDPL-based schemes, there are other techniques based on Angle of Arrival (AoA) [21], Time of Arrival (ToA) [22], and Time Difference of Arrival (TDoA) [23].

Simultaneous Localization and Mapping (SLAM) is a technique originating from the robotics community. SLAM relies on a robot to explore the space of interest with discrete landmarks or obstacles. Based on the laser ranging and cameras in the robot, we can determine the relative locations of the landmarks, and the robot can infer its relative location. WiFi-SLAM [24] uses a Gaussian process to model the relation of Wi-Fi signal strengths. With more sensors embedded in the smartphone, many techniques combine IMU sensor data with human movements to realize SLAM. For example, the schemes in [25], [26] use the indoor constraints to map human mobility traces to the floor plan and then generate the fingerprint database. LiFS [27] maps fingerprints by comparing the similarity between the high-dimensional fingerprint space and the stress-free floor plan.

There are also systems based on a leader-follower navigation model. Escort [3] navigates users based on crowd-encounter information and dead reckoning. Magnetic information is adopted by some systems [13], [14] for navigation. The leader first records the magnetic signal along the route. The user then compares the magnetic signals he collects with the signals from the leader to determine his location. Travi-Navi [5] is a vision-guided navigation system. The leader (store employee) first takes photos along the route to the store. The system then gives the instructions based on the photos and dead-reckoning techniques. This line of systems do not construct a context-aware floor plan.

People have also studied indoor floor-plan construction. SemSense [6] presents a floor-plan labeling method based on a given unlabeled floor plan and shoppers' manual check-ins in each store. Jigsaw [15] presents a floor-plan construction technique by combing IMU sensors and landmarks extracted from images taken by crowdsourcing workers. CrowdMap [16] is a sensor-rich video-based approach for indoor floor-plan construction, which furnishes the consistent video frame relation to generate spatial information for the indoor environment. Unloc [4] constructs the floor plan by sensing natural indoor landmarks such as elevators and stairs, which are then connected via dead reckoning. Walkie-markie [3] generates the floor plan based on the RSS trend when the user passes the AP. Other IMU-based floor-plan construction methods are presented in [28]–[30]. None of these techniques could automatically generate a high-fidelity indoor floor plan with accurate POI labels and dimensions. In addition, our system IndoorWaze incurs minimal effort on crowdsourcing users.

VIII. Conclusion

This paper presented IndoorWaze, the first indoor navigation system which can automatically construct an accurate labeled indoor floor plan. We prototyped IndoorWaze on Android smartphones and evaluated it in a large shopping mall. The experiment confirmed the high efficacy and usability of IndoorWaze.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive and helpful advice.

References

[1] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-Level localization with a single WiFi access point," in *Proc. USENIX NSDI*, Santa Clara, CA, USA, Mar. 2016, pp. 165–178.

[2] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proc. MobiCom*, Maui, HI, USA, 2014, pp. 165–178.

[3] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie–Markie: Indoor pathway mapping made easy," in *Proc. USENIX NSDI*, Lombard, IL, USA, Apr. 2013, pp. 85–98.

[4] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM MobiSys*, Low Wood Bay, U.K., 2012, pp. 197–210.

[5] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi–Navi: Self-deployable indoor navigation system," in *Proc. ACM MobiCom*, Maui, HI, USA, Sep. 2014, pp. 471–482.

[6] M. Elhamshary and M. Youssef, "SemSense: Automatic construction of semantic indoor floorplans," in *Proc. IEEE IPIN*, Calgary, AB, Canada, Oct. 2015, pp. 1–11.

[7] T. Li, Y. Chen, R. Zhang, Y. Zhang, and T. Hedgpeth, "Secure crowdsourced indoor positioning systems," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, Apr. 2018, pp. 1034–1042.

[8] L. Yuan, Y. Hu, Y. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Secure RSS-fingerprint-based indoor positioning: Attacks and countermeasures," in *Proc. IEEE CNS*, Beijing, China, May 2018, pp. 1–9.

[9] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Zhao, "Defending against sybil devices in crowdsourced mapping services," in *Proc. ACM MobiSys*, Singapore, Jun. 2016, pp. 179–191.

[10] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, Jun. 2016.

[11] *Understanding WiFi Signal Strength*. Accessed: Apr. 8, 2020. [Online]. Available: https://www.metageek.com/training/resources/wifi-signal-strength-basics.html

[12] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. ACM MobiSys*, Seattle, WA, USA, Jun. 2005, pp. 205–218.

[13] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-mile navigation using smartphones," in *Proc. ACM MobiCom*, Paris, France, 2015, pp. 512–524.

[14] T. H. Riehle *et al.*, "Indoor magnetic navigation for the blind," in *Proc. IEEE EMBS*, San Diego, CA, USA, Sep. 2012, pp. 1972–1975.

[15] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Bian, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. ACM MobiCom*, Maui, HI, USA, Sep. 2014, pp. 249–260.

[16] S. Chen, M. Li, K. Ren, and C. Qiao, "CrowdMap: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos," in *Proc. IEEE ICDCS*, Columbus, OH, USA, Jun. 2015, pp. 1–10.

[17] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[18] M. Azizyan, I. Constandache, and R. Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. ACM MobiCom*, Beijing, China, Sep. 2009, pp. 261–272.

