GROPING: Geomagnetism and cROwdsensing Powered Indoor NaviGation

Chi Zhang, Kalyan P. Subbu, Jun Luo, and Jianxin Wu

Abstract—Although a large number of WiFi fingerprinting based indoor localization systems have been proposed, our field experience with Google Maps Indoor (GMI), the only system available for public testing, shows that it is far from mature for indoor navigation. In this paper, we first report our field studies with GMI, as well as experiment results aiming to explain our unsatisfactory GMI experience. Then motivated by the obtained insights, we propose GROPING as a self-contained indoor navigation system independent of any infrastructural support. GROPING relies on geomagnetic fingerprints that are far more stable than WiFi fingerprints, and it exploits crowdsensing to construct floor maps rather than expecting individual venues to supply digitized maps. Based on our experiments with 20 participants in various floors of a big shopping mall, GROPING is able to deliver a sufficient accuracy for localization and thus provides smooth navigation experience.

Index Terms—Indoor Navigation, Indoor Localization, Geomagnetism, Mobile Crowdsensing

1 Introduction

Successful indoor navigation requires computing location information and visualizing that information on a map *in real-time*. Though commercial products (e.g., [2], [3]) and innumerable academic solutions (e.g., [5], [14], [37], [32]) have been developed for indoor localization, indoor navigation still appears to be a challenging issue. On one hand, wireless signal (e.g., WiFi and GSM), the most exploited source for inferring location [37], [34], [24], [15], may not be suitable for navigation purposes. On the other hand, presuming the availability of floor maps is common in most existing proposals, but digitized floor maps are not easily available due to proprietary and privacy issues.

It is well known that RF signals suffer from instability, which implies that achieving a satisfactory location accuracy demands heavy computations [37]. Moreover, RF sensing is notoriously energy consuming. As both factors go against navigation that entails a continuous and real-time location estimation, a fully functional navigation service seems to demand a lightweight localization scheme efficient in both computation and energy consumption.

The dependence of navigation on digitized maps is not as strong as we often expect. As described in [16], people build cognitive maps by subconsciously remembering landmarks and moving between them to reach their destinations. Therefore, the imperceptible signs contained in a digitized map may not be that relevant; a more practical solution could be to involve human users themselves to collectively construct a map and also to provide semantic landmark information. Specifically, people carrying smartphones loaded with sensors can either volunteer or be recruited to gather information from the ambient environment for both map construction and landmark identification. This form of information collection through human participation is indeed a type of *mobile crowdsensing* [12].

Localization commonly requires a fingerprint library against which certain newly sampled signal may compare and hence determine the location. However, the localization function required by indoor navigation differs in two main aspects from a pure localization scheme that pinpoints the current position of a user. On one hand, it requires real-time and constant location computations. This means that it demands very stable fingerprints, as it may not afford comparing with a library in which a single location is associated with a large number of fingerprints (e.g., WiFi fingerprints [23], [15]). On the other hand, it does not require a very high accuracy, as the navigation service only needs to lead a user to a point within the visual range of the actual destination. This makes it unnecessary to have a meter level accuracy achieved by, for example, dead reckoning systems [19] at the cost of handling directional/drift errors and performing calibrations/computations with multi-sensor data on resource constrained devices. Therefore, our design applies the magnetometer and exploits *geomagnetism* as the location indicating fingerprint: it is lightweight (only a 3D vector) and very stable, and it is completely independent of any kind of wireless infrastructure.

To better motivate our design philosophy, we first report a study on Google Maps Indoor (GMI) [3], the only indoor navigation system available for *public* testing, as well as on basic properties of both WiFi and geomagnetism in location estimation; this study reveals issues pertaining to the aforementioned ones. In response to these issues, we

- *Chi Zhang and Jun Luo are with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. E-mail: {czhang8, junluo}@ntu.edu.sg*
- *Kalyan P. Subbu is with Amrita University, India. The work was done when he was a post-doctoral researcher at NTU. E-mail: kalyansasidhar@am.amrita.edu*
- *Jianxin Wu is with the National Key Laboratory for Novel Software Technology, Nanjing University, China. E-mail: wujx2001@nju.edu.cn*

propose *Geomagnetism and cROwdsensing Powered Indoor NaviGation* (GROPING) as a completely self-contained, lightweight, and practical prototype for indoor navigation. GROPING encapsulates three functions, namely *map building*, *localization* and *navigation*, into one unit. It first builds a map using user contributed sensor data and semantic labels; it then performs localization based on the magnetic fingerprints, and finally it runs a navigation service on top of these two functions: it computes navigational routes using the early constructed map and the real-time location information. In this way, GROPING eliminates *infrastructure dependence*: it needs neither wireless infrastructure nor digitized floor maps. Our intensive experiments with GROPING demonstrate its usability and also show that it compares favorably with typical WiFi-based localization systems in supporting indoor navigation.

2 Studies on Google Maps Indoor

In spite of the huge numbers of proposals on indoor localization, the only system that is available for public testing is Google Maps Indoor (GMI) [3]. Therefore, we organize a group of 11 people to perform a detailed study on it. Given that GMI appears to a user as a blackbox, our study is separated into three parts. The first part is a field study in five big-scale shopping malls (above 10000 m^2) to test the accuracy of GMI, as well as to make sure if WiFi is used by GMI (which appears to be true). The second part, assuming WiFi is the main source for GMI, is a lab test on the stability of WiFi fingerprints; it aims to explain the observation that we have made in the field study. The third part reports an evaluation of the energy efficiency of WiFi-based localization systems, leveraging on the energy profiles obtained as a by-product of the earlier studies.

2.1 A Field Study on GMI

As GMI works only for venues that contribute floor maps to Google, we are confined in choosing test sites (Fig. 1(a) shows two of them). In fact, only 11 shopping malls in Singapore have GMI support available. The mobile phones we use include Samsung Galaxy S2/S3, Sony Xperia S, and HTC One X. In this study we mainly want to answer the following three questions.

- **Q1**: What is the accuracy of GMI's localization?
- **Q2**: Does GMI's navigation work well?
- **Q3**: Does GMI heavily rely on WiFi infrastructure?

2.1.1 Location Accuracy

The team members unanimously agree that GMI usually produce unsatisfactory localization accuracy. We first show a few screenshots taken on GMI in Fig. 1(b), in which both actual locations (pinpointed by the users on-site) and the locations indicated by GMI (the blue arrows) are shown.

To quantify GMI's localization accuracy, we perform tests at 30 randomly chosen positions in each of the five malls. The accuracy results are shown in Fig. 2 (with the number of available WiFi APs shown alongside the names of the malls): four malls have average localization error of

(a) Two shopping malls as examples of our GMI test site.

(b) Three examples of inaccurate localization

Fig. 1. Screenshots taken on GMI.

around or above 20 meters, which can be hardly usable for indoor localization. Only test cases in ION exhibit reasonable errors: half of them are less than 10 meters. This is partially due to the smaller size of ION and hence a much denser WiFi deployment there.

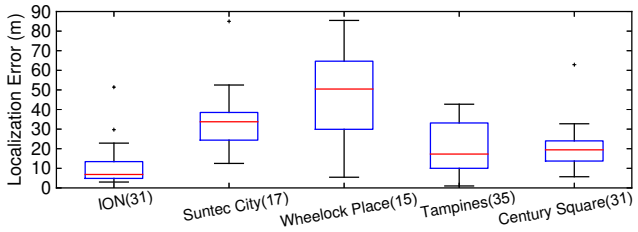


Fig. 2. GMI localization errors in five shopping malls.

We also use Fig. 3 to show the satisfactory level of users. A user is *satisfied* with a GMI location indicator if he feels that the indicator helps to locate himself (i.e., location errors within visual range is tolerable); otherwise *unsatisfied*. Obviously, the satisfactory levels are generally

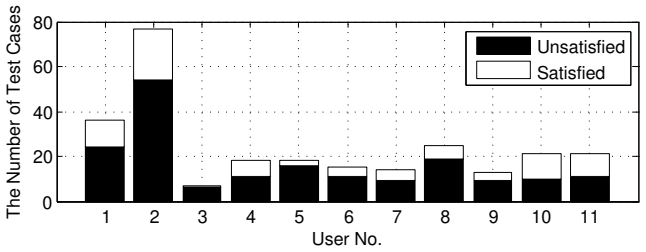


Fig. 3. The number of satisfied and unsatisfied cases for 11 users.

low. As mentioned before, out of the five shopping malls, ION has denser WiFi access points (APs) than others. So quite some satisfactory cases are obtained there.

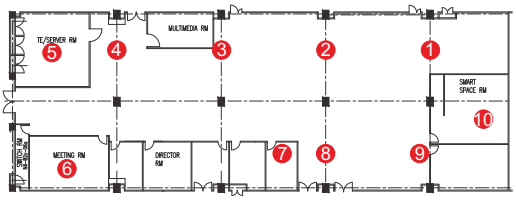
2.1.2 Navigation and WiFi Reliance

GMI's navigation function does not appear to be useful due to the unsatisfactory location accuracy (see the above discussions). Note that even if the initial location is satisfactory, a few unsatisfactory location estimations on the way may ruin the navigation.

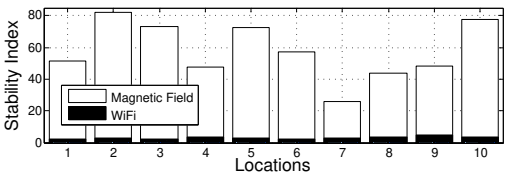
All our results have evidently confirmed GMI's heavy reliance on WiFi infrastructure: when either a phone's WiFi interface is switched off or the WiFi signals become very weak (in a basement level where WiFi hotspots are not installed), the GMI's location indicator is often expelled outside of the building, suggesting that some sort of cellular-based location estimation is applied.

2.2 Stability of WiFi Fingerprints

We suspect that the unsatisfactory performance of GMI in localization is due to its reliance on WiFi fingerprints. Therefore, we perform studies in our research center to compare the signal stability of WiFi with that of magnetic field (adopted by GROPING). We choose 10 locations in the 800 m^2 area shown in Fig. 4(a). At each location,



(a) Ten test locations in a research center of 800 m^2

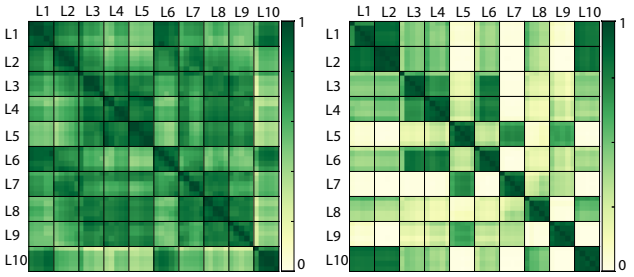


(b) Comparing mean values of stability index

Fig. 4. Signal stability comparison between WiFi and ambient magnetic field. The higher the index, the more stable a signal is.

we measure both WiFi RSSI vector (5 components with 5 hotspots around) and ambient magnetic field vector for 5 minutes, and we repeat this for ten rounds spreading over five different days. For each round, we compute the mean and standard deviation of the magnitude (or strength) of the vectors, and we use $\frac{\text{mean}}{\text{standard deviation}}$ as the *stability index* (a metric similar to Signal-to-Noise Ratio, or SNR). In Fig. 4(b), we compare, at different locations, the average stability indices (over ten rounds) of WiFi and magnetic field. It is obvious that even the most unstable case of the magnetic field is far better than that of WiFi.

