

ppNav: Peer-to-Peer Indoor Navigation for Smartphones

Zuwei Yin^{*†}, Chenshu Wu^{*}, Zheng Yang^{*}, Nicholas Lane[‡], Yunhao Liu^{*}

^{*} School of Software and TNList, Tsinghua University

[†] Zhengzhou Institute of Information Science and Technology

[‡] University College London and Bell Labs

{yinzw81, wucs32, hmilyyz}@gmail.com, niclane@acm.org, yunhaoliu@gmail.com

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 priori. In this paper, we design and realize a Peer-to-Peer navigation system, named *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. 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 0.9m in trace tracking and a maximum delay of 9 samples (about 4.5s) in deviation detection.

I. INTRODUCTION

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

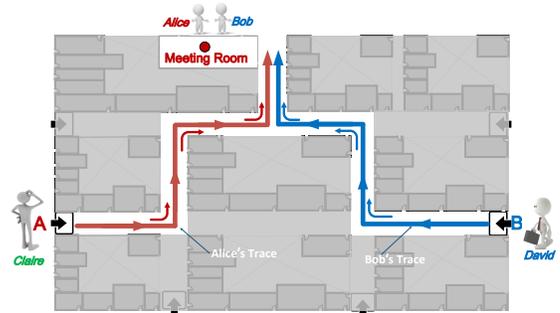


Fig. 1. A usage example of *ppNav*

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. 1, 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]. 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 Peer-to-Peer indoor Navigation system for smartphones. In *ppNav*, a leader records trace information along his travel along a path to a specific destination. Such information, including

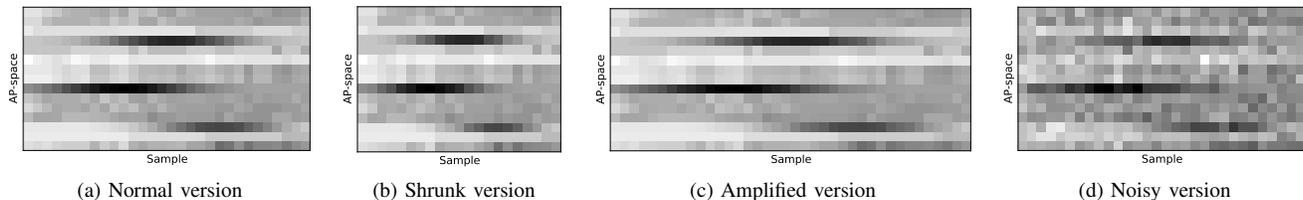


Fig. 2. Fingergrams of the same trace with (a) normal, (b) lower, (c) higher sampling frequency and (d) measurement noises

location specific features such as WiFi measurements and 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 entrance and demands an identical destination place, *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) 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. 2, 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 (section III-B and III-C) for trace synchronization and deviation detection.

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

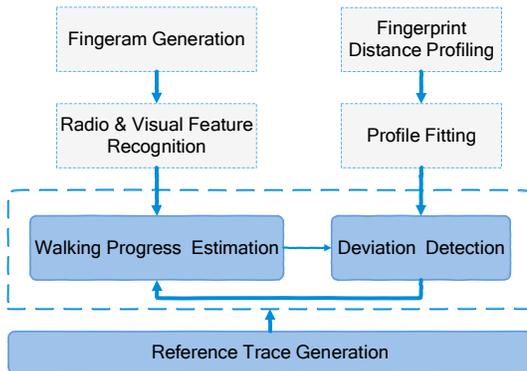


Fig. 3. System Architecture

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 crowdsourcing 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. 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 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. 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.

Reference Trace Generation. We sample accelerometer, 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.

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.

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.

III. 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], [9], [10], [11]. 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 [12], [13], 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 i th AP in the j th 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. 2a 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], [10], 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

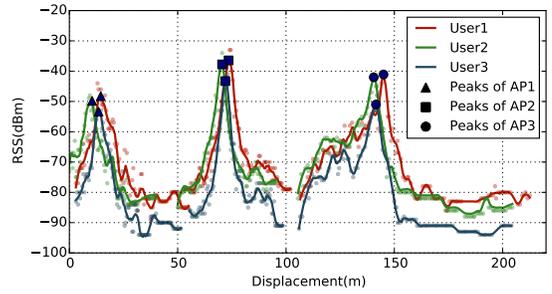


Fig. 4. RSS peaks appear to be stable regarding physical locations(RSS trends of 3 APs observed by different users).

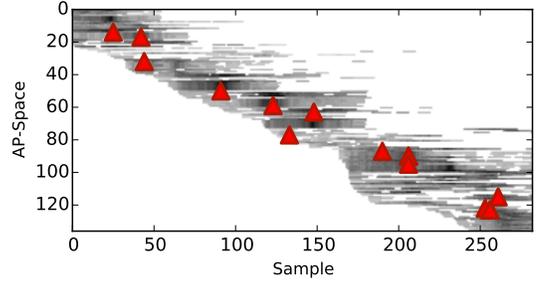


Fig. 5. Radio markers in fingeram

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, 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 [12], [13]. Recall Fig. 2, 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], [8]. 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. 4, 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. 5, consistent radio

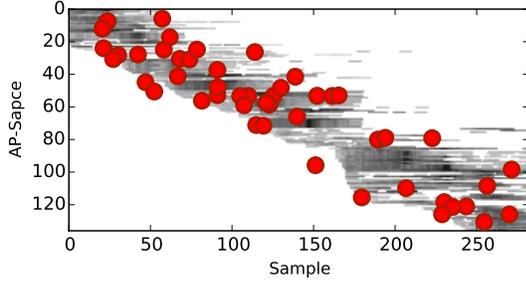


Fig. 6. Visual markers in fingeram

markers can be detected from the fingerams, even though they are distorted in lengths and noises.

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 [12], [13]. 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) [12], 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. 6, a considerable number of visual markers can be detected from fingerams, from which we can select reliable ones for walking progress estimation.

IV. PPNV DESIGN

A. Reference Trace Generation

When a user takes a walk along a path destined for a place, *ppNav* can automatically construct a reference trace $J_R = \langle F, E \rangle$ 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.

As the fingerprints are continuously measured during walking, we employ a low-pass filter to sift out partial uncertain