[19] H. Lim, L. Kung, J. Hou, and H. Luo, "Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure," *Wireless Netw.*, vol. 16, no. 2, pp. 405–420, 2010.

[20] D. Turner, S. Savage, and A. Snoeren, "On the empirical performance of self-calibrating WiFi location systems," in *Proc. IEEE LCN*, Bonn, Germany, Oct. 2011, pp. 76–84.

[21] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," in *Proc. USENIX NSDI*, Lombard, IL, USA, Apr. 2013, pp. 71–84.

[22] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "PinPoint: An asynchronous time-based location determination system," in *Proc. ACM MobiSys*, Uppsala, Sweden, Jun. 2006, pp. 165–176.

[23] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. ACM MobiCom*, Boston, MA, USA, Aug. 2000, pp. 32–43.

[24] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. IJCAI*, Hyderabad, India, Jan. 2007, pp. 2480–2485.

[25] A. Rai, K. Chintalapudi, V. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM MobiCom*, Istanbul, Turkey, Aug. 2012, pp. 293–304.

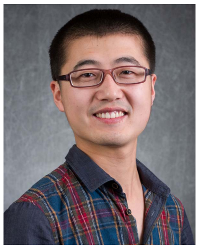
[26] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM UbiComp*, Pittsburgh, PA, USA, Sep. 2012, pp. 421–430.

[27] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM MobiCom*, Istanbul, Turkey, Aug. 2012, pp. 269–280.

[28] M. Alzantot and M. Youssef, "CrowdInside: Automatic construction of indoor floorplans," in *Proc. ACM GIS*, Redondo Beach, CA, USA, Nov. 2012, pp. 99–108.

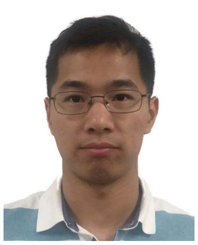
[29] Y. Jiang *et al.*, "Hallway based automatic indoor floorplan construction using room fingerprints," in *Proc. ACM UbiComp*, Zürich, Switzerland, Sep. 2013, pp. 315–324.

[30] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 889–898, Nov. 2012.



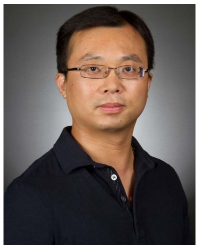
Tao Li (Student Member, IEEE) received the B.E. degree in software engineering from Hangzhou Dianzi University, China, in 2012, and the M.S. degree in computer science from Xi'an Jiaotong University, China, in 2015. He is currently pursuing the Ph.D. degree in computer engineering with Arizona State University. His research interests include indoor navigation, security, and privacy issues in computer and networked systems.

Dianqi Han received the B.S. degree in information security from the University of Science and Technology of China, China, in 2010, and the M.S. degree in electrical and computer engineering from the University of California at Davis, Davis, in 2015. He is currently pursuing the Ph.D. degree in computer engineering with Arizona State University. His research interests include indoor navigation, security, and privacy issues in computer and networked systems.

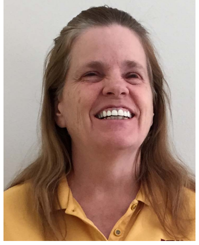


Yimin Chen (Student Member, IEEE) received the B.S. degree from Peking University, China, in 2010, the M.Phil. degree from The Chinese University of Hong Kong, Hong Kong, in 2013, and the Ph.D. degree from Arizona State University, Tempe, in 2018, all in electrical engineering. He is currently a Post-Doctoral Associate with the Computer Science Department, Virginia Tech. His research interests include security and privacy in machine learning systems.

Rui Zhang (Member, IEEE) received the B.E. degree in communication engineering and the M.E. degree in communication and information system from the Huazhong University of Science and Technology in 2001 and 2005, respectively, and the Ph.D. degree in electrical engineering from Arizona State University in 2013. He was an Assistant Professor with the Department of Electrical Engineering, University of Hawaii at Manoa, from 2013 to 2016. He has been an Assistant Professor with the Department of Computer and Information Sciences, University of Delaware, since 2016. His current research interests include network and distributed system security, wireless networking, and mobile computing. He received the NSF CAREER Award in 2017.



Yanchao Zhang (Fellow, IEEE) received the B.E. degree in computer science and technology from the Nanjing University of Posts and Telecommunications in 1999, the M.E. degree in computer science and technology from the Beijing University of Posts and Telecommunications in 2002, and the Ph.D. degree in electrical and computer engineering from the University of Florida in 2006. He is currently a Professor with the School of Electrical, Computer and Energy Engineering, Arizona State University. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is an IEEE Fellow for contributions to wireless and mobile security. He received the U.S. NSF CAREER Award in 2009. He is/was on the Editorial Boards of the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He also chaired the 2017 IEEE Conference on Communications and Network Security (CNS), the 2016 ARO-Funded Workshop on Trustworthy Human-Centric Social Networking, the 2015 NSF Workshop on Wireless Security, and the 2010 IEEE GLOBECOM Communication and Information System Security Symposium.



Terri Hedgpeth received the B.S. degree in psychology and the Ph.D. degree in higher education leadership and policy studies from Arizona State University. She is currently the Director of the Disability Resource Center, Arizona State University.