Peer-to-Peer Indoor Navigation Using Smartphones

Zuwei Yin, *Student Member, IEEE*, Chenshu Wu, *Member, IEEE*, Zheng Yang, *Member, IEEE*, and Yunhao Liu, *Fellow, IEEE*

***Abstract*— Most of existing indoor navigation systems work in a client/server manner, which needs to deploy comprehensive localization services together with precise indoor maps *a prior*. In this paper, we design and realize a peer-to-peer navigation system (*ppNav*), on smartphones, which enables the fast-to-deploy navigation services, avoiding the requirements of pre-deployed location services and detailed floorplans. *ppNav* navigates a user to the destination by tracking user mobility, promoting timely walking tips and alerting potential deviations, according to a previous traveller's trace experience. Specifically, we utilize the ubiquitous WiFi fingerprints in a novel diagrammed form and extract both radio and visual features of the diagram to track relative locations and exploit fingerprint similarity trend for deviation detection. We further devise techniques to lock on a user to the nearest reference path in case he/she arrives at an uncharted place. Consolidating these techniques, we implement *ppNav* on commercial mobile devices and validate its performance in real environments. Our results show that *ppNav* achieves delightful performance, with an average relative error of 0.9 m in trace tracking and a maximum delay of nine samples (about 4.5 s) in deviation detection.**

***Index Terms*— Peer-to-peer, indoor navigation, sequential fingerprints.**

I. Introduction

THE past decade has witnessed the conceptualization and development of smartphone-based indoor localization and navigation. Various approaches based on WiFi [1], [2], GSM [3], Sound [4], etc., have been proposed to enable localization indoors and further provide navigation services to end users, on the basis of path planing algorithms in addition to precise indoor maps.

While qualified of providing navigation service once appropriately deployed, such localization-enabled navigation systems are limited in two folds. First, they highly rely on an accurate and stable localization service pre-deployed by specific provider, which entails great challenges since current indoor localization systems are not yet ready for wide and easy deployment. Despite of numerous efforts on indoor localization, the applicability of previous attempts, either model-based or fingerprint-based, is significantly limited by labor-intensive deployment and insufficient accuracy. For example, WiFi fingerprinting-based localization requires remarkable initial efforts for bootstrapping yet yields pretty mediocre accuracy [1], [2], [5]. Second, a meticulous indoor map is always required to provide precise building structure and semantic place information, yet is hard and costly to acquire. Mainstream map providers such as Google Maps and Baidu Map currently supply indoor maps only in limited areas like large malls and airports. And these maps are generated mainly from manual construction by expert engineers. Despite of some approaches proposed to crowdsource indoor maps [6], [7], a sustainable incentive mechanism is yet required and the quality of the resulting data is not guaranteed. To conclude, previous navigation systems work in a classical client/server (C/S) thinking, which significantly depend on pre-deployed comprehensive location and map services and have yet to be achieved. A lightweight, efficient, and easy-to-deploy approach is strongly urged for practical uses.

In this paper, we specify an alternative *Peer-to-Peer (P2P) navigation* mode, which enables efficient navigation without resorting to pre-deployed location service or the availability of indoor maps. As shown in Fig. 2, the P2P navigation employs a previous traveller(Alice or Bob) to record his/her trace information (e.g., key turning points) along a path and share it with followers(Claire or David) for navigation. The idea is inspired by two fronts. First, people have widely resorted to non-technical solutions through nature navigation descriptions from others or landmarks in surroundings in daily life. It is demonstrated that people are able to navigate themselves to the destinations if provided timely hints, e.g., to take a turn or go upstairs. Second, P2P architecture has been extensively adopted in computer networks where participants (a.k.a peers) voluntarily offer to provide their own resources available to other peers without the need of a central server. Applying a similar thinking in navigation, P2P navigation enables self-motivated users who have travelled through a path to act as leaders, i.e., recording and sharing specific trace information to navigate potential followers. Several pioneer works have conditionally carried out the leader-follower mode for navigation [8], [9]. Such P2P mode advances especially in social and personal scenarios, where, for example, a group of users arrive successively for an appointment at a specific place or a vendor desires to direct customers to his own shop.

Translating the idea into practice, we design *ppNav*, a *P*eer-to-*P*eer indoor *Nav*igation system for smartphones. In *ppNav*, a leader records trace information along his travel along a path to a specific destination. Such information, including location specific features such as WiFi measurements and

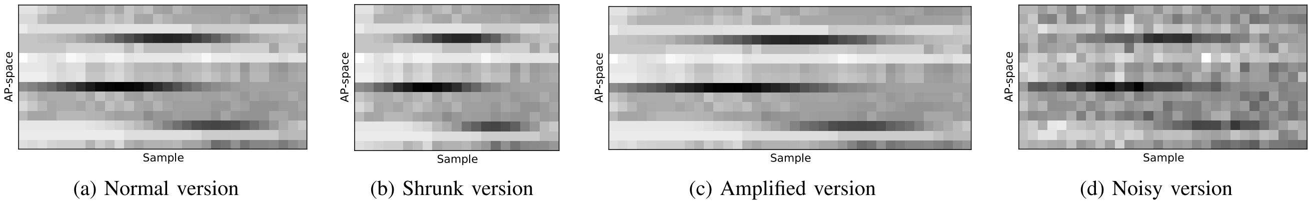
Manuscript received September 22, 2016; revised January 13, 2017; accepted January 26, 2017. Date of publication March 9, 2017; date of current version May 24, 2017. This work was supported in part by the NSFC under Grant 61522110, Grant 61361166009, Grant 61672319, Grant 61602381, Grant 61472057, and Grant 61632008, and in part by the National Key Research Plan under Grant 2016YFC0700100.

Z. Yin is with the School of Software and TNList, Tsinghua University, Beijing 100084, China, and also with the Zhengzhou Institute of Information Science and Technology, Zhengzhou 450001, China (e-mail: yinzw81@gmail.com).

C. Wu, Z. Yang, and Y. Liu are with the School of Software and TNList, Tsinghua University, Beijing 100084, China (e-mail: wucs32@gmail.com; hmilyyz@gmail.com; yunhaoliu@gmail.com).

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

Digital Object Identifier 10.1109/JSAC.2017.2680844



(a) Normal version (b) Shrunk version (c) Amplified version (d) Noisy version

Fig. 1. Fingerams of the same trace with (a) normal, (b) lower, (c) higher sampling frequency and (d) measurement noises.

user walking events of interests (e.g., headings, turning, going upstairs/downstairs), are fused to form a reference trace that will be later shared to other followers for guidance. When a follower arrives at the same building and demands an identical destination place, *ppNav* first directs him/her to lock on a certain point (typically yet not necessarily the same starting point as one leader's) that connects to the leader-provided reference trace. From then on, *ppNav* continuously measures the navigation trace information and synchronizes the walking progress with the reference trace. Accordingly, *ppNav* navigates the follower to the destination by promoting timely walking hints (such as directions and turns) at appropriate locations.

ppNav enables P2P navigation on smartphones without resorting to any pre-deployed services or precise maps, but its realization entails particular challenges. (1) We expect *ppNav* to alert users timely and properly by providing correct walking hints at the right time and locations. Without comprehensive localization services, however, we are unable to obtain precise location estimations. Thus given the environmental dynamics and the diversity of devices and walking speeds, how to accurately synchronize the follower's walking progress against the leader's reference trace turns into the first important problem to address. (2) As users may visit a specific destination from different starting positions, a friendly navigation system should intelligently lead users to splice the closest reference trace before the trace-driven navigation can come into operation. Since the leader's reference trace only covers the sampled path and we have neither the knowledge of other unknown areas nor the user's precise locations, how to guide a user to lock on a path connected to a reference trace arises as a key issue for practical uses of P2P navigation. (3) Incorrect and untimely direction hints both lead to wrong paths for a follower. As we only have the knowledge of the sampled path, *ppNav* should be able to responsively detect the deviation once a follower has walked off a reference path.

To overcome the challenges, we devise several novel techniques. First we propose a novel concept of fingerprint engram, termed as *fingeram,* which represents a diagrammed form of sequential WiFi fingerprint measurements along a path. We adopt WiFi signals as the basic vehicle due to its ubiquitous availability worldwide. Although WiFi fingerprints are known to be vulnerable to environment dynamics and only produce ordinary location accuracy for indoor localization, we observe that fingeram holds nice properties for precise trace synchronization. As shown in Fig. 1, being the diagrammed form of a sequence of fingerprints, certain features of fingeram remain stable in regards to contraction or amplification (caused by different WiFi sample frequency and walking speeds) and noises (induced by environment dynamics and device heterogeneity). We extract both radio and visual features for trace synchronization and deviation detection. Diving into the spatial propagation characteristics of WiFi signals, we further develop a rotation-based direction finding approach to guide a user to a closest starting location.

Unifying the above techniques together, we implement *ppNav* on Android platform. We conduct extensive experiments in multiple real-world scenarios. Experimental results demonstrate that *ppNav* yields a 90 percentile spatial error of 1.6m and an average error of 0.9m for follower tracking and detects deviations timely within 4.5s (9 samples at 2Hz).

In a nutshell, our core contributions are as follows.

- First we specify the Peer-to-Peer navigation model and propose a systematic design *ppNav* that achieves navigation on smartphones without pre-deployed localization services or even the knowledge of exhaustive indoor maps. In addition, we also envision such easy-to-deploy system as a complementation to outdoor navigation as well as an alternative to progressively crowdsoucing routing data for common localization services.
- Second, we propose fingeram, a novel diagrammed form of sequential WiFi fingerprints, and extract stable features that are resistant to RSS variations caused by diverse users, heterogeneous devices and environmental dynamics. We believe the fingeram can also benefit conventional fingerprint-based localization and other RSS-based applications.
- Finally, we implement *ppNav* on commodity smartphones and evaluate it in multiple modern buildings. The results demonstrate delightful performance of *ppNav* for everyday navigation.

The subsequent sections begin with the motivations and challenges, followed by a brief overview and the definition of fingeram. The detailed design, implementation and evaluation of *ppNav* is then presented. And finally we discuss some open issues, review related work and conclude this work.

II. Motivations and Challenges

A. Model Description

In everyday life, users urge for effective navigation in many scenarios. For example, a shop owner would like to deploy a self-owned navigation service to direct a customer to his shop. In contrast, a customer also demands for a navigation service to visit a specific shop in gigantic malls. A meeting coordinator would like to guide other participants smoothly all the way to

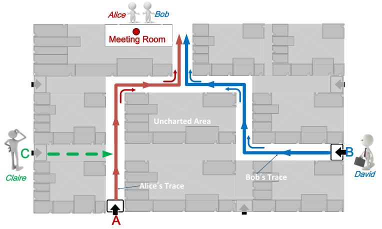


Fig. 2. A usage example of *ppNav*.

the gathering place, while one may need a navigation service to help reach a specific appointment location. Furthermore, when a driver parks his car in a large lot and go shopping, a friendly guidance to find his car when he finishes shopping would save much boresome time and thus largely improve the quality of shopping experience.

Due to the lack of pre-deployed comprehensive localization services and precise indoor maps, most of previous navigation based on accurate locations are infeasible for such scenarios. In this work, we attempt to develop an alternative model that does not rely on pre-installed location services and knowledge of indoor maps. Note that in above scenarios, multiple users will traverse the same path. Thus our key idea is nothing more than to leverage the experience of previous travellers, just as we naturally behave in routine life. Specifically, we collect specific trace data from previous travellers and extract appropriate walking hints to navigate others. Every user travelled a path could act as a leader for that path by recording trace information and sharing with others. And one could participate as either a leader or a follower, depending on the specific scenario. The leader and follower could even be the same person in case that one travels a path for multiple times. We term such operational model as *Peer-to-Peer navigation*, which does not necessarily rely on pre-deployed services and is very different from most conventional localization and navigation systems that work in a client/server mode [1], [10], [11].

B. Usage Examples

As shown in Fig. 2, suppose a meeting is arranged in a room marked by the red point. The meeting coordinator Alice arrives earlier at entrance A and walks from there to the meeting place as indicated by the red arrowed line. One participant Bob, who is familiar with the place, arrives at entrance B and travels to the meeting room along the blue arrowed path. To provide P2P navigation services for other participants, Alice and Bob could run the *ppNav* App during their own walk and generate a corresponding reference trace. Without such services, other participants who visited the meeting location for the first time could only get the directions by themselves or ask for directions step by step from others like property staffs in the building.

Specifically, during Alice and Bob's walks, the *ppNav* App running on their smartphones measures WiFi fingerprints and necessary sensor data (including gyroscope, magnetometer, barometer), which will be automatically processed and packed into a reference trace. Guiding information, e.g., turning, walking upstairs/downstairs, are also recognized and recorded in the reference trace for subsequent navigation. When other participant, e.g., David, arrives at the same entrance B with Bob, *ppNav* locks on to him at entrance B with Bob's reference trace and delivers the reference information provided by Bob to David. Then *ppNav* navigates David by estimating the walking progress, i.e., relative locations to Bob's reference trace. Based on the walking progress estimation, relevant walking hints (e.g., turning) extracted from the reference trace will be timely promoted on David's phone. In case a participant Claire arrives at a different entrance, e.g., C in Fig. 2, *ppNav* first directs her to the nearest reference trace, i.e., the one provided by Alice, and then perform regular navigation with the reference information from Alice. Similar data are measured using smartphone during a follower's walk.

During either leading walks or following walks, users need not to intentionally hold the smartphone in hand except that the follower may need to hold the phone in front of the chest when locking on the path of a reference trace since we make use of body blockage effects for this purpose. In general, they merely need to turn on the App and keep it running during a walk. In addition, the reference traces contributed by Alice and Bob to their friends for this meeting can be shared via cloud with the public when authorized. Furthermore, the following trace of Claire, if no deviation occurs during her walk, can also be leveraged as a reference trace for others.

Another attractive potential applications of *ppNav* is self-navigation for car finding in large indoor parking lots, a well-known annoying and knotty problem in modern society. When one user parks his/her car at a specific pool (denoted as P), he/she can use *ppNav* to record the trace from P to the parking lot exit/entrance (denoted as E). Then the recorded trace information can be used as the reference trace from E to car pool P (by appropriately reversing it). Later, when he/she comes back to E, *ppNav* could navigate him/her back to the specific pool with the reference trace collected by him/herself.

C. Design Challenges

ppNav only exploits the sensor data (including WiFi measurements and inertial sensory readings) along the pathways. Instead of accurately locating a user for navigation, we need to track the follower's relative locations with respect to the leader's reference trace. We adopt WiFi signals as fingerprints in *ppNav* thanks to the worldwide availability and easy accessibility on commodity smartphones. Existing fingerprint-based localization, even with a complete radio map of the whole building, yields considerable location errors [12], [13], e.g., >5m, which easily result in confusions among adjacent pathways in modern buildings. As we do not sample the whole building in *ppNav*, the accuracy would be even worse. Thus previous localization approaches can not be directly applied to obtain accurate relative locations in *ppNav*. Existing techniques also fail to detect deviation when a user veers off the correct path. Given that the fingerprints observed by different users differ in sampling frequency, total amount and absolute RSS values, a new model to utilize WiFi measurements for precise trace synchronization and deviation detection needs to be designed.

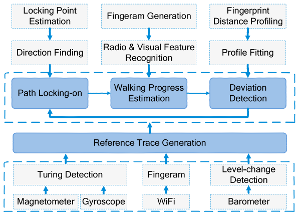


Fig. 3. System architecture.

Moreover, a friendly navigation system does not assume all users destined for a specific place start from the same location. When a follower appears at a position that has not been travelled by any leader, *ppNav* first needs to direct him to lock on a certain path. As we only sampled specific pathways and most areas are still unexplored, guiding a user from an unknown position to a reference path is like navigation in the dark without global knowledge or precise locations. Novel techniques are needed to shed light on the blinded period before the trace-driven navigation could take over.

III. Overview

As shown in Fig. 3, *ppNav* consists of two major parts, the trace generation part for the leader application and the navigation part for the followers.

A. Reference Trace Generation

We sample gyroscope, magnetometer and barometer to detect the guider's motion events during the trip, such as steps, turns, going upstairs/downstairs, walking/stop status, etc. Such motion events, together with a fingeram generated from sequential WiFi measurements, are shaped into a reference trace when the guider finishes the travel. The reference trace can be shared directly to a particular follower or via cloud with other users.