We also use the confusion matrices in Fig. 5 to further illustrate the problem caused by WiFi instability. Due to instability of WiFi fingerprints, systems relying on WiFi have to accumulate a large amount of fingerprints for each location and then use their mean value to represent the location [23], [24]. As a result, the ability of differentiating among locations by WiFi fingerprints is apparently worse than that by magnetic fingerprints.

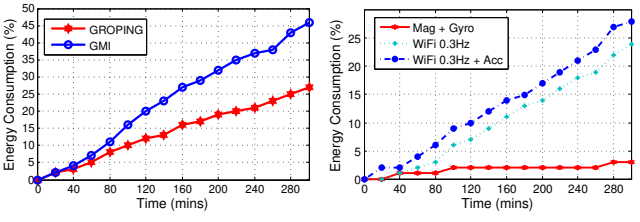


(a) WiFi RSSI (b) Magnetic Field

Fig. 5. Normalized confusion matrix of WiFi and Magnetic field signals (sampled in 5 meters).

2.3 Energy Efficiency Evaluation

We record the energy consumption (in terms of battery percentage by Android's battery meter) of our Samsung Galaxy S2 (with a 1650mAh battery) during the field studies on GMI and GROPING. During our lab tests, we further monitor the energy consumption for four configurations, namely idle (i.e., no sensor running), sampling magnetometer and gyroscope at 5Hz (GROPING), WiFi scanning at 0.3Hz, and lastly a combination of WiFi scanning and accelerometer sampling both at 0.3Hz. The sampling frequency of 0.3Hz comes from our observation that GMI updates its location estimation in about every three seconds. For all the tests, the 1650mAh battery is fully charged before continuously operating for 300 minutes, and the drop in battery life is recorded every 20 minutes. To focus on the energy consumption of sensing, we deduct the energy consumed under the idle configuration from all other configurations, and the results are shown in Fig. 6.



(a) Application consumptions (b) Sensing consumptions

Fig. 6. Energy consumption comparisons.

According to Fig. 6, GMI consumes much more energy than GROPING, but both consume more than pure sensing (possibly due to the use of 3G). Fig. 6(b) further shows that sensing configurations involving WiFi use 24% to 28% of the battery in 3 hours. As a sharp contrast, using two inertial sensors (as the case with GROPING) consumes only 3% of the battery for the same period, exhibiting roughly 10 times battery savings than other lab settings.

2.4 Summary

We summarize the key insights on GMI that motivate our designs of GROPING in the following:

- GMI's implementation of WiFi-based localization has not worked accurately yet. This may attribute to the instability of WiFi signals, sparse WiFi deployments, and insufficient fingerprints.
- GMI cannot be very helpful in indoor navigation due to unsatisfactory location accuracy, as navigation requires consistent location estimations.
- Localization over the whole map area is not helpful for navigation purpose, as errors in location estimation may render the user location off a pathway and hence reduce the chance of successful navigation.[1]
- High energy consumption is another major drawback of WiFi-based indoor localization systems, and this issue is exacerbated under navigation due to its need for constant location updates.

3 GROPING System Overview

GROPING provides services that caters to users' location and navigation requests in various indoor facilities, and it relies on the regular occupants of a certain indoor facility to assist in building floor maps. Basically, the end users include *map explorers* and *strayed users*. Map explorers are recruited due to their familiarity with a particular building. They walk along various pathways and upload their trajectories (consisting of sensed data) to the server. Strayed users are those who are unclear about their locations and hence require localization or/and navigation services. We illustrate the architecture of GROPING in Fig. 7. The system consists

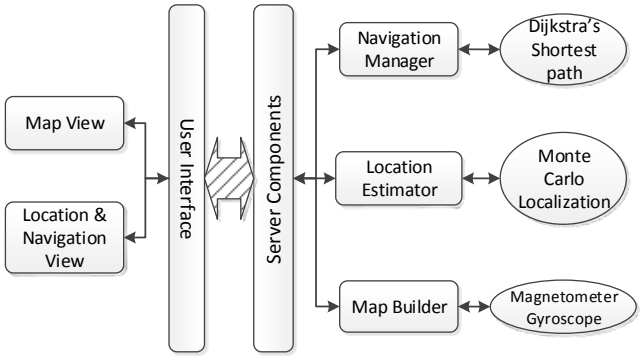


Fig. 7. GROPING system architecture.

of smartphone clients and a server. Each client provides a user interface for collecting data, as well as visualizing the constructed map, the current (estimated) location, and the navigation routes. The server is a cloud service; it consists of modules that build floor maps, estimate locations, and deliver real-time navigation. We shall briefly discuss these components in this section.

1. Google Maps (outdoor) only works for places where road systems come across.

3.1 Map Building

We hereby illustrate by an example how GROPING utilizes the contributions from map explorers to gradually build an indoor map. Alice and Bob are regular visitors to a shopping mall shown in Fig. 8. One day Alice installs GROPING but finds no map exists for the mall yet. She decides to create one by making the first contribution. Starting from position A, Alice walks toward an arbitrary direction and records the ambient magnetic field by her smartphone running GROPING. After walking for a while, she sees a three-way conjunction point B ahead of her. She could rely on the gyroscope in her phone to tag such a junction, but she may also choose to tag it manually (see Sec. 3.3.2). Such tags help GROPING to partition trajectory into segments. Eventually, Alice stops at junction E and uploads the trajectory data (top-right of Fig. 8) to the GROPING server.

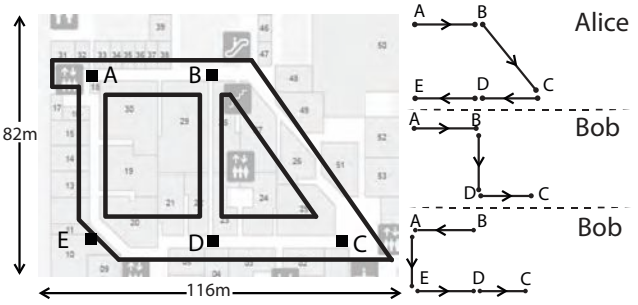
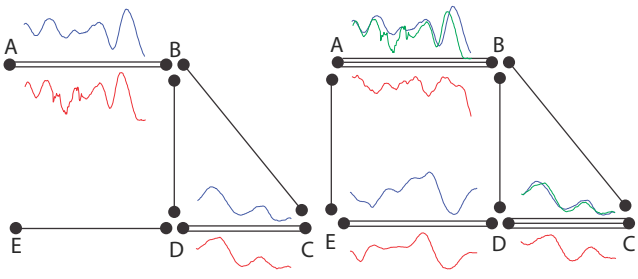


Fig. 8. Floor map of a shopping mall and the walking trajectories contributed by Alice and Bob.

The next day Bob comes to the same mall and finds the incomplete map contributed by Alice. So he decides to complete it, which first results in a trajectory shown in the mid-right of Fig. 8. GROPING server uses the similarity in magnetic fingerprints to infer the overlapping segments among the trajectories and sticks them together. After a few seconds, Bob receives a map (Fig. 9(a)) shown on his screen, waiting for him to either confirm or revert. Bob feels satisfied with the map, so he confirms and starts another trajectory recording procedure, which eventually end up with a complete map shown in Fig. 9(b).



(a) Sticking the first two trajectories (b) Completing the map

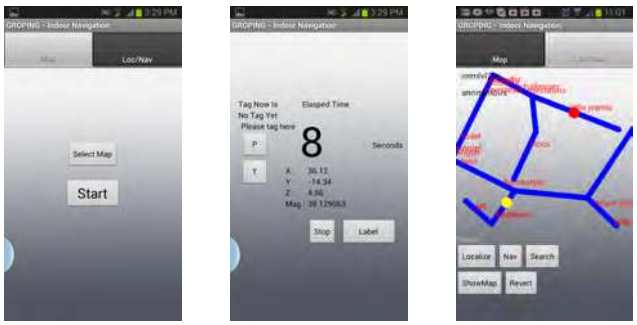
Fig. 9. Virtual map generation using three trajectories and the associated magnetic fingerprints.

3.2 Localization and Navigation

Based on the constructed map, the localization and navigation functions are integrated in GROPING, in the sense that user mobility facilitates localization that in turn drives further mobility (i.e., navigate a user). This is achieved by a revised Monte Carlo Localization (MCL) algorithm whose details are explained in Sec. 4.2. To continue our illustrating example, let us consider a strayed user Cindy. She starts her GROPING client, chooses the already constructed map, and requests a navigation service by either providing a semantic label (a shop name) or pinpointing a location on the map. Without knowing the initial location of Cindy, GROPING recommends a tentative route. While Cindy is walking along the route, her GROPING client keeps updating the sampled magnetic field information to the server. This allows GROPING to refine the location estimation for Cindy and also updates the route accordingly, until Cindy reaches her destination.

3.3 User Interface

A GROPING client has a simple interface as shown in Fig. 10. The starting screen, Fig. 10(a), requires each user to select a map. For a map explorer, one may choose to build a new map or to reinforce an existing map. For a strayed user, one needs to choose a map from a list. As the GSM location information is attached to every map when it is first built, the map list shown to a user is confined to the region close to the user's estimated location by GSM and is sorted in increasing distances. As the GSM localization is just an ancillary function and it runs only when GROPING starts up or a user switches to a different building, the incurred overhead is negligible.



(a) Starting screen (b) Map panel (c) Loc/Nav panel

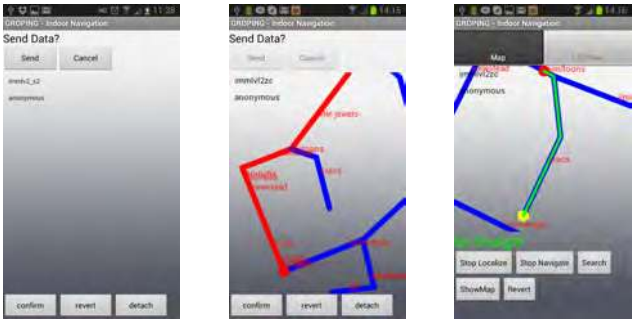
Fig. 10. The user interface of a GROPING client.

After choosing an existing map or initiating a new one, the "Start" button allows an explorer to start data collection and map generation, as shown by Fig. 10(b). Otherwise a strayed user may switch to the "Loc/Nav" panel to find his/her location and/or to obtain navigation guidance towards a certain destination, as shown by Fig. 10(c). We provide more details on these two panels in the following.

3.3.1 Map Panel

Map view allows explorers to collect trajectory data and upload them for map construction. The data collected along each walking trajectory includes both magnetic fingerprints and gyroscope readings. While GROPING uses the fingerprints to represent individual pathways, it also exploits the gyroscope readings to identify turns. All together, these data help the server to assemble a floor map. While showing the instant sensor readings, this panel also offers two tags P and T for an explorer to complement the sensing procedure with his/her perception. In particular, when the explorer passes a conjunction, the T could be optionally pressed, then the tag P should be pressed upon returning to a pathway.

In case of encountering any interesting landmark, the explorer can input the description of the landmark using the Label button. The sensing procedure is suspended when the explorer inputs the landmark label and it automatically resumes after. These landmarks are stored in a map library (residing in the server) as semantic labels for the benefit of semantic navigation. Pressing the Stop button invokes another panel, Fig. 11(a), suggesting to either upload collected data to the server or cancel them. Upon uploading, the constructed map is presented to the explorer for judging whether it is satisfying. If the explorer observes any issues with the newly constructed map, he/she can revert the map to the previous state. Fig. 11(b) shows the constructed map annotated with the landmarks labeled by explorers.



(a) Upload screen (b) Updated map (c) Nav route

Fig. 11. More about GROPING client user interface.

3.3.2 Navigation and Localization Panel

This panel first presents the location of the user with a yellow dot on the selected floor map. This location may not be accurate, but the server will gradually refine it after the user starts to move. If the user chooses a destination (red dot), the navigation route to the destination from current location is depicted in green color on the map, shown in Fig. 11(c). To define a destination, the user can either pinpoint it on the map or perform a semantic label searching. Label searching may cause all related labels being highlighted for further selection. For example, when a user searches for "cafe", all labels containing "cafe" will be highlighted. The navigation route is computed as the shortest path between the current location and the chosen destination. Because the current location can be updated by the server (especially at the beginning), the route may experience some changes initially but should stabilize soon. If a user diverts from the specified route

due to missing a correct turning point, a new route will be highlighted accordingly. To better assist the navigation, walking instructions such as "go straight" and "turn right" are given either regularly or before a certain event.

Remarks: As illustrated by Fig. 10(c) and 11(c), our map differs significantly from those of GMI. This follows from the rationale that only the "road system" is necessary for navigation, adding other components such as rooms or cubicles may confuse users, given that the location estimation cannot be perfect. Our GROPING is meant to be a navigation service, so it may not provide a comprehensive localization function over the whole floor, as promised by GMI and other existing proposals [23], [24], [32]. Therefore, GROPING is rather a complement to the existing WiFi or dead reckoning based localization systems than a competitor to them, and it can be combined with other systems to perform both lightweight navigation and accurate localization.

4 System Components

In this section, we dive into the technical details of the three components comprising GROPING: map builder, location estimator, and navigation.

4.1 Map Builder: A Joint Venture of The Crowd

Map builder is the most unique part for GROPING compared with the existing indoor localization literature [33], [23], [24], [36], [19] (where a known map is always assumed). The principle behind GROPING map builder is that, when a certain number of explorers walk indoors, there is a high possibility of their trajectories overlapping. Merging these overlapping trajectories results in a floor map that comprises of only the indoor route structure. This simplifies the information content (pertaining to the floor map), making it easy for users to follow the map. Moreover, each map is enriched with semantic information, i.e., landmarks provided as labels, to facilitate navigation.

4.1.1 Virtual Map Terminologies

We consider three components of a floor plan, namely hallway, conjunction points, and semantic labels. As shown in Fig. 12, the blue areas are hallways, the red area is

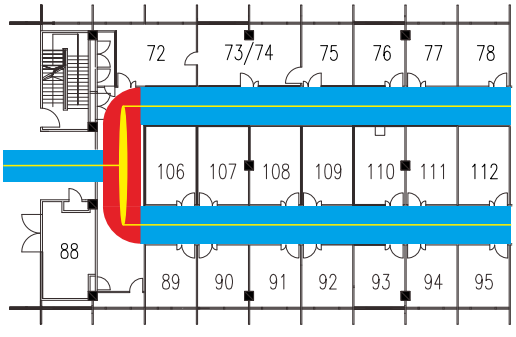


Fig. 12. Three components of a floor plan.

a conjunction point, and the numbered blank spaces are semantic labels attached to hallways. The objective of GROPING virtual map generation is to re-construct the map to the extent as illustrated by the yellow skeleton, using sensor data collected by the users. Based on the idea of crowdsensing, we let a group of users to arbitrarily pick up walking trajectories and use their smartphones to collect sensor data while walking. To endow the map with semantics, each user is supposed to *label* a couple of rooms (by names or numbers) along each trajectory.

We define a virtual map M as one that contains route structure information, semantic labels l, and magnetic fingerprints F. Route structure information include segments (pathways), conjunctions/linkages between segments, and time spent on each segment. Semantic labels are stored as texts but are associated with respective locations in terms of segment percentages. Fingerprints of a segment are the magnetic field signals collected along that segment. Multiple fingerprints from different trajectories are allowed to be associated with the same (overlapping) segment. In particular, the map library $\mathcal{M}$ contains a set of virtual maps $\{M_1, M_2, ..., M_k\}$, where each $M_i = \{C_i, E_i\}$ is represented as a graph with vertex set C_i and edge set E_i, with each vertex $c_{ij} \in C_i$ indicating a conjunction and each edge $e_{ij} \in E_i$ representing a segment. Moreover, each c_{ij} is associated with a set of angles (obtained from gyroscope readings), and each edge e_{ij} is associated with a set of fingerprints $\{F_{ij}^1, F_{ij}^2, \cdots, F_{ij}^n\}$ and a set of labels $\{l_{ij}^1, l_{ij}^2, \cdots, l_{ij}^m\}$. We explain in the following the three steps taken by GROPING to form a map.

4.1.2 Trajectory Segmentation

To identify hallways, we need to partition a user's walking trajectory (represented by the sensor data collected on the way) into segments. This is done by two approaches. In the first approach, we integrate the gyroscope reading g_y within a sliding window $W_{\text{turn}} = 5$ seconds. If the value goes beyond a threshold (20 degree in our setting), a conjunction point is detected, as shown in Fig. 13. The total turning

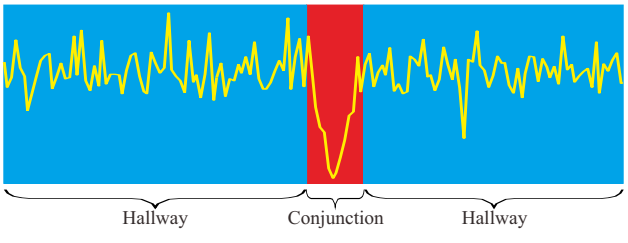


Fig. 13. Segmenting a trajectory using gyroscope reading g_y.

angle is estimated by gradually enlarging the integration window until the result of integration stops increasing. As a result, the corresponding data segment is marked as T (i.e., turning point) and the total turning angle becomes the fingerprint associated with this segment. The second approach explores the human sensing ability, which we term *tagging*. Specifically, users manually tag the sensor

data with T upon a conjunction and then tag P on the data upon returning to a hallway (see Sec. 3.3.1).

The first approach is automatic without the need for human intervention, but it fails to detect a conjunction point if the user goes straightly through it. The second approach works perfectly if a user remembers to tag all conjunctions. In practice, both approaches work fine if we have sufficient number of trajectories, as we can anyway drop those containing conjunctions that we fail to detect due to either a straight going through or a user's oblivion of tagging. We mainly use the first approach but optionally augmented by the second one. At the end of the segmentation phase, each trajectory T consists of at least one segment. A hallway segment (marked as P) contains magnetic fingerprints of the corresponding hallway, and a conjunction segment (marked as T) is associated with its angle. Also, each segment is sporadically labeled with room numbers or names.

4.1.3 Segment Matching

GROPING makes use of the overlaps between trajectories to stitch them together. Given a sufficient amount of trajectories that cover the whole floor and that overlap with each other, the skeleton of the floor map can be re-generated. To this end, we need to identify overlapping segments of an arbitrary pair of trajectories. However, the segments of magnetic fingerprints can be time misaligned since the walking speeds vary across users collecting data. Therefore, we use the DTW algorithm [27] to compute the similarity. DTW is well known to handle sequences that follow a similar trend but vary across the time axis. The main idea behind DTW is to compress or stretch the time axis of one (or both) sequences for getting a better alignment.

Consider two segments of magnetic fingerprints, $F_1 = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_K\}$ and $F_2 = \{\boldsymbol{f}'_1, \boldsymbol{f}'_2, \cdots, \boldsymbol{f}'_L\}$. The goal is to find the best match between these two segments by an alignment w^* called *optimal warping path*. A warping path is given by $w = w(1), w(2), ..., w(N)$, in which $w(n) = [i(n), j(n)]$ is the set of matched samples, where $i(n)$ and $j(n)$ belong to the index sets of F_1 and F_2, respectively. The optimal warping path w^* minimizes the overall cost function given by $\sum_{n=1}^{N} \delta\left(w(n)\right)$, where $\delta(w(n))$ is the distance measure computed using the inverse of cosine similarity given as:

$$\delta\left(i(n), j(n)\right) = \cos^{-1}\left(\frac{\boldsymbol{f}_{i(n)} \cdot \boldsymbol{f}'_{j(n)}}{\|\boldsymbol{f}_{i(n)}\| \|\boldsymbol{f}'_{j(n)}\|}\right). \qquad (1)$$

Given a pair of segments, their minimized cost function $\sum_{n=1}^{N} \delta\left(w^*(n)\right)$ characterizes their similarity: a lower function value indicates a higher similarity.

4.1.4 Map Formation

One major difference between our map formation and photo stitching is that we face a much more complicated topology: topologies involved in photo stitching often contain no loop. Our idea is to start the map from a single trajectory, then sequentially invoke **Algorithm 1** to gradually stitch incoming trajectories to the existing map.