B. Path Locking-On

As a user may appear at a location that is different from the starting point of the leader, we need to direct a user to approach a nearest path that leads to the destination and has been travelled previously. This module designs a simple yet effective method for this purpose. Our method searches for the most likely directions to an estimated target point via easy cooperation of the follower himself.

C. Walking Progress Estimation

When a user is locked on a reference trace, *ppNav* navigates him to the destination by promoting timely motion hints, according to the synchronous events indicated by the reference trace. To achieve this, *ppNav* synchronizes the follower's instantaneous trace with the reference one using specific features extracted from the fingerams.

D. Deviation Detection

This module detects whether a user, after locked on a reference trace, is still on the correct path, in case that he does not follow well, e.g., taking a wrong turn. We propose a novel effective and efficient deviation detection algorithm based on fingerprint distance trend.

Note that the reference trace generation module works only on leaders' side while some common engines such as motion event detection and fingeram formation run on both the leader and follower applications.

IV. Fingeram Specification

A. Definition and Generation

WiFi fingerprint matching yields only limited location estimation accuracy, we observe that sequential fingerprints along a set of continuous locations (i.e., a path) behave much more stable. Fingerprint matching using sequential or just multiple fingerprints has been demonstrated to improve location accuracy [5], [12], [14]. Since our core task is to synchronize the reference trace and the following one, we naturally adopt sequential fingerprints observed along the path rather than discrete fingerprints of individual locations.

Trace synchronization based on sequential fingerprints is non-trivial. Walking along an identical path with the same starting and ending points, different users can observe different fingerprint sequences due to diverse behaviors and hardware capabilities. As a result, raw fingerprints of different traces could not match each other well. Inspired by advanced image processing techniques [15], [16], we propose to diagram the sequential WiFi fingerprints into a visual image and incorporate image-related features for matching. We term the diagrammed sequential fingerprints as *fingeram* thanks to its connotations of *fingerprint diagram* in addition to *fingerprint engram*. The former phrase depicts our core novelty in trace synchronization via fingerprint matching with both radio and visual features. The later embodies the key idea of *ppNav* to reuse the indications left behind by previous travellers as the term *engram* means a memory trace stored as biophysical or biochemical changes in the brain.

The fingerprint sequence of a trace can be represented as a matrix $F = [f^{(i,j)}]_{m \times n}$, where each column corresponds one fingerprint, $f^{(i,j)}$ indicates the RSS value of the ith AP in the jth fingerprint measurement, m is the total amount of observed APs and n is the number of samples during the trace. To transform a fingerprint sequence into a fingeram, we map the RSS matrix into an 8-bit grayscale image, which results in a visual form of the fingerprint sequence. Without loss of generality, we still use F to denote the diagrammed fingeram. Fig. 1a shows an illustrative example of fingeram, where one pixel corresponds an RSS value in the matrix of fingerprint sequence. Note that the AP order affects the image patterns of fingeram but does not pose an issue since the order, once determined, is unchanged throughout the navigation.

Compared to the naive fingerprint sequence [5], the unique advantages of fingeram lie on two fronts. First, although the absolute RSS fingerprints differ from leader's trace to the follower's, certain trends may keep stable for a sequence of fingerprints along a spatial path. For example, despite of absolute RSS variations, the RSS trend of each AP along a specific path keeps relatively consistent [7]. Second, the fingerprints,

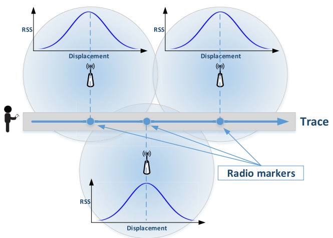


Fig. 4. Illustrative RSS peaks.

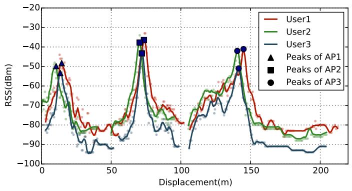


Fig. 5. RSS trends of three APs observed by different users.

when appropriately visualized, also enjoys graceful image attributes, e.g., different images with similar visual pattern but different sizes, scales, and resolutions could be accurately matched [15], [16]. Recall Fig. 1, although fingerprint sequences along the same path may vary due to various distortions such as lower/higher sampling frequency and measurement noises, which result in contracted, amplified and stained images respectively, the internal patterns of corresponding fingeram remain unchanged. These underpin the feasibility of trace synchronization via fingeram matching and encourage us to extract signal-based radio features and image-based visual features to serve trace-driven navigation.

B. Radio Features

As a visual version of sequential fingerprints, fingeram contains internal radio features that keeps stable over different traces along path. Specifically, the RSS changing trend is observed to be relatively stable along a specific path [7], [9]. Taking Fig. 4 as an example, suppose a user walks along the pathway. When he walks towards and then away from an AP, the corresponding RSS values may first increase and then decrease, resulting in the trend with a peak. Despite of diverse devices and walking speeds, different users would observe such a similar trend along the same path. Generally, the peak of an AP appears at the closest location on the path to that AP and thus turns out to be a spatially stable marker for a path.

As shown in Fig. 5, the location offsets of the RSS peaks observed by different users are limited within 3.7 meters, while the RSS differences can be as large as 10 dBm. In addition, one would encounter multiple RSS peaks from different APs within a path. Thus RSS peaks of all APs observed within a trace, which we term as *radio markers* hereafter, turn out to be effective alignment anchors for trace synchronization. As shown in Fig. 6, consistent radio markers can be detected from the fingerams of the reference trace, even though they are distorted in lengths and noises.

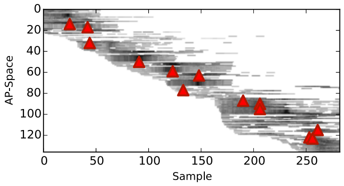


Fig. 6. Radio markers in fingeram.

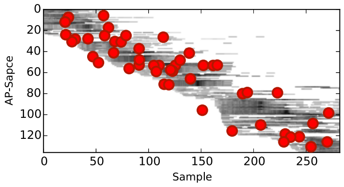


Fig. 7. Visual markers in fingeram.

C. Visual Features

In addition to the inherent radio features, fingeram also shares wonderful advantages of visual images. Various image matching techniques can be utilized for fingeram alignment [15], [16]. In practice, many factors impact the quality of generated fingerams. For example, different device gains lead to biased RSS observations, resulting in fingerams with diverse brightness. In addition, different sampling frequencies and different walking speeds both result in fingerams of different sizes. Yet we argue that certain features constrained by AP distribution and space geometry may keep unchanged under these distortions. Therefore, we can seek distinctive image features and efficient matching methods that are robust to changes in images scale, illumination and noises for precise fingeram alignment.

Specifically, we adopt SURF (Speeded Up Robust Features) [15], a widely used feature detector that can be used for object recognition, registration and classification in computer vision. SURF detects points of interests, usually on high-contrast regions on an image, as “feature descriptors” of the image that are detectable under distortions. Similar features will be identified and matched from other images as long as they preserve similar patterns or contain the same objects. In *ppNav*, we observe that for fingerams along the same path, the relative positions between a portion of such key points would persist and hence provide another dimension of anchors for fingeram alignment. Similar to radio markers, we term these image-based anchors as *visual markers* hereafter. As shown in Fig. 7, a considerable number of visual markers can be detected from fingerams, from which we can select reliable ones for walking progress estimation.

V. ppNav Design

A. Trace Generation

When a user takes a walk along a path destined for a place, *ppNav* can automatically construct a reference trace $J_R =< F, E >$ for that path. Specifically, F is the fingeram of measured WiFi fingerprints along the path, where each column corresponds an individual fingerprint. E marks a series of concomitant motion events such as turning and upstairs/downstairs extracted from inertial sensor data. Without location information, F and E are time-stamped as well as indexed with WiFi samples relative to the starting points. *ppNav* involves common sensors that are available on commodity smartphones for navigation. Specifically, we employ the following data as system inputs of *ppNav*. We measure WiFi observations as sequential fingerprints for constructing fingeram, which is used for estimating the follower's walking progress by fingeram synchronization. Gyroscope and magnetometer are used for detection of turning events while barometer is further leveraged for level-change detection.