Algorithm 1: Trajectory Stitching

Input: New trajectory T, current map $M = \{C, E\}$

1 **foreach** $c \in C$ **do** $P_{\text{temp}}(c) \leftarrow 0$; $P_{\text{coin}}(c) \leftarrow 0$
2 $s \leftarrow T.firstSeg$
3 **while** $s \neq$ NULL **do**
4 **foreach** $s_f \in E$ **do**
5 **if** $s.tag = \mathsf{T} \ \wedge \ s_f.tag = \mathsf{T}$ **then**
6 $s_p \leftarrow s_f.prevSeg$; $p_{\text{coin}} \leftarrow P_{\text{coin}}(s_p.bEnd)$
7 $P_{\text{temp}}(s_f.bEnd) \propto simA(s, s_f) \times p_{\text{coin}}$
8 $s'_f \leftarrow reverseSeg(s_f)$
9 $s_p \leftarrow s'_f.prevSeg$; $p_{\text{coin}} \leftarrow P_{\text{coin}}(s_p.bEnd)$
10 $P_{\text{temp}}(s'_f.bEnd) \propto simA(s, s'_f) \times p_{\text{coin}}$
11 **else if** $s.tag = \mathsf{P} \ \wedge \ s_f.tag = \mathsf{P}$ **then**
12 **foreach** $s_p \in s_f.prevSeg$ **do**
13 $p_{\text{coin}} \leftarrow p_{\text{coin}} + P_{\text{coin}}(s_p.bEnd)$
14 $P_{\text{temp}}(s_f.bEnd) \propto simM(s, s_f) \times p_{\text{coin}}$
15 $s'_f \leftarrow reverseSeg(s_f)$
16 **foreach** $s_p \in s'_f.prevSeg$ **do**
17 $p_{\text{coin}} \leftarrow p_{\text{coin}} + P_{\text{coin}}(s_p.bEnd)$
18 $P_{\text{temp}}(s'_f.bEnd) \propto simM(s, s'_f) \times p_{\text{coin}}$
19 $[c_{\max}, p_{\max}] \leftarrow maxProb(P_{\text{temp}})$
20 **if** $p_{\max} >$ defiThreshold **then**
21 $mergeSeg(s, c_{\max}.endSeg)$; $backBProp(T, s)$
22 **if** $formLoop(M)$ **then** $relaxLoop(M)$
23 $P_{\text{coin}} \leftarrow P_{\text{temp}}$; $s \leftarrow s.nextSeg$

The stitching process is based on Bayes filter [10]. It associates with an end point, $s_f.bEnd$, of $s_f \in E$ (an existing segment) the probability P_{coin} of s_f coinciding with an incoming segment $s \in T$, and it keeps updating P_{coin} while scanning sequentially through all segments in T (lines 2 to 23). For a conjunction segment s_f, the probability is updated according to the similarity in angle between s and s_f (computed by $simA(s, s_f)$, a function of the absolute difference between the two angles) multiplied by the probability associated with the hallway segment preceding s_f (lines 7 and 10), where we use $\propto$ to indicate that probabilities are to be normalized to satisfy unitarity. The situation is slightly more complicated for a hallway segment s_f, as all the preceding (conjunction) segments should be counted (lines 12 and 16). The similarity evaluation done by $simM(s, s_f)$ follows what was discussed in Section 4.1.3. For both cases,the similarity should be computed from both directions (lines 8 and 15). Since the first segment has no preceding one, we bootstrap it with a small probability.

If a certain coincidence probability (associated with an end point $v_{\max}$) becomes larger than defiThreshold (i.e., *definiteness threshold*) (lines 19 and 20), $s \in T$ is merged with the segment in M whose end point is $v_{\max}$, and a backward belief propagation $backBProp(T, s)$ is applied to trace back the stitching history such that all the previous segments are properly merged into M (line 21). If a loop is formed after stitching, $relaxLoop(M)$ is invoked to adjust

the geometry of the loop such that the graph M can be embedded into a 2D plane.

The resulting map M, on one hand, contains all the fingerprints that have been collected, and each segment is associated with a set of fingerprints collected from the corresponding hallway or conjunction. On the other hand, it has the same topology as the original floor map, as well as a similar geometry (the length of a hallway can be estimated by the number of points contained in a corresponding segment fingerprint). In fact, the index of a sample point in a segment fingerprint also indicates a rough location on the corresponding hallway. For example, if a point is the 100-th point out of 1000 samples of a segment, then the location is at the 10% length of the whole hallway. Of course, indicating location in this way may lead to error, but it is within the tolerable range of the applications targeted by GROPING. Later in Section 4.2.1, we shall abuse the terminology by using $\boldsymbol{l} \in M$ to denote that $\boldsymbol{l}$ belongs to the index set of the sample points in M; in other words, $\boldsymbol{l}$ is a location on our virtual map M.

4.2 Location Estimator: A Bayesian Approach

The basis of GROPING's location estimator is a classification process similar to other WiFi-based localization schemes (e.g., [15], [32]), where a user's sensor data are compared with the existing fingerprints to obtain a list of similarity indices, and the location is suggested by the highest similarity index. However, the ambient magnetic field that we rely on offers less information than the WiFi-based infrastructure: the former is just a 3D vector field (magnetic field strength in X, Y, and Z directions) but the latter, given a sufficient amount of available WiFi hotspots, may produce fingerprints in a much higher dimensional space. As a result, we have to resort to a filtering technique that spans the temporal dimension to gain more information for achieving a sufficiently accurate location estimation.

4.2.1 Revised Monte Carlo Localization

To involve the temporal dimension, a sequential estimation technique is needed. This motivates us to use the *Monte Carlo Localization* (MCL) approach [31]. Under a Bayesian framework, MCL recursively computes the posterior distribution of the location $\boldsymbol{l}_t$ (a.k.a. *belief*) $B(\boldsymbol{l}_t) = p(\boldsymbol{l}_t|\boldsymbol{m}_{1:t})$ at time t, considering different measurements $\boldsymbol{m}_{1:t}$ (collected sensor readings) from time 1 up to time t. We briefly walk through the algorithm below, while emphasizing on our revisions. Using the Bayes rule, we have

$$B(\boldsymbol{l}_t) = \gamma p(\boldsymbol{m}_t|\boldsymbol{l}_t)p(\boldsymbol{l}_t), \tag{2}$$

where γ is the normalizing constant.

While a user keeps walking (and collecting new sensor readings), the belief is recursively updated as follows:

$$B(\boldsymbol{l}_t) = \gamma p(\boldsymbol{m}_t|\boldsymbol{l}_t) \sum_{\boldsymbol{l}_{t-1} \in M} p(\boldsymbol{l}_t|\boldsymbol{l}_{t-1})B(\boldsymbol{l}_{t-1}), \tag{3}$$

where M refers to the virtual map that we build using the techniques presented in Section 4.1.4.

After a certain period t, an MLE estimator is applied to select the location with the highest posterior probability, giving a location estimation:

$$\hat{\boldsymbol{l}} = \underset{\boldsymbol{l}_t \in M}{\operatorname{argmax}}[B(\boldsymbol{l}_t)]. \tag{4}$$

In general, a larger t leads to a higher estimation accuracy. However, as we shall show in Section 6.3, the accuracy is sufficiently high after only a few tens of seconds. To implement (3), we need both $p(\boldsymbol{m}_t|\boldsymbol{l}_t)$ (observation model) and $p(\boldsymbol{l}_t|\boldsymbol{l}_{t-1})$ (motion model). In the following, we discuss how we tailor these two models to accommodate the features of GROPING. As both models are time-invariant, we drop the subscript t hereafter.

Observation Model: We evaluate $p(\boldsymbol{m}|\boldsymbol{l})$, the observation model, in the following way. For a new measurement $\boldsymbol{m}$, we compare it with all sample points in M. This comparison is again based on the cosine similarity between a sample point $\boldsymbol{s} \in M$ and $\boldsymbol{m}$. We have a few sample points sharing the same index (i.e., at the same location $\boldsymbol{l} \in M$), as a result of the clustering procedure explained in Sec. 4.2.2). So we take the maximum cosine similarity value to build the observation model:

$$p(\boldsymbol{m}|\boldsymbol{l}) \propto \max_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{l}}} \cos(\boldsymbol{s}, \boldsymbol{m}), \tag{5}$$

where $\mathcal{S}_{\boldsymbol{l}}$ is the set of sample points indexed by $\boldsymbol{l}$. The operator $\propto$ is again for normalization purpose.

In fact, what estimated by $\max_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{l}}} \cos(\boldsymbol{s}, \boldsymbol{m})$ is rather $p(\boldsymbol{l}|\boldsymbol{m})$. However, according to Bayes rule, $p(\boldsymbol{m}|\boldsymbol{l}) \propto p(\boldsymbol{l}|\boldsymbol{m})$ if we assume non-informative priors for $\boldsymbol{l}$ and $\boldsymbol{m}$ (i.e., $p(\boldsymbol{l})$ and $p(\boldsymbol{m})$ both follow a uniform distribution).

Motion Model: The motion model is represented by a Markov transition matrix, in which nearest locations in both directions from the current location have non-zero transition probabilities, and all other probabilities are zero. In the most ideal case (where the walking speed of the current user coincides with that implied by the normalized length of the segment fingerprints), only two transitions are possible: forward and backward, shown by an example within one segment as follows:

$$p(\boldsymbol{l}^+ \mid \boldsymbol{l}) = \begin{array}{c} \\ \ell_1 \\ \ell_2 \\ \ell_3 \\ \vdots \\ \ell_n \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccc} \ell_1 & \ell_2 & \ell_3 & \cdots & \ell_n & \cdots \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & \cdots \\ 0.5 & 0 & 0.5 & \cdots & 0 & \cdots \\ 0 & 0.5 & 0 & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 0 & 0 & 0 & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \end{array},$$

where $\ell_i \in M$ are indices of sample points. In our implementation, we assume that the actual walking speed of a user can be at most α ($\leq$ 2) times faster than the normalized one. Therefore, there might be up to 2α possible transitions from each location. Our motion model differs from the traditional one assuming a continuous transition distribution, simply because the virtual map M is a discretized version of the original map.

There is yet another issue that we need to handle before proceeding to actual localization. As incoming segment fingerprints increase linearly with user participation, this tends to increase the complexity of location estimation, since more and more fingerprints need to be compared against the newly sampled sensor data. To this end, we apply a clustering algorithm to obtain the representatives among the fingerprints for a given segment in the following.

4.2.2 *Clustering with Affinity Propagation*

Although we have demonstrated in Sec. 2.2 that the ambient magnetic field is very stable (far more stable than WiFi RSSI), different phone models may still obtain different (albeit involving similar features) readings, as shown in Fig. 14. However, if we kept all the data associated with

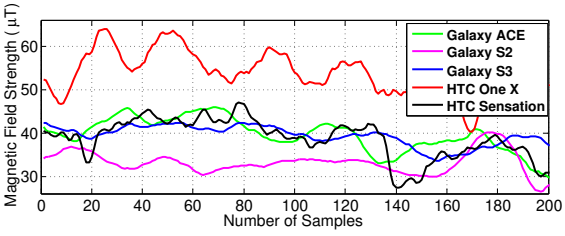


Fig. 14. The same ambient magnetic field sensed by different smartphones.

a segment as its fingerprints, the complexity of location estimation would keep increasing. Our idea here is to classify the fingerprints for each segment, and choose one fingerprint for each cluster to represent it. The outcome is that only a few fingerprints need to be compared during the location estimation procedure.

Obviously, typical clustering algorithms such as k-means do not work, as we do not know k a priori, and those algorithms may not return existing values in a data set. Therefore, we apply the *Affinity Propagation* (AP) algorithm [11] to obtain a few representatives out of the fingerprint set. AP is a message passing algorithm, where the magnitude of each message passed showcases the current belief or affinity one data point (segment fingerprint in our case) has for choosing another data point as its exemplar among a set of points pertaining to a particular cluster. AP does not assume a priori knowledge of k, i.e., the number of clusters. It proceeds iteratively using a similarity matrix containing the similarity score between each pair of fingerprints and updated by the messages passed.

In Fig. 15, the background (light blue curves) shows about 100 fingerprints associated with a certain hallway segment, whereas the foreground (red curves) are the seven representatives chosen by AP. This significantly reduces the complexity of executing (5). Note that we cannot use DTW to compute the similarity scores, as the outcome of DTW is not a metric. Therefore, we first apply the DTW-based time-normalization procedure [30], in order to normalize all fingerprints associated with a certain segment to the same length (the median length) and variance. Then the

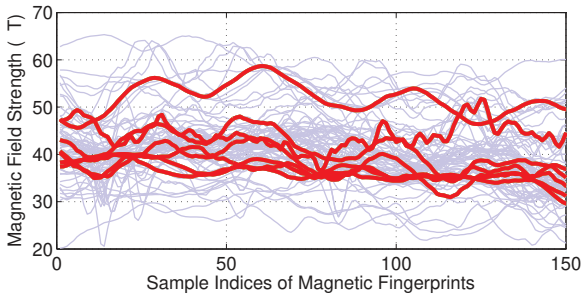


Fig. 15. The outcome of AP clustering.

similarity scores are computed as the negative Euclidean distances among these normalized fingerprints.

4.3 Navigation: Client–Server Interactions

We first sketch the client-side navigation by pseudo-codes in **Algorithm 2**. While navigating, the client periodically queries location from the server (line 2). If the current location is sufficiently close to the destination, the navigation is completed (line 3). Otherwise if the current location is off the route, a new route is queried from the server (line 5). At the end of each round, the route is rendered on the map and certain instructions are also shown. The client-side of GROPING only performs simple computations, while the heavy computations are offloaded to the server-side.

Algorithm 2: Client-side Indoor Navigation

Input: Destination point d, Map M

1 **while** $navigationOn$ **do**
2 $\quad c \leftarrow currentLocation()$
3 $\quad$ **if** $inRange(c, d)$ **then break**
4 $\quad$ **if** $notOnRoute(c, rt)$ **then**
5 $\quad\quad rt^{+} \leftarrow requestNavigation(d, c, rt, M_i)$
6 $\quad renderRoute(rt^{+})$; $showInstruction(c, rt^{+})$

Given a destination chosen by a user and the current location returned by the location estimator, the navigation manager (server-side) calculates the route to destination on a map and provides continuous instructions. As we have discussed, give a map $M_i = \{C_i, E_i\}$, we use the average number of sample points for the fingerprints associated with a segment in E_i to roughly represent the length of that segment. This allows the server to compute a shortest path on M_i from the current location to the destination.

As shown by **Algorithm 3**, given the current location c and destination d, the server treats two segments (the segment containing the current location e_c and that containing the destination e_d) differently. Basically, the server divides e_c into e_c^1 and e_c^2 by the current position r_c with respect to e_c, and e_d into e_d^1 and e_d^2 similarly. The length of the new edges are assigned proportionally, and the new (temporary) map M_i' is fed to the Dijkstra's algorithm to compute the shortest path between the current location and destination. In practice, a slight preference will be given to the current walking direction of the user. For each

Algorithm 3: Server-side Indoor Navigation

upon $recvNavRequest(c, d, rt, M_i)$
$(e_c, p_c) \leftarrow pointInMap(c, M_i)$
$(e_d, p_d) \leftarrow pointInMap(d, M_i)$
$M_i.removeEdge(\{e_d, e_c\})$
$(e_c^1, e_c^2) \leftarrow e_c.breakAt(p_c)$; $(e_d^1, e_d^2) \leftarrow e_d.breakAt(p_d)$
$M_i.addEdge(\{e_c^1, e_c^2, e_d^1, e_d^2\})$
$rt^+ \leftarrow Dijkstra(p_c, p_d, M_i)$
if $rt^+ \subseteq rt$ **then** $deliverToClient(\text{NULL})$
else $deliverToClient(rt^+)$

navigation request coming from a client, a shortest path to destination is calculated by navigation manager using **Algorithm 3** and is then delivered to client. Instead of sending only one navigation request at the beginning, the client actually generates such a request on a regular basis. However, the server does not send a new route back as long as the user's location is still on the previously determined route; a route update is sent back only if the server finds out that the user is off the previously determined route.

5 System Evaluation Design

We briefly explain how we perform user studies and performance evaluations on GROPING in this section, and also discuss some issues we have encountered for the user studies, as well as our current and future solutions.

5.1 Experiment Setting

We recruited 20 users to participate in our user study and evaluations. As our emphasis is rather on the functionalities of GROPING than on the usability of its interface, we only involve participants with CS background, but they are all first-time users of GROPING. Eight of them were selected to play the role of map explorers due to their familiarity with the test site, and the rest were strayed users. Their specific tasks include familiarizing with GROPING interface, collecting sensor data, labeling landmarks, and providing feedback. While walking, a user is required to hold the phone horizontally and point it ahead.

We have evaluated GROPING by three studies mainly in one test site (a shopping mall with 3 floors). In the first one, we measure the time needed to complete a map in each floor. In the second one, we focus on quantitative evaluations on the accuracy of the localization service. Finally, we qualitatively study the navigation service, and report the user experiences on it. We also implemented FreeLoc [35], a recently proposed WiFi-based indoor localization system, and we compare the localization accuracy of GROPING with both FreeLoc and GMI. Further comparisons with two canonical proposals RADAR [5] and Horus [37] are conducted in the smaller test site shown in Fig. 4(a), as these proposals entail intensive WiFi fingerprinting. Due to the page limit, we have postponed some detailed evaluations to the Appendix.

As user acceptance is important to a navigation system [4], we try to understand user preferences before and experiences after using our navigation system, by employing user feedback. To understand user expectations, we first conduct a questionnaire-based on-line study about the users experiences of getting directions inside large buildings without a navigation service and on what they expect from a navigation system (in which we involve extra participants through Amazon Mechanical Turk (AMT) [1]). The user experience is the outcome of the aforementioned third study: each of our 20 (local) participants delivers a feedback on GROPING after using it.

5.2 Incentives for Crowdsensing

Getting a sufficient number and diversity of participants for our user studies has been a challenge, because one of the main tasks that we assign to our users is mobile crowdsensing (for map generation) using their individual smartphones. Crowdsensing data collection differs significantly from traditional crowdsourcing since it demands individuals' utilizing of time, energy (e.g., physical activities) and resource (e.g., smartphone usages), so the incentive to "entice" participants into providing high quality data may need to be very substantial. In other words, some form of remuneration is necessary to encourage active participation in crowdsensing. Incentive mechanisms are often task dependent and can range from monetary incentives (cash, lottery tickets, gift cards, etc.) to valuable services (e.g., free WiFi access or storage spaces) [26], [22].

6 Experiment Results

Beside the comparisons made in Sec. 2, we further evaluate GROPING in this section.

6.1 Why GROPING is Needed

To show why a portable indoor navigation solution is needed, we design a questionnaire survey about people's indoor experience. The survey is done in two groups. The first group includes our 20 participants, and the 118 participants of the second group are involved by extending the survey to AMT [1] and the questions are raised towards a familiar mall. Table 1 shows the answers to the first five questions and Fig. 16 shows the outcome of the last question. Because the first group is restricted to choose

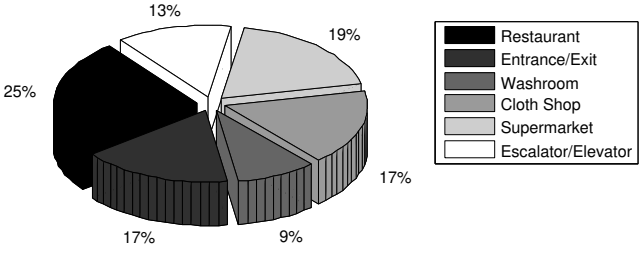


Fig. 16. Types of landmarks remembered by people.

our test site and the second group can choose any familiar

S/N	Question	Our Test Site	General
1	How many times have you been to the mall within last year	10.61 ± 8.74	18.07 ± 11.04
2	How familiar are you with the indoor space (1-10)	5.11 ± 2.80	7.11 ± 2.17
3	How easy it is to navigate to a particular store based on you-are-here maps (1-10)	5 ± 1.43	7.03 ± 2.21
4	How helpful it will be to have a smartphone-based indoor navigation system (1-4)	3.5 ± 0.67	2.7 ± 0.77
5	How many landmarks do you usually recall every time you enter the mall	5 ± 2.66	4.27 ± 1.55

TABLE 1
User perceptions on indoor navigation solutions (average $\pm$ standard deviation).

shopping mall, participants of the second group show more confidence in navigating by you-are-here maps than those of the first group. Also, as our test site has a more complicated route structure (see Fig. 17), it is reasonable that the first group expresses more eagerness to have a portable navigation system. In fact, both groups can recall on average less than five locations, so a handy navigation system may always help to avoid finding/checking you-are-here maps. The outcome shown in Fig. 16 has independent interests. Although each participant only remembers about five landmarks, those landmarks are well spread across different types. In other words, by asking map explorers to sporadically label landmarks, there is a fairly good chance that the labels would cover diverse landmarks in a mall.

6.2 Efficiency of GROPING Map Construction

We perform a field study in a shopping mall with three floors shown in Fig. 17 (top row). The most intriguing aspect of this mall is its complicated indoor route structure, which makes indoor navigation an actual necessity (most of our participants often get confused whenever they enter this mall). We also show the constructed map as the screenshots on our phone in Fig. 17 (bottom row).

Fig. 17. Floor maps and the corresponding GROPING maps of 3-floor shopping mall.

We summarize the time needed to complete the map construction for individual floors with different numbers of explorers in Table 2. A map is completely constructed if the topology of the route system is fully captured; we do not count the time to completely label all shops. The results in Table 2 show that GROPING can construct a rather complicated map in less than one hour. According to Singapore Straits Times www.straitstimes.com, Google needs a couple of weeks to furbish the map contributed by a venue before it can be used by GMI.

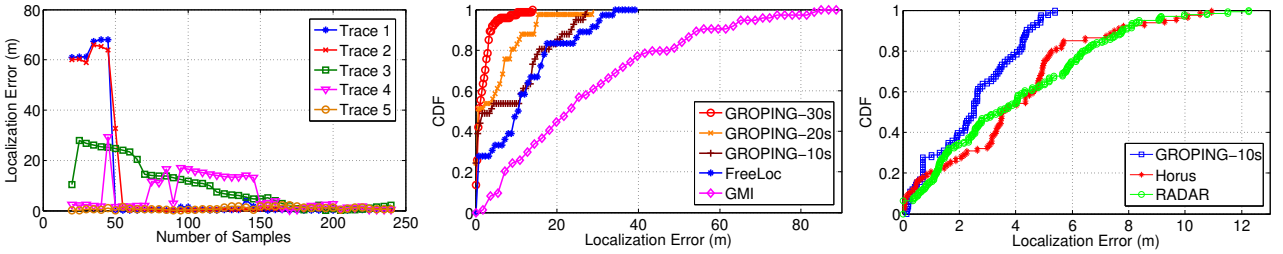
	8 explorers	4 explorers	2 explorers
Floor 1	14 minutes	34 minutes	1.5 hours
Floor 2	12 minutes	24 minutes	1 hour
Floor 3	5 minutes	15 minutes	45 minutes

TABLE 2
Map construction times with different number of explorers.

6.3 Accuracy of GROPING Location Estimation

As GROPING needs to first find out the current location of a strayed user before being able to navigate him/her, a sufficient accuracy in localization is very important. In Fig. 18, we report the statistics of the data accumulated during our field studies on the GROPING navigation service (reported later). Fig. 18(a) shows the localization errors as a function of the number of samples (i.e., the time a user spends on walking). Five exemplar traces were obtained by different users from five distinct locations, and the GROPING location estimator starts to report location only after 20 samples (4 seconds). We can see that initially the errors can be large but approximately after 150 samples (30 seconds), the algorithm converges with errors less than 5 meters. There are also cases where the initial 20 samples are sufficient to obtain accurate location estimations.

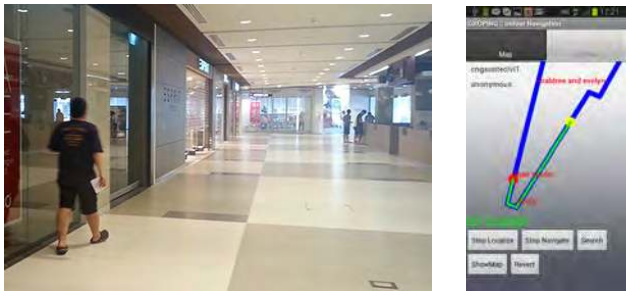
Fig. 18(b) depicts the distribution of location errors for all our experiments. It shows that after 30 seconds, 90% of the errors are within 5 meters, coinciding well with Fig. 18(a). Only a very small fraction is between 10 to 15 meters. As GROPING is a navigation service, such an accuracy is sufficient and the sporadic large errors can be visually corrected, because two adjacent units could well be spaced anywhere between 5 to 15 meters in large scale entertaining facilities. We further compare GROPING with GMI and our implementation of FreeLoc [35] in Fig. 18(b). While GMI always performs the worst, FreeLoc shows comparable accuracy with GROPING in the first 10 seconds, but much lower when it comes to 20 seconds.



(a) Convergence of localization (b) CDFs of location errors (c) CDFs of location errors

Fig. 18. Localization accuracy of GROPING. Comparisons are made in a large test site (b) and a small one (c).

As the comparisons with RADAR [5] and Horus [37] are done in a different test site, we present them separately in Fig. 18(c). While all the three systems achieve very good localization accuracy due to the small area of the test site, GROPING still outperforms its competitors. In addition, GROPING has an accuracy similar to what are claimed in [23], [24], [19], without the need for a WiFi infrastructure or a map. Although UnLoc [32] performs better than GROPING, GROPING, using only two inertial sensors and requiring far less user interventions, is a lightweight system consuming much lower energy, as we have discussed earlier.



(a) Walking under navigation (b) Screenshot

Fig. 19. Snapshots of the GROPING navigation.

6.4 User Feedback on GROPING Navigation

After the maps were constructed, we let the remaining 12 participants (except the 8 map explorers) to install the GROPING client on their own smartphones (which include Samsung Galaxy S2/S3/Ace Plus, Sony Xperia S, and HTC One X). The whole evaluation process has lasted for several weeks with participants visiting our test site sporadically and performing hundreds of tests (each test involving an arbitrary source-destination pair). In Fig. 19, we show a participant walking under the navigation guidance, as well as the screenshot of his phone at that moment.

The feedback provided by a participant after each test included two points:

1) Was the navigation process successful or not?
2) A mark on the satisfactory level.

A navigation succeeds if it guides a participant to be within the visual range of the destination in 10 minutes. Participants all give a mark (1 to 10) on the satisfactory level to represent their navigation experience, and they also provide us with comments to explain their marks.

The outcome shows that *all* the tests ended up successfully, and Fig. 20 illustrates the distribution of the satisfactory levels. Apparently, users are rather satisfied with their navigation experiences. There are a few cases where the satisfactory level falls between 1 to 6, which are often caused by the initial "jumping" of the current locations and also by our rudimentary user interface that does not allow map rotation. While the interface issue can be easily handled, our temporary solution for preventing location jumping is to delay the display of the current location (hence the navigation route). However, users may still feel unsatisfied as they have to walk "blindly" for tens of seconds. In our future work, we could combine a (one-time) WiFi-based localization with GROPING, such that GROPING may quickly obtain an accurate location to start with, while still retaining the benefit of geomagnetism-based navigation.

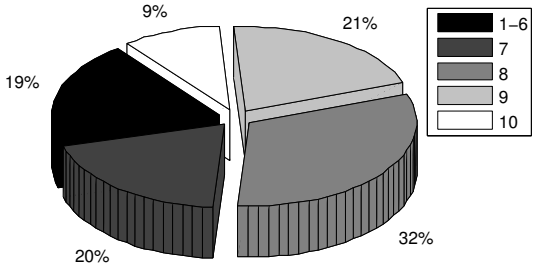


Fig. 20. Navigation satisfactory level from 1 to 10.

7 Related Work

To supplement the void of GPS's availability indoors, variety of indoor localization approaches have been proposed in the last decade. Due to the page limit, we have to omit discussions on peer-assisted and/or range-based approaches [9], [7], [21], [17], but rather focus on those related to our proposal include fingerprinting, crowdsensing, magnetism-based localization, and general indoor navigation.

7.1 Fingerprinting Approach

Traditional localization techniques measure signals from RF beacons to triangulate the mobile users' coordinates [5].

These techniques are crippled by their high cost of specialized mobile devices/infrastructures and the unstable localization accuracy due to construction interference. What replaces them is the fingerprinting approach, and the location discriminating fingerprints can be WiFi access points (AP) Received Signal Strength (RSS) [23], [24], [35], general RF signal [6], and light intensity [25]. These fingerprints can be captured by sensors embedded in smartphones, which saves the trouble of building and carrying expensive but cumbersome devices.

7.2 Crowdsensing Fingerprints Collection

All fingerprinting systems work in two phases: spot survey and localization. Spot survey collects fingerprints from known locations, then they are used to create a fingerprint database. Localization estimates user locations by comparing fingerprints sensed online to the database. However, the labor-intensive spot survey and the constant maintenance of the database have largely hampered a wide deployment of these systems.

To this end, crowdsensing (a.k.a. organic fingerprinting) is adopted by recent proposals. Redpin [6] takes a folksonomy-like approach that allows users to identify location themselves when they are wrongly located and then to correctly associate fingerprints to these locations. OIL [23] applies a similar approach to Redpin, but it further handles spatial uncertainty and labeling errors made by users. Zee [24] uses particle filter and dead reckoning to identify user's walking trace and enriches the fingerprint database with the WiFi data collected along the trace. ARIEL [15] differentiates rooms through clustering on WiFi fingerprints collected by randomly moving users to achieve a room level localization accuracy. Unloc [32] uses distinct patterns from accelerometer, WiFi RSSI, and magnetic fluctuations detected by smartphones as organic landmarks to help locating users. Loci [18] improves semantic location service through user feedback, in which user inputs are used to correct place detections by the service. Walk&Sketch [38] attempts to create floor maps using high resolution cameras mounted on users' backpacks.

7.3 Magnetism-based Indoor Localization

It is well known that geomagnetism can be "twisted" by building structures and can hence be used to depict indoor locations; a few proposals have exploited this property. Chung *et al*. [8] attach a compass to a rotating motor to develop an indoor location system based on geomagnetism. This approach demands a huge amount of time to fingerprint a single hallway, making its scalability questionable. Again based on geomagnetism, proposals in [13] and [29] apply particle and Kalman Filters, respectively, to robot navigation. All these proposals require dedicated devices.

7.4 Indoor Navigation Systems

Existing indoor navigation systems often assume the existence of certain localization support. Building upon a positioning middleware, Schougaard *et al*. [28] propose a hybrid navigation system that models indoor locations in both symbolic and geometric manners. An earlier work [20] focuses on multimedia user interface design that navigates people with cognitive impairments. As we discussed earlier, an integrated design involving both localization and navigation is necessary, exactly due to the need for real-time localization by an indoor navigation service.

8 Conclusions

Whereas a plethora of proposals on WiFi-based indoor localization systems have been proposed, we believe that an indoor navigation service may require features that are not provided by these existing localization systems. Motivated by the incompetent navigation service of Google Maps Indoor (GMI), we aim to eliminate the heavy reliance on a WiFi infrastructure and also on contributory floor maps can be beneficial to indoor navigation.

To this end, we proposed GROPING, an all-in-one system that includes map generation, localization, and navigation. GROPING relies on the geomagnetic field to characterize indoor locations. This allows GROPING to i) utilize crowdsensing for magnetic fingerprinting and for constructing a floor map from an arbitrary set of walking trajectories, and ii) to perform lightweight localization and hence navigation based on magnetic fingerprints and the constructed maps. Evaluations and user studies in a large shopping mall with 20 participants have demonstrated the high usability of GROPING's navigation service.

Whereas WiFi-based indoor localization systems show disadvantage in energy efficiency and fingerprint stability, the higher dimensionality of WiFi fingerprints, if properly used, may still offer better location discriminability than magnetic fingerprints. Therefore, we are considering the possibility of a hybrid system combining both technologies in our future work. Moreover, we also plan to make GROPING more autonomous by minimizing the required user interventions.

References

[1] Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome.

[2] Ekahau Positioning Engine. http://www.ekahau.com/.

[3] Google Maps Indoor. http://maps.google.com/help/maps/indoormaps/.

[4] K. Arning, M. Ziefle, M. Li, and L. Kobbelt. Insights Into User Experiences and Acceptance Of Mobile Indoor Navigation Devices. In *Proc. of ACM MUM*, pages 1–10, 2012.

[5] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In *Proc. of IEEE INFOCOM*, pages 775–784, 2000.

[6] P. Bolliger. Redpin - Adaptive, Zero-configuration Indoor Localization through User Collaboration. In *Proc. of ACM MELT*, pages 55–60, 2008.

[7] K. Chintalapudi, A. Iyer, and V. Padmanabhan. Indoor Localization Without the Pain. In *Proc. of ACM MobiCom*, pages 173–184, 2010.

[8] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman. Indoor Location Sensing using Geo-magnetism. In *Proc. of ACM MobiSys*, pages 141–154, 2011.

[9] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see Bob?: Human Localization Using Mobile Phones. In *Proc. of ACM MobiCom*, pages 149–160, 2010.

[10] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[11] B. J. Frey and D. Dueck. Clustering by Passing Messages between Data Points. *Science*, 315:972–976, 2007.

[12] R. Ganti, F. Ye, and H. Lei. Mobile Crowdsensing: Current State and Future Challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.

[13] J. Haverinen and A. Kemppainen. Global Indoor Self-Localization based on the Ambient Magnetic Field. *Robotics and Autonomous Systems*, 57:1028–1035, 2009.

[14] J. Hightower and G. Borriello. Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In *Proc. of ACM UbiComp*, pages 88–106, 2004.

[15] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. ARIEL: Automatic Wi-Fi based Room Fingerprinting for Indoor Localization. In *Proc. of ACM UbiComp*, pages 441–450, 2012.

[16] E. Jonsson. *Inner Navigation: Why We Get Lost and How We Find Our Way*. Scribner, 2002.

[17] X. L. K. Liu and X. Li. Guoguo: Enabling Fine-grained Indoor Localization via Smartphone. In *Proc. of ACM MobiSys*, pages 235–248, 2013.

[18] D. H. Kim, K. Han, and D. Estrin. Employing User Feedback for Semantic Location Services. In *Proc. of ACM UbiComp*, pages 217–226, 2011.

[19] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A Reliable and Aaccurate Indoor Localization Method using Phone Inertial Sensors. In *Proc. of ACM UbiComp*, pages 421–430, 2012.

[20] A. L. Liu, H. Hile, H. Kautz, G. Borriello, P. A. Brown, M. Harniss, and K. Johnson. Indoor Wayfinding: Developing a Functional Interface for Individuals with Cognitive Impairments. In *Proc. of ACM Assets*, pages 95–102, 2006.

[21] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the Limit of WiFi based Localization for Smartphones. In *Proc. of ACM MobiCom*, pages 305–316, 2012.

[22] M. Musthag, A. Raij, D. Ganesan, S. Kumar, and S. Shiffman. Exploring Micro-incentive Strategies for Participant Compensation in High-Burden Studies. In *Proc. of ACM UbiComp*, pages 435–444, 2011.

[23] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an Organic Indoor Location System. In *Proc. of ACM MobiSys*, pages 271–284, 2010.

[24] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort Crowdsourcing for Indoor Localization. In *Proc. of ACM MobiCom*, pages 293–304, 2012.

[25] N. Ravi and L. Iftode. FiatLux: Fingerprinting Rooms using Light Intensity. In *Proc. of Pervasive*, 2007.

[26] S. Reddy, D. Estrin, M. Hansen, and M. Srivastava. Examining Micro-Payments for Participatory Sensing Data Collections. In *Proc. of ACM UbiComp*, pages 33–36, 2010.

[27] H. Sakoe and S. Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *Acoustics, Speech and Signal Processing, IEEE Trans. on*, 26(1):43 – 49, 1978.

[28] K. R. Schougaard, K. Grnbk, and T. Scharling. Indoor Pedestrian Navigation Based on Hybrid Route Planning and Location Modeling. In *Proc. of Pervasive (LNCS 7319)*, pages 289–306. Springer, 2012.

[29] W. Storms, J. Shockley, and J. Raquet. Magnetic Field Navigation in an Indoor Environment. In *Ubi Pos. Indoor Nav. and Loc. Based Service (UPINLBS)*, pages 1–10, 2010.

[30] R. Tanawongsuwan and A. Bobick. Gait Recognition from Time-Normalized Joint-Angle Trajectories in the Walking Plane. In *Proc. of IEEE CVPR*, pages 726–731, 2001.

[31] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128:99–141, 2001.

[32] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No Need to War-Drive: Unsupervised Indoor Localization. In *Proc. of ACM MobiSys*, pages 197–210, 2012.

[33] O. Woodman and R. Harle. Pedestrian Localisation for Indoor Environments. In *Proc. of ACM UbiComp*, pages 114–123, 2008.

[34] J. Xiong and K. Jamieson. Towards Fine-Grained Radio-based Indoor Location. In *Proc. of ACM HotMobile*, pages 13:1–13:6, 2012.

[35] S. Yang, P. Dessai, M. Verma, and M. Gerla. FreeLoc: Calibration-Free Crowdsourced Indoor Localization. In *Proc. of IEEE InfoCom*, pages 2581–2589, 2013.

[36] Z. Yang, C. Wu, and Y. Liu. Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention. In *Proc. of ACM MobiCom*, pages 269–280, 2012.

[37] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In *Proc. of ACM MobiSys*, pages 205–218, 2005.

[38] Y. Zhang, C. Luo, and J. Liu. Walk&Sketch: Create Floor Plans with an RGB-D Camera. In *Proc. of ACM UbiComp*, pages 461–470, 2012.

Chi Zhang received his BS degree from Zhejiang University, China, in 2011. He is currently a PhD candidate at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests are indoor localization/navigation, social networks, and wireless sensor networks.

Kalyan P. Subbu is currently an assistant professor at Amrita university, Kerala, India. He was a postdoctoral researcher at Nanyang Technological University, Singapore from 2012-2013. He obtained a PhD in Computer Science from the University of North Texas, USA. His research areas are mobile and pervasive computing, applied machine learning.

Jun Luo received his BS and MS degrees in Electrical Engineering from Tsinghua University, China, and the PhD degree in Computer Science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland. From 2006 to 2008, he has worked as a post-doctoral research fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School of Computer Engineering, Nanyang Technological University in Singapore, where he is currently an assistant professor. His research interests include wireless networking, mobile and pervasive computing, applied operations research, as well as network security. More information can be found at http://www3.ntu.edu.sg/home/junluo.

Jianxin Wu received his BS and MS degrees in computer science from Nanjing University, and his PhD degree in computer science from the Georgia Institute of Technology. He is currently a professor in the Department of Computer Science and Technology at Nanjing University, China, and is associated with the National Key Laboratory for Novel Software Technology, China. He was an assistant professor in the Nanyang Technological University, Singapore. His research interests are computer vision and machine learning. He is a member of the IEEE.

APPENDIX A
MORE ON THE MAP GENERATION

Unlike prior proposals that often perform evaluations on simple building floors [36], [24], we choose venues with fairly complicated floor maps containing loops, as shown in Fig. 21 for the first floor of our large test site. While

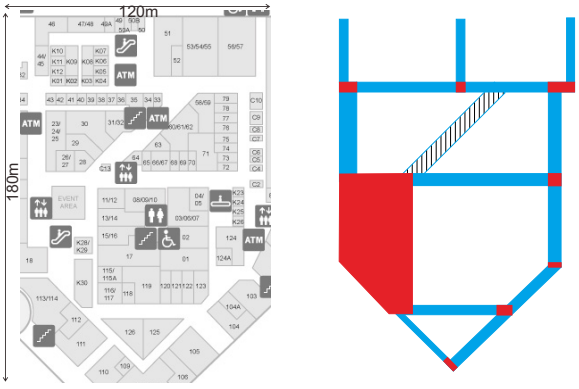


Fig. 21. The floor map of a shopping mall. The slight difference between this map and that shown in Fig. 17 is caused by a renovation in-between these two sets of experiments.

the left figure in Fig. 21 shows the ground truth floor map, the right figure is the skeleton (virtual) map we are aiming at generating by GROPING.[2] The floor has a polygonal shape with hallways forming many loops. It also contains conjunction points of either open area type (bottom-left corner) or turns with arbitrary degrees (bottom-right corner) instead of exact right angles in many existing tests. We first illustrate how GROPING's map generation module works on this floor map in Fig. 22, and then we summarize the performance of map generation in Fig. 23, along with discussions on its implications.

In Fig. 22(a) to (h), we use dashed lines to illustrate the actual walking trajectories of users and solid lines to represent the virtual trajectories "seen" by GROPING (through interpreting the sensor data). There are two major differences between these two set of trajectories. First, the virtual trajectories tend to be less straight, which is mainly due to the small angle estimation errors at the conjunction points. Second, the virtual trajectories are often shorter; the reason is that, as we only use the gyroscope readings collected from the conjunction points to estimate the angle, we assume the length of each conjunction to be negligible for now. The two errors are handled by $relaxLoop(M)$ at a later stage.

The stitching procedure is shown by Fig. 22(i) to (p). The procedure runs pretty smooth from (i) to (k), but certain distortion can be observed in (l) (fourth step). Now the whole outer loop has been explored, but it is yet to be determined whether the two end points actually coincide. When the sixth trajectory is introduced, the loop is closed but it is geometrically distorted (which results from the aforementioned two errors). Consequently, $relaxLoop(M)$ kicks in to make proper adjustments. The adjustment shown by Figure 22(n) is that i) each conjunction point is expanded based on the number of sample points involved, which results in the detection of the open space, and ii) each angle is computed as the average among all associated fingerprints. For brevity, the plots stop at Fig. 22(p) with some open loops, but they are closed with a couple of new trajectories in our experiments.

As a probabilistic algorithm, GROPING's map generation is prone to erroneous stitching. In Fig. 23, we use *precision* and *recall* as two metrics to evaluate the performance of the map generation. As discussed in Section 4.1.4, the performance of GROPING's map generation is controlled by defiThreshold, the definiteness threshold. Fig. 23 shows that, when increasing the threshold, we have a higher precision but lower recall. As a higher precision implies lower *false positive* rate and a higher recall suggests a lower *true negative* rate, we prefer to have a large value of defiThreshold, simply because true negative (i.e., overlapped segments are not detected) is almost harmless apart from wasting data. Therefore, we set defiThreshold $= 0.7$ to achieve a precision of 90%, which in turn wastes about 60% of trajectories.

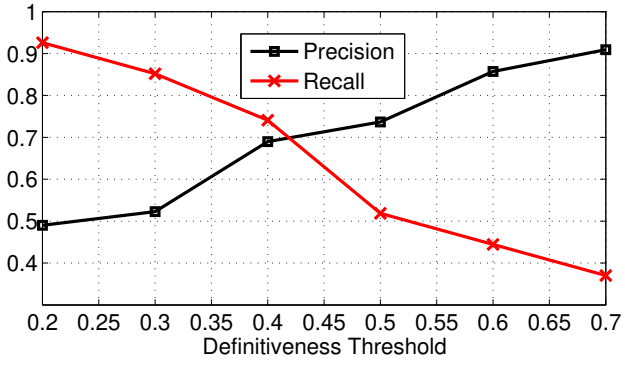
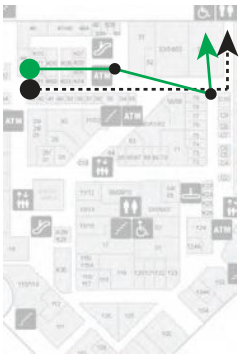


Fig. 23. Statistical evaluation of the map generation performance.

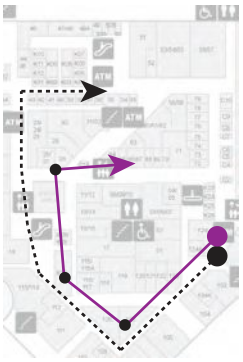
The remaining 10% of false positives may affect the virtual map of GROPING in two ways. First, it associates fingerprints that do not belong to a segment with that segment. Second, it creates hallways that do not exist on the real map. In reality, we may remove those non-existent hallways (hence the corresponding trajectories, along with the fingerprints) using crowdsensing (again). The basic idea is that non-existent hallways will never be passed by any future user, while existing hallways are bound to have some users walk along them. Therefore, GROPING keeps monitoring the user appearance on individual hallways, and it removes a hallway (and the corresponding trajectory) if no one appears on it for a long time. This also helps to detect a newly renovated hallway as illustrated by comparing Fig. 21 with Fig. 17.

2. One hallway (the hatched area in the right figure), though shown by the real map, is identified by our first team as blocked for renovation. Should such a "stale" map be used for indoor localization [33], [24], [36], it may incur large location errors. However, GROPING's crowdsensing map generation can handle such situations automatically: after the renovation was finished (see Fig. 17), the map were updated as some users are bound to pass through the new hallway.

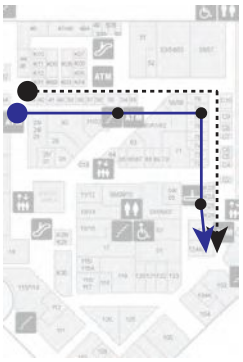
(a) First trajectory



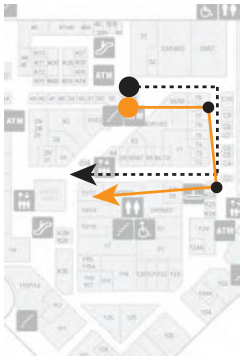
(b) Second trajectory



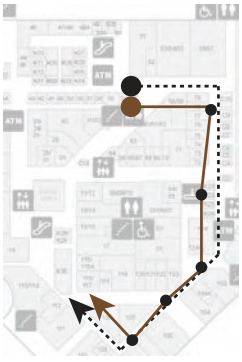
(c) Third trajectory



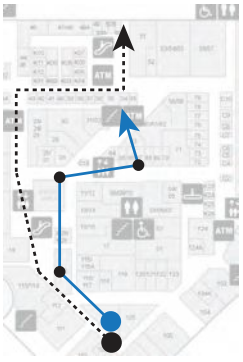
(d) Fourth trajectory



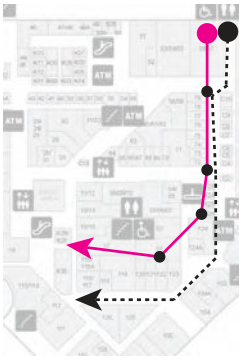
(e) Fifth trajectory



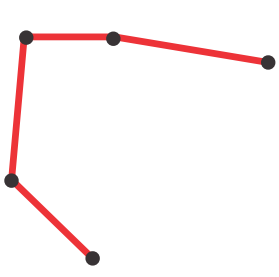
(f) Sixth trajectory



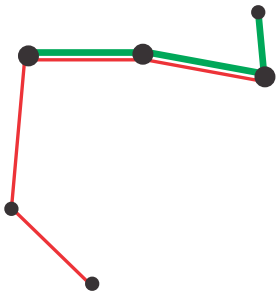
(g) Seventh trajectory



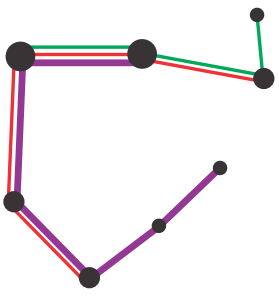
(h) Eighth trajectory



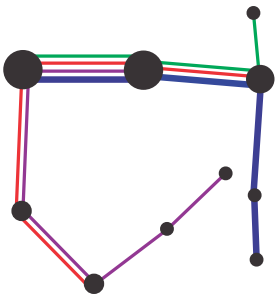
(i) First step



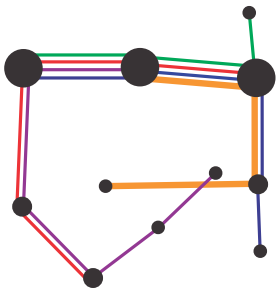
(j) Second step



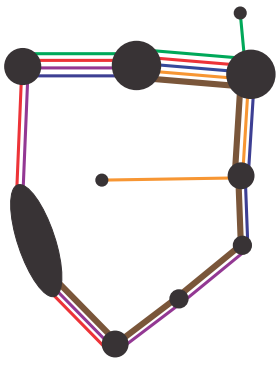
(k) Third step



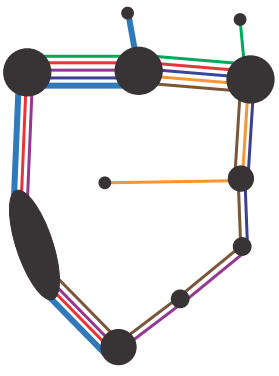
(l) Fourth step



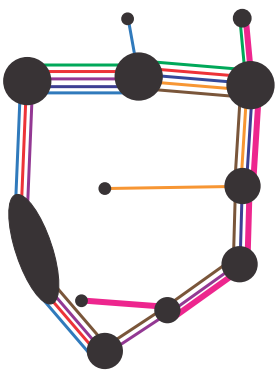
(m) Fifth step



(n) Sixth step

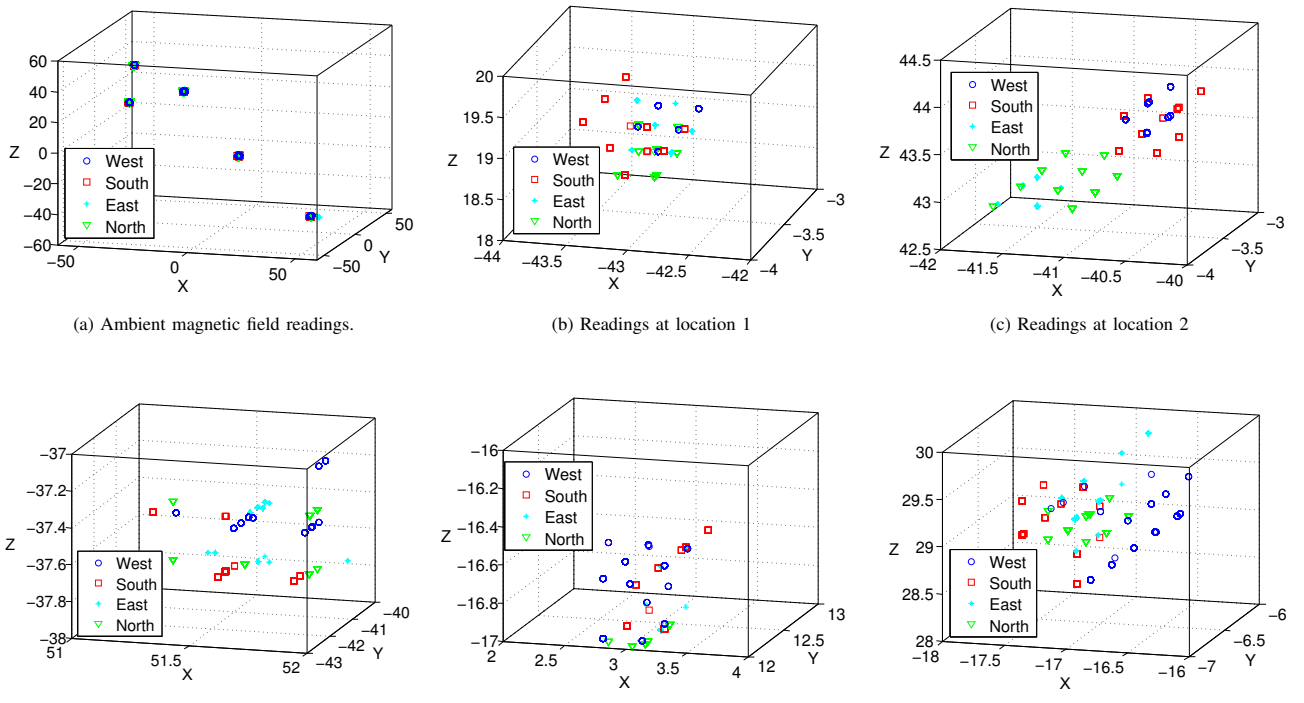


(o) Seventh step



(p) Eighth step

Fig. 22. Virtual map generation using eight trajectories and the associated magnetic fingerprints.



(a) Ambient magnetic field readings.

(b) Readings at location 1

(c) Readings at location 2

(d) Readings at location 3

(e) Readings at location 4

(f) Readings at location 5

Fig. 24. Ambient magnetic field reading, in μ-Tesla (or μT), taken by a smartphone at five different locations. For each location, we use West, South, East, and North to indicate the quadrants (centered at the smartphone) within which the "interfering" person is moving.

APPENDIX B
MORE ON THE AMBIENT MAGNETIC FIELD

In this section we present more results in understanding the stability of the ambient magnetic field. Since the application environments of GROPING may involve constant human movements around the smartphone client, we want to know how such movements affect the magnetometer readings taken by the smartphone. To this end, we measure the magnetic field at five different locations. At each location, the phone has a fixed orientation, but we let one person to move arbitrarily within each of the four quadrants centered at the phone location, namely West, South, East, and North. The distance between the person and the phone is limited within 2 meters. A reading is taken by the phone for each of such quadrant-limited movements.

We plot the 3D magnetic field vectors in Fig. 24. Fig. 24(a) shows the readings collected at these five locations altogether. Apparently, the differences in magnetometer readings caused by human movements within different quadrants are negligibly small so that they do not affect the location discriminating ability of the ambient magnetic field. To give a closer look at these differences, we plot the reading for individual locations in Fig. 24(b)-(f), respectively. Clearly, the variances of the ambient magnetic field may reach 100μT (see the scale of Fig. 24(a)), whereas those caused by human movements never go beyond 3μT (see the y-axis of Fig. 24(d) as an example). These experiments allow us to firmly conclude that human movements have very limited influence on the ambient magnetic field, so they do not affect the localization accuracy achievable by GROPING.