As the fingerprints are continuously measured during walking, we employ a low-pass filter to sift out partial uncertain noises before converting them into a fingeram. Considering abundant APs crowded in modern buildings, a portion of low quality APs that appear few times or always hold extremely low RSS values are not necessary for fingeram and can be removed. Specifically, we define a continuous effective sampling rate r_s, defined as follows, to further sift out low-quality APs.

$$r_s = e_s/c_s,$$

where c_s denotes the max sample amount among all APs within a trace segment, and e_s denotes the amount of detected samples of current APs. We merely select the APs whose $r_s \geq 0.6$ to form fingerams in our scenarios, which would hold certain RSS trends with high confidence, although they might still experience missing RSS problem to some extent. We also remove APs with less than 10 samples within the whole trace, since they generally do not exhibit effective trends or peaks.

Considering that one or more APs, e.g., personal hotspots, might appear in the leader's reference trajectory yet do not present at all in the follower's trace (and vice versa), we only select the common APs existing in both trajectories when aligning the reference and following traces. There might also be several APs whose locations change between a pair of leading and following traces although they are observed in both traces. The APs may yield radio features with inconsistent relative positions in a pair of traces and thus influence the alignment results. Fortunately, such APs are commonly only a very small portion of detectable APs and we can apply robust techniques to mitigate their impacts (See Section V-B).

As for motion event detection, various advanced techniques have been developed and can be used [6], [17]–[19]. Hence in this section we mainly focus on the navigation design and leave a brief introduction on motion event detection in the next section.

B. Walking Progress Estimation

Given a reference trace J_R, we first assume a user arrives at the same starting position and discuss navigation in such case.

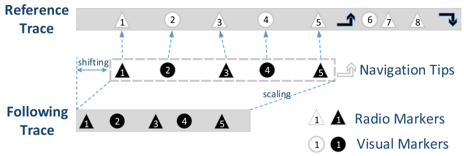


Fig. 8. Trace alignment by fingeram markers via scaling and shifting.

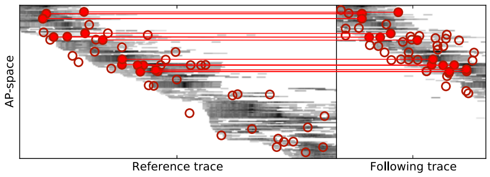


Fig. 9. Visual markers filtered by AP space (markers denoted by hollow circles are omitted.

As mentioned above, the key to provide timely walking tips is to accurately estimate the follower's walking progress relative to the reference trace. In *ppNav*, we synchronize the follower's trace with the reference trace by aligning their fingerams based on their radio and visual features as specified in Section IV. Suppose a user is locked on to a reference trace $J_R =< F_R, E_R >$ and denote his own navigation trace as $J_N =< F_N, E_N >$. The trace synchronization problem is to find an optimal alignment between the two fingerams $F_N = [f_N^{(i,j)}]_{m \times n_N}$ and $F_R = [f_R^{(i,j)}]_{m \times n_R}$. Formally, the alignment of the ith sample in F_N and the kth sample in F_R is represented as follows:

$$h : \mathbf{f}_N^{(i)} \mapsto \mathbf{f}_R^{(k)},$$

where $\mathbf{f}_N^{(i)}$ and $\mathbf{f}_R^{(k)}$ denote the ith (index of latest sample) and kth column in F_N and F_R, respectively. Since no rotation is involved in the mapping, there are basically scaling and translation operations during the alignment. Hence h can be derived in the form of $h(i) = \alpha i + \beta$, where α and β indicates the respective scale and translation factors to align F_N to F_R. Fig. 8 shows an illustration of alignment by fingeram markers via scaling and shifting. We extract two types of markers, i.e., radio markers and visual markers, for the fingeram alignment thanks to their location stability regarding various distortions. Radio markers of the same AP are observed at close physical locations, while visual markers corresponding to similar pattern generally appear at the same relative locations of a fingeram.

As shown in Fig. 9, each extracted visual marker $\mathbf{v} = (v^{(x)}, v^{(y)})$ corresponds to $v^{(y)}$th AP in the $v^{(x)}$th sample of the fingeram. Original SURF algorithm outputs a considerable number of such markers, which are not all necessary for alignment since in our case there is no image rotation issue. For all matched visual markers of two fingerams, we only need to consider those having the same y-axis coordinate $v^{(y)}$ (i.e., corresponding to the same AP), which efficiently reduces the amount of visual markers. Fig. 9 shows an illustrative example of the visual markers extraction matching results. For the fingeram of a sequence of 250 fingerprints, SURF detector can output as many as >200 visual markers, which can be pruned to around one-fifth. The remained matched visual markers are denoted as $V_N = \{\mathbf{v}_N^{(1)}, \mathbf{v}_N^{(2)}, \cdots, \mathbf{v}_N^{(n_v)}\}$ and $V_R = \{\mathbf{v}_R^{(1)}, \mathbf{v}_R^{(2)}, \cdots, \mathbf{v}_R^{(n_v)}\}$ for the follower's and the leader's trace, respectively.

In addition to visual markers, we also identify radio markers once they appear (note that they may not be observed within too few fingerprint samples). Radio markers are recognized by

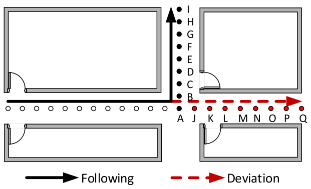


Fig. 10. Experimental scenario of fingerprint distance profile.

identifying potential RSS peaks of an RSS sequence as done in [7]. Fusing the radio markers contributed by different APs in the navigation fingeram, we obtain a set of radio markers $R_N = \{\mathbf{r}_N^{(1)}, \mathbf{r}_N^{(2)}, \cdots, \mathbf{r}_N^{(n_r)}\}$ where $\mathbf{r}^{(k)} = (r_N^{(k,x)}, r_N^{(k,y)})$ is a radio marker of the $r_N^{(k,y)}$th AP and is observed at the $r_N^{(k,x)}$th sample. Similarly, we can also extract a series of radio markers of corresponding APs in the reference trace, denoted as $R_R = \{\mathbf{r}_R^{(1)}, \mathbf{r}_R^{(2)}, \cdots, \mathbf{r}_R^{(n_r)}\}$. Only radio markers of APs that are observed in both traces will be considered for alignment.

Integrating all markers, we solve the alignment by minimizing the least square errors as follows via linear regression [20]:

$$arg \min_{\alpha,\beta} \sqrt{\sum_{k=1}^{n_v} \left(v_R^{(k,x)} - h(v_N^{(k,x)})\right)^2 + \sum_{k=1}^{n_r} \left(r_R^{(k,x)} - h(r_N^{(k,x)})\right)^2}. \tag{1}$$

To mitigate the influence of abnormal data, caused by unstable WiFi signals or moving personal hotspots, we employ a robust linear model (RLM) [20], which uses iteratively re-weighted least squares with a bisquare weighting function, instead of the ordinary linear model (OLM). Fig. 12 demonstrates that, with sufficient radio markers, OLM could evidently mitigate the impacts of abnormal data, which will be further validated in Section VI.

According to the derived α and β, we could scale and shift follower's trace to map to the reference trace by mapping function $h(i) = \alpha i + \beta$. By doing this, the follower's latest sampling index is mapped to a specific index in reference trace, which is the follower's walking progress relative to the reference trace. Then appropriate tips for navigation(e.g. turn or level change) could be promoted to the follower according to the motion events indicated in the reference trace.

C. Deviation Detection

In case the follower deviates from the trace, e.g., taking a wrong turn, we need to intelligently detect whether the user veers off the path or not and alert the user timely if yes. An intuitive way to achieve this is applying a threshold on the optimization errors in Eqn. 1. However, due to the variant trace lengths and uncertain amounts of markers, a fixed threshold value does not apply to various scenarios. In *ppNav*, we propose a novel algorithm based on fingerprint distance trend to effectively detect deviation.

Our key observation is that fingerprint similarity roughly decreases over increased distances. Specifically, if we match a fingerprint **f** against a series of nearby fingerprints observed on its opposite sites along the path, i.e., preceded and posterior fingerprints of **f**, we can obtain a "V"-zone pattern in decrease even to zero and then increases. As we can precisely estimate a user's walking progress if he is on the path, a "V"-like pattern will be still produced when we compare a fingerprint of the navigation trace to the sequence of nearby fingerprints of its aligned fingerprint in the reference trace. In contrast, fingerprint distances compared to whatever segment of fingerprints on the reference trace only yields random trends if the user deviates the path since the fingerprint is distant to any of those within the reference trace. Inspired by this observation, we devise novel algorithm to detection user deviation by identifying and comparing the fingerprint distance profiles.

Suppose the latest fingerprint observed by the follower is $\mathbf{f}^{(k)}$, which is mapped to a fingerprint $\mathbf{f}^{(k')}$ in the reference trace sampled at $k' = h(k)$. Considering respective d fingerprints preceded and posterior to $\mathbf{f}_R^{(k')}$ on the reference trace, we can derive a fingerprint distance profile for $\mathbf{f}_N^{(k)}$ using Euclidean distance as follows:

$$\phi\left(\mathbf{f}_N^{(k)}, \mathbf{f}_R^{(k'+i)}\right) = \parallel \mathbf{f}_N^{(k)} - \mathbf{f}_R^{(k'+i)} \parallel, i \in [-d, d], \tag{2}$$

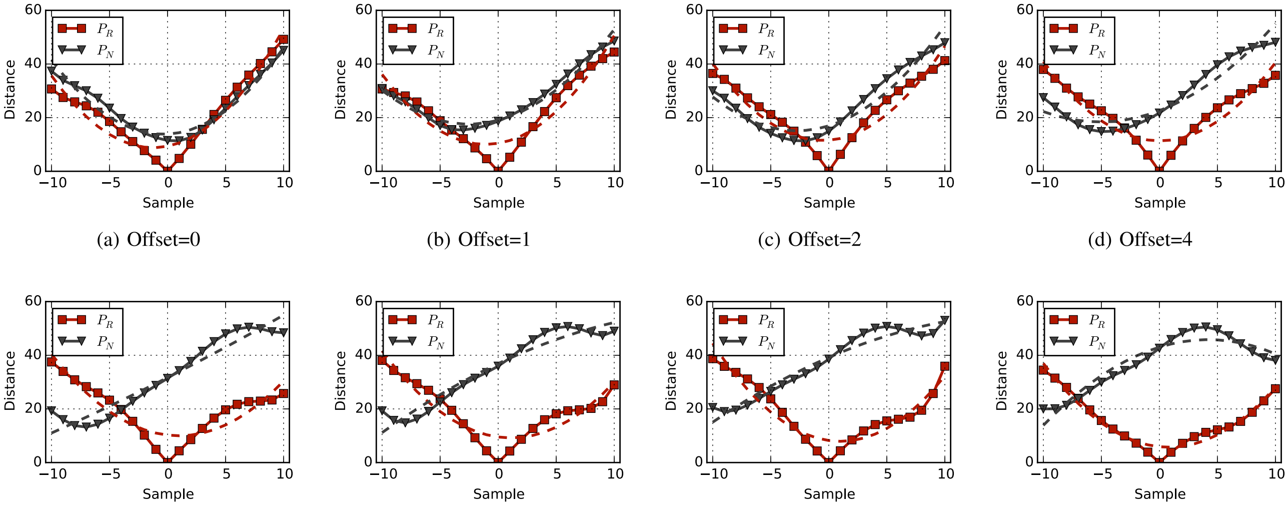
which results in a distance profile P_N of $2d + 1$ items. Note that we only take common APs that appear in both traces into account and typically $d < 10$ fingerprints are sufficient, as demonstrated by our experiments. Similarly, we can calculate a profile for $\mathbf{f}_R^{(k')}$, which is supposed to be the aligned fingerprint with $\mathbf{f}_N^{(k)}$, as $P_R = \{\phi(\mathbf{f}_R^{(k')}, \mathbf{f}_R^{(k'+i)}), i \in [-d, d]\}$.

We conduct preliminary measurements to validate our observations. As shown in Fig. 10, suppose the reference trace should be "A-B-C-D-E-F-G-H-I", but the follower turns mistakenly at point A and takes a trace as "A-J-K-L-M-N-O-P-Q". We calculate the distance profile P_N and P_R for each fingerprint sample during the trace from A to I. The results are depicted in order in Fig. 11. As seen, when the user does not deviate away, both P_N and P_R exhibit apparent V-pattern, although slight distortions are observed on P_N due to environmental dynamics and device diversity. While the graceful shape of P_R consistently holds (actually for any sample), P_N suffers significant deformation as the user veers off the path. Thus by examining the profile shapes of P_N to P_R, we can confidently identify whether the user deviates from the destined path.

Instead of directly evaluating the absolute P_N distribution, we attempt to compare P_N to P_R for deviation detection in purpose of achieving adaptivity to various scenarios. Specifically, we model the "V"-like distribution as a parabola and apply quadratic model $y = a(x - b)^2 + c$ to fit the profiles and examine the similarity between them. As shown in Fig. 11, we mainly concern coefficient a that embodies the curvature of the output parabola to compare the coupled distance profiles. Concretely, we devise a relative metric for this purpose:

$$\tilde{a} = \frac{a_N}{a_R}, \tag{3}$$

where a_N and a_R denote the estimation parameters of P_N and P_R, respectively. Larger $\tilde{a}$ indicates more similar profiles and vice versa. We then apply a threshold-based method to detect a deviation by examining the estimated



(a) Offset=0

(b) Offset=1

(c) Offset=2

(d) Offset=4

(e) Offset=5

(f) Offset=6

(g) Offset=7

(h) Offset=8

Fig. 11. Fingerprint distance profiles of different samples from deviation.

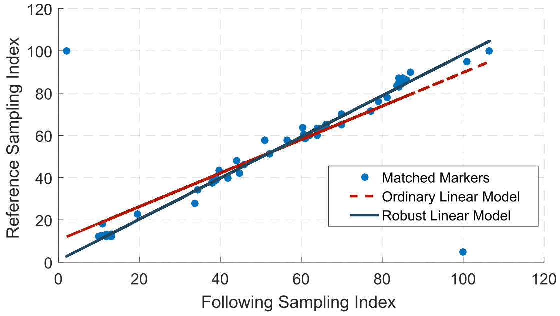


Fig. 12. Reduce the influence of abnormal data via robust linear regression.

$\tilde{a}$ of successive three samples. According to our experiments, we set the threshold of $\tilde{a}$ as 0.6 which balances the detection rate and delay.

D. Path Locking-On

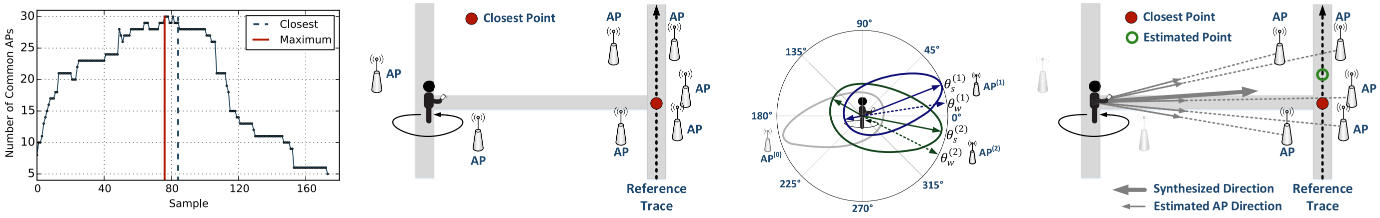
Previously we simply assume that a follower is locked on to the same entrance of a reference trace, which, however, is not guaranteed in practice. Recall Fig. 2, a user could arrive at different entrances. In such case, *ppNav* should direct him to the nearest path covered with at least one reference trace. Without pre-deployed location service and global map knowledge, the task is very difficult to navigate in the unexplored dark area. Previous works [8], [9] merely omits this problem and assumes followers will appear at the same entrance with the leader.

If a user is outside the building, he can navigate to a certain entrance using GPS service, which, however, will be no longer applicable when he has entered the building. We address this challenge in *ppNav* by exploring and exploiting the characteristics of AP space. The task is divided into two parts: First, we search for a locking-on point on the nearest reference trace. And second, find a direction towards the targeted point and direct the user there.

In *ppNav*, we seek the closest locking-on point as starting position to save locking-on costs. We observe that closer locations usually observe more common APs. Thus we induce an intuitive metric of common AP amounts to select the point that shares the most common APs with the user's current location as the targeted starting point. As shown in Fig. 13a, we compare the fingerprint measured at the user's current location to all fingerprints within a reference trace and depict the common AP amounts. As seen, the point that reports the highest common AP amount is pretty close to the spatially nearest point, with only an offset of 6 samples (about 3 meters). Although the targeted point selected in this way is not always the closest one, it is acceptable for a follower to take a slightly long walk for initial location.

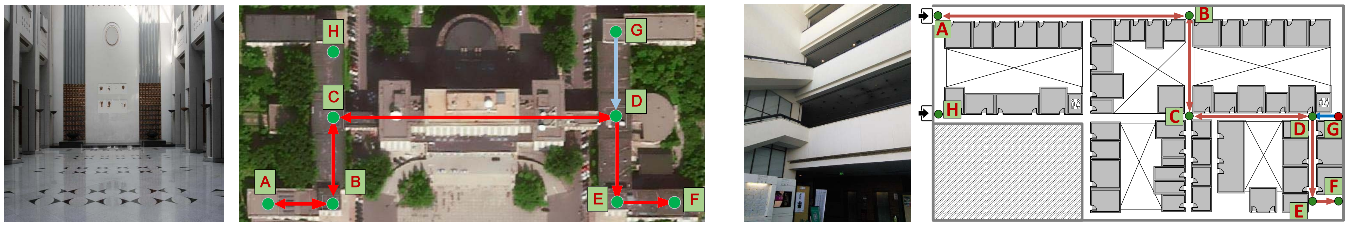
Once a targeted position is selected, we attempt to guide a user to there with the least efforts and without external location services or map information. The key insight to find the right direction towards targeted starting point is to leverage signal blockage of human body itself, as inspired by [21]. Specifically, when a user turns a circle with his phone, he may observe stronger RSS when facing a certain AP and lower RSS when he turns his back to it. Such phenomenon is leveraged in [21] to find the direction of an AP. In *ppNav*, we extend to multiple APs for direction finding between two distant locations.

Fig. 13 demonstrates an illustrative scenario where the user can easily navigate to the targeted starting point with not complex intervention. As shown in Fig. 13b, suppose the user is currently at L_C and the targeted point is L_T. We ask the user to turn a circle with a relatively slow speed, with the phone in front of the chest, and record a series of RSSs for each AP. To mitigate the RSS gains brought by device diversity, we only remain those APs that are also observable at L_T, yet with RSS values consistently lower than those measured at L_T. Then for each remained AP, we can estimate its direction θ_s, which is roughly the direction that observes the strongest RSS (when the user is facing the AP), and direction θ_w, which is the opposite to the direction that observes the weakest RSS (when the user is backing to the AP) within the circle (Fig. 13c). Note that we do not use the APs whose observed RSS values during



(a) Target point locking (b) The user rotates in place (c) RSS by different direction (d) Synthesized direction

Fig. 13. Lock on right direction for target point: (a) The point shares the largest number of common APs is physically close to the nearest location; (b) The user spins around slowly with a smartphone in front of his/her body; (c) Observed RSS values of several APs; (d) Direction synthesization based on estimated APs' directions.



(a) Academic building (b) Office building

Fig. 14. Experiment environments in (a) academic building and (b) office building.

the turning at L_C are consistently higher than those at L_T, because these APs may lead to wrong direction estimations if they are located between L_C and L_T. Suppose there are k APs involved and the ith AP is in direction $\theta^{(i)}$. The forward direction of the user is then derived by synthesizing all these directions as follows.

$$\bar{\theta} = \frac{1}{k}\sum_{i=1}^{k}\theta^{(i)}, \tag{4}$$

where $\theta^{(i)} = (\theta_s^{(i)} + \theta_w^{(i)})/2$ for robustness.

Given an estimated forwarding direction, the user may need to judge the path by himself in the uncharted area before he is locked on. We request the user to iteratively turn a circle to refine the direction, especially when he encounters a crossing or a wall. To determine whether the user is on the path or not, we successively calculate the fingerprint similarity, which is supposed to increase when the user get closer to the path and decrease in contrast, as we have done in the deviation detection module. Hence the point from which the fingerprint similarity starts decreasing from an increasing trend is probably the targeted location where the user gets on the path.

VI. Implementation and Experiments

A. Experimental Setup

To evaluate *ppNav*, we build a prototype on commodity smartphones (Google Nexus 5 and Nexus 7) running Android platform. We conduct experiments in both (I) an academic building with a testing area of $68000m^2$ and (II) an office building with an area of $11000m^2$, as shown in Fig. 14.

We recruit four volunteers to participate in our experiments. As shown in Fig. 14, we let the leader travel along the path "A→F" and "G→A" as the reference trace in both area. Then we employ three followers to walk long the trace with timely tips by *ppNav*. Specifically, the followers are not informed of the destination and the exact start-point. We let the follower start from "H" if the leader start from "A", and let the follower start from "E" if the leader start from "G" accordingly. The users naturally hold their smartphones in hand during walking.

B. Performance of Motion Event Detection

1) Turning Detection: To detect turning events and capture the rotating angle, we employ both magnetometer and gyroscope, which reports the absolute angles and the angular velocity respectively [22], [23]. We acquire the ground truth turning angles from the floor plans. To extensively understand the turning angle estimation error and turning detection delay, we let users manually tag checkpoints at each turning's starting and ending moments. With such results, the turning hints for a follower should be notified several seconds before the synchronized position of the turning starting events. As shown in Fig. 16, *ppNav* achieves great performance in turning angle estimation with a 90 percentile error of 18°, which is sufficient enough for turning detection since a turning angle in real buildings is in general greater than 45°. According to our experiments, turning events typically last for a duration of 2.75s. Fig. 17 illustrates the performance of detecting turning position (i.e., turning start points). As seen, *ppNav* identifies 90% of turning events in 0.5s after turning starts, which means that the average spatial shift is 0.5m, assuming a natural walking speed of 1m/s.

2) Level-Change Detection: To detect level changes, i.e., going upstairs/downstairs events, we employ the barometer sensor to read the atmospheric pressure data. As shown in Fig. 15, the pressure values within the same level are relatively stable (with a maximum variation of 10Pa as measured in

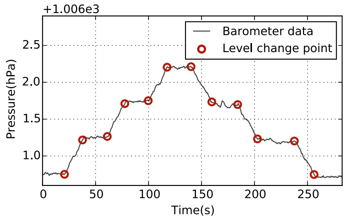


Fig. 15. Level-change detection.

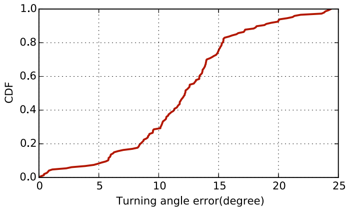


Fig. 16. Turning angle accuracy.

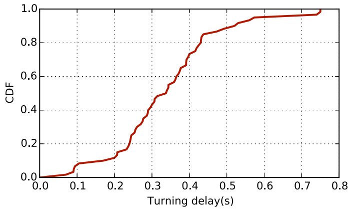


Fig. 17. Turning detection delay.

our experiments), yet vary significantly over different floors (with a minimum different between adjacent floors of 44Pa). Thus it is feasible to perform a threshold-based method to detect the level changes [24]. As a level-change event would typically last for a short period, we claim an event (i.e., the starting of a level change) when the pressure changes for over 12Pa and confirm the event (i.e., the starting of a level change) when the pressure varies over 40Pa. We collect 5 traces of going upstairs and downstairs in both experimental buildings, which contain 60 level-change events in total. In our experiments, we successfully detect all upstairs/downstairs events with a sliding window of around 8 seconds. The sliding window may cause certain delays, but does not affect the navigation performance since the level-change detection module only runs in reference trace generation part.

C. Performance of Trace Synchronization

As ordinary linear model (OLM) is sensitive to outliers caused by interferences such as fragile WiFi signals, we leverage a robust linear model(RLM) to mitigate the effects of noises, which uses iteratively reweighted least squares method [20] to find the maximum likelihood estimates.

To evaluate the performance of trace synchronization in term of physical space errors, we let leaders and followers manually mark a set of checkpoints along the paths as ground truth. Specifically, we set in total 41 and 20 checkpoints on each pathway in area I and area II, respectively. As shown in Fig. 18, RLM remarkably outperforms OLM, yielding an average alignment error of 0.9m and a 90%ile error of 1.6m. For comparison, the average and 90 percentile errors of OLM are 1.7m and 4.3m, respectively. Although RLM produces relatively higher computation costs, we need to perform regression only once and can reserve the regression coefficients for subsequent alignments during the same trace, since the speed differences and starting point offsets between the leader and follower are relatively stable for a specific pair of navigation trace and reference trace.

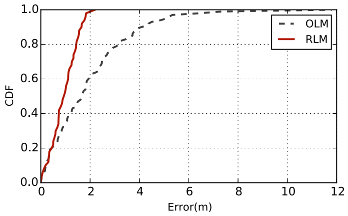


Fig. 18. Trace alignment errors.

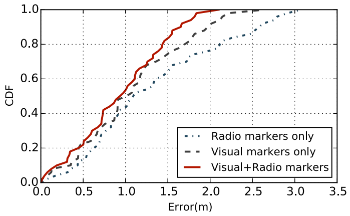


Fig. 19. Impacts of different markers.

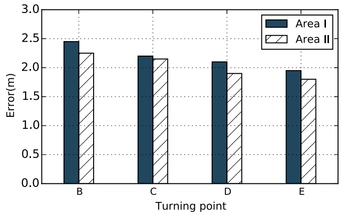


Fig. 20. Alignment error of turning points.

In addition, we particularly test errors of fingeram alignment for the turning points B,C,D,E in both area I and II, which are critical to navigation quality. As shown in Fig. 20, the errors are all under 2.5m and decrease gradually with respect to increasing walking distances. This is because that longer trace yields more fingeram markers, which may lead to more robust trace alignment. The reason that the alignment errors of the turning points are larger than the average ones over the whole trace is that turning detection errors further cause location offsets, which are included in the alignment errors.

We further evaluate the effects of individual type of markers for trace alignment using the same trace data. Fig. 19 shows the performance of fingeram alignment under three settings: radio markers only, visual markers only, and radio markers and visual markers. As seen, while either type of markers results in considerable accuracy, accounting for both markers yield the best performance. Specifically, the maximum error is limited by 2.1m, while the average error is about 0.9m.

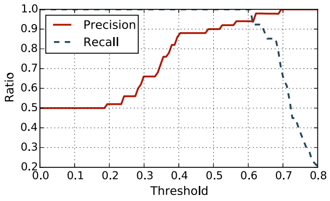


Fig. 21. Deviation detection rate.

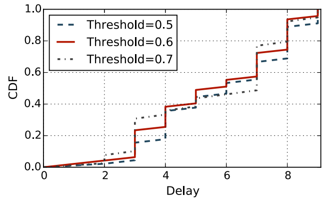


Fig. 22. Deviation detection delay.

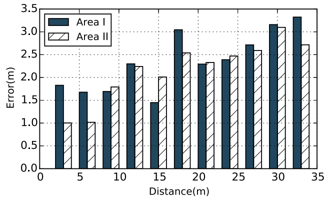


Fig. 23. Target point errors.

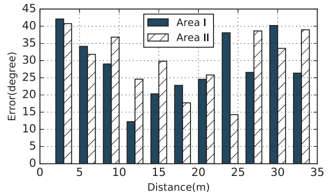


Fig. 24. Target direction error.

D. Performance of Deviation Detection

Now we evaluate the delay of deviation detection. As deviations usually appear in intersections, we set purposed deviation point at a special intersection of each pathway (point D in both experimental area I and II) and let users mark on smartphone actively when passing the points for ground truth. Fig. 21 depicts the precision and recall of deviation detection on 50 pairs of normal following traces and 50 pairs of deviated traces with different threshold values. The results show that any threshold within the range of [0.50, 0.64] yields considerable precision and recall on detection. Fig. 22 plots the delay distributions of deviation detection using these threshold values. As seen, the delays in all cases within different thresholds are all less than 10 samples. Larger thresholds result in smaller delay in detection, yet yields more false alarms. In contrast, smaller threshold values achieves more robust deviation alert, yet lead to larger delay. In our experiments, we adopt an intermediate value of 0.6, which produces balance performance in detection rate (precision of 93% and recall of 92%) and detection delay (with an average delay of 5.6 samples). Considering a typical WiFi sampling frequency of 2Hz and an average walking speed of 1.0m/s, the results equivalently indicate that we could detect deviations within a maximum delay of 4.5s in time and 4.5m in physical distance.

E. Performance of Lock On Direction

To evaluate accuracy of path locking-on, we let followers rotate and move for locking on estimated direction with an interval of 3m. The users spin around slowly (18°/s) at each stop with smartphone in front of body. We first evaluate the accuracy of finding a locking-on points at different distances by calculating the location errors between the physical closest point on the reference path and the target points estimated at different distances, from 33m to 3m. As shown in Fig. 23, the estimation errors decreased when the user gets closer to the path. Note that results in area II are in average better than those in area I, which is because narrower corridors in area II result in severer changes in amounts of common APs. Fig. 24 shows the direction errors for the targeted points determined at a distance of 33m. The average accuracy over all distances is about 29.5°, which is sufficient for direction guiding in practice since two adjacent pathways are usually 90° connected. In addition, as can be seen, the direction errors first decrease and then increase again when the user iteratively spins and gets closer to the targeted points. The reason is that few common APs are available when the user is too far and no notable RSS difference would be observed if he is too close.

VII. Discussion

A. Initial Locking-On

If the user is too far away from any reference traces, *ppNav* fails to guide him to a close path since perhaps no common AP would be observed at the user's location and any point on the reference trace. Nevertheless, if we gather sufficient leaders' traces that cover a certain part of the building pathways, such case will be efficiently avoided.

B. Quality of Traces

Although *ppNav* does not exert any constraints to participants during walking, arbitrary user behaviors, e.g., stop-and-go and scurrying from side to side, will impair the quality of the gathered trace information. Thus to record only the trace information along a path when the user is walking, we could employ inertial sensors (e.g., accelerometer) to detect the walking status of the user. In practice, when a user offers to provide hints for his friend, he will general try the best to provide good quality information. In case of sharing reference trace among strangers, quality control and privacy protection mechanisms need to be developed in the future.

C. Energy Consumption

ppNav calls WiFi interface that generally consumes more energy than the low-power inertial sensors. Yet we argue the power consumption is affordable because the *ppNav* application only runs during navigation, which usually will not be too long. In addition, compared to computation-hungry vision-based approach [9], our design is actually significantly more energy efficient.

D. Evolving to Generic Navigation Services

ppNav can be extended to progressively construct a generic navigation services by gathering sufficient reference traces and devising effective techniques to splice them to form an accessible roadmap. To achieve this, we need to design path planning algorithms to find the shortest path for friendly navigation [9].

VIII. Related Works

Indoor localization and navigation have been extensively studied in recent years. Many systems achieve precise location estimation using dedicated hardware such as infrared [25], ultrasound sensors [26], RFID [11], etc. Various approaches have been proposed to leverage the pervasive WiFi infrastructure for localization based on either ranging [27], [28] or fingerprinting [1], [12], [14], [29]–[32]. Other ubiquitously available radio signals such as FM [33], GSM [3] are also exploited as fingerprints. Thanks to the prosperity of mobile sensing, location fingerprints are recently further extended to magnetism [34], sound [4], [35], and multi-modal ambient features [14]. These systems either rely on specialized hardware or require labor-intensive site survey for deployment.

Traditional WiFi fingerprint-based localization for smartphones, even with a complete radio map of the whole building, usually yields considerable location errors [13], e.g., >5m, which is not sufficient for navigation. Recent advances improve the accuracy significantly by leveraging acoustic ranging [12] or image matching [14]. Recently decimeter-level localization methods have been proposed by leveraging the physical layer Channel State Information (CSI) [36] or specially modulated signals, such as Wideo [37], Widar [38], SpotFi [39], Chronos [40] etc. These works, however, rely on CSI that is only available on specific networks interface cards (e.g., Intel 5300) with firmware modifications and is not accessible on commercial smartphones.

Many crowdsourcing-based schemes have been proposed to reduce deployment costs of indoor localization systems by distributing the fingerprint calibration task to a large number of participatory users [5], [10], [19], [41]. While facilitating the deployment stage, these works usually assume the availability of precise indoor maps, which, however, are difficult to obtain in practice. Some researchers explore to incorporate inertial sensing and crowdsourcing to construct indoor floorplans for indoor localization and navigation [6], [7], [19], [42]. Appropriate mechanisms need to be designed to guarantee the precision of resulted maps as well as encourage user participation [43], [44]. Different from traditional navigation services, *ppNav* is an easy-to-deploy system that do not depend on pre-deployed comprehensive location services and precise digital maps.

The leader-follower mode has been referenced in several recent systems [8], [9], [34]. Reference [34] navigates blind users using customized device for magnetic sensing. Two recent systems, Travi-Navi [9] and FOLLOWME [8], both employ trace-driven navigation on smartphones. Travi-Navi synthesizes vision, WiFi and other inertial information to enable a user to easily bootstrap navigation services without infrastructure support. FOLLOWME exploits magnetic sensing and dead-reckoning to achieve last-mile navigation for smartphone users. The design of *ppNav* is inspired by these systems, but is very different from several aspects. First, both the two systems assume that followers will arrive at the same entrance with previous leaders, which is not realistic in practice. In contrast, *ppNav* can direct a user to lock on a closest path from a random start point. Second, while vision-based Travi-Navi provide intuitive tips for followers, the image quality is easily impaired by human motions and lighting conditions. To overcome this, users may need to hold smartphone vertically and steadily during walking. *ppNav* minimizes the cooperation efforts for path locking-on and does not exert any constraints during navigation. Finally, although both Travi-Navi and *ppNav* exploit WiFi measurements,we exploit sequential fingerprints in a novel diagrammed form, which is demonstrated to be more efficient and could further benefit various other WiFi-based applications, such as indoor localization, frequent trajectory mining, location-based AP access control, etc.

IX. Conclusion

In this paper, we present *ppNav*, a Peer-to-Peer navigation system for smartphones, which enables efficient navigation without resorting to pre-deployed location service or the availability of indoor maps. *ppNav* employs a previous traveller to record the trace information along a path and share them with later users for navigation. We implement *ppNav* on commercial phones and validate its performance via real experiments. In addition to a fast-to-deploy navigation service, we envision and intend to extend *ppNav* as an alternative way to progressively crowdsource data for generic localization systems.

References

[1] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 775–784.

[2] M. Youssef and A. Agrawala, "Handling samples correlation in the Horus system," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1023–1031.

[3] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-ID sequence matching," in *Proc. ACM MobiSys*, 2011, pp. 293–306.

[4] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proc. ACM MobiSys*, 2011, pp. 155–168.

[5] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 444–457, Feb. 2015.

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

[7] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: Indoor pathway mapping made easy," in *Proc. USENIX NSDI*, 2013, pp. 85–98.

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

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

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

[11] J. Wang and D. Katabi, "Dude, where's my card? RFID positioning that works with multipath and non-line of sight," in *Proc. ACM SIGCOMM*, 2013, pp. 51–62.

[12] H. Liu *et al.*, "Push the limit of WiFi based localization for smartphones," in *Proc. ACM MobiCom*, 2012, pp. 305–316.

[13] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Proc. ACM IPSN*, 2015, pp. 178–189.

[14] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu, "Enhancing WiFi-based localization with visual clues," in *Proc. ACM UbiComp*, 2015, pp. 963–974.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[17] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. ACM UbiComp*, 2013, pp. 225–234.

[18] 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*, 2012, pp. 421–430.

[19] 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*, 2012, pp. 197–210.

[20] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Commun. Statist.-Theory Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[21] Z. Zhang *et al.*, "I am the antenna: Accurate outdoor ap location using smartphones," in *Proc. ACM MobiCom*, 2011, pp. 109–120.

[22] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, "Mobility increases localizability: A survey on wireless indoor localization using inertial sensors," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 54:1–54:34, Apr. 2015.

[23] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proc. ACM MobiCom*, 2014, pp. 605–616.

[24] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, "Barometric phone sensors: More hype than hope!" in *Proc. ACM HotMobile*, 2014, p. 12.

[25] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91–102, 1992.

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

[27] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. ACM MobiCom*, 2010, pp. 173–184.

[28] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proc. ACM MobiSys*, 2013, pp. 249–262.

[29] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning," in *Proc. ACM UbiComp*, 2013, pp. 147–158.

[30] J. Machaj, P. Brida, and R. Piché, "Rank based fingerprinting algorithm for indoor positioning," in *Proc. IEEE IPIN*, 2011, pp. 1–6.

[31] X. Tian, R. Shen, D. Liu, Y. Wen, and X. Wang, "Performance analysis of RSS fingerprinting based indoor localization," *IEEE Trans. Mobile Comput.*, to be published.

[32] X. Tian, Z. Song, B. Jiang, Y. Zhang, T. Yu, and X. Wang, "HiQuadLoc: A RSS fingerprinting based indoor localization system for quadrotors," *IEEE Trans. Mobile Comput.*, to be published.

[33] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. ACM MobiSys*, 2012, pp. 169–182.

[34] T. H. Riehle *et al.*, "Indoor magnetic navigation for the blind," in *Proc. IEEE EMBC*, Aug./Sep. 2012, pp. 1972–1975.

[35] Y.-C. Tung and K. G. Shin, "EchoTag: Accurate infrastructure-free indoor location tagging with smartphones," in *Proc. ACM MobiCom*, 2015, pp. 525–536.

[36] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, p. 25, 2013.

[37] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "WiDeo: Fine-grained device-free motion tracing using RF backscatter," in *Proc. USENIX NSDI*, 2015, pp. 189–204.

[38] K. Qian, C. Wu, Z. Yang, C. Yang, and Y. Liu, "Decimeter level passive tracking with WiFi," in *Proc. ACM HotWireless*, 2016, pp. 44–48.

[39] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. ACM SIGCOMM*, 2015, pp. 269–282.

[40] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. USENIX NSDI*, 2016, pp. 165–178.

[41] C. Wu, Z. Yang, C. Xiao, C. Yang, Y. Liu, and M. Liu, "Static power of mobile devices: Self-updating radio maps for wireless indoor localization," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 2497–2505.

[42] S. Chen, M. Li, K. Ren, X. Fu, and C. Qiao, "Rise of the indoor crowd: Reconstruction of building interior view via mobile crowdsourcing," in *Proc. ACM SenSys*, 2015, pp. 59–71.

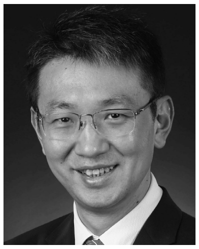
[43] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. ACM MobiCom*, 2012, pp. 173–184.

[44] X. Zhang *et al.*, "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, 1st Quart., 2016.

Zuwei Yin (S'16) received the B.E. and M.S. degrees from the Zhengzhou Institute of Information Science and Technology, Henan, China, in 2003 and 2007, respectively. He is currently pursuing the Ph.D. degree with the School of Software, Tsinghua University. He is currently an Assistant Professor with the Zhengzhou Institute of Information Science and Technology. His research interests include wireless networks and pervasive computing.

Chenshu Wu (M'15) received the B.E. degree from the School of Software, Tsinghua University, Beijing, China, in 2010, and the Ph.D. degree from the Department of Computer Science, Tsinghua University, Beijing, China, in 2015. He is currently a Post-Doctoral Researcher with Tsinghua University. He is a member of the Tsinghua National Laboratory for Information Science and Technology. His research interests include wireless networks and pervasive computing.

Zheng Yang (M'11) received the B.E. degree in computer science from Tsinghua University, in 2006, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, in 2010. He is currently an Associate Professor with Tsinghua University. His main research interests include wireless ad-hoc/sensor networks and mobile computing. He is a member of the ACM.



Yunhao Liu (F'16) received the B.S. degree in automation from Tsinghua University, China, in 1995, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, USA, in 2003 and 2004, respectively. He is currently a Cheung Kong Professor and the Dean of the School of Software, Tsinghua University, China. His research interests include RFID and sensor network, the Internet of Things, mobile computing, and cloud computing. He is a Fellow the ACM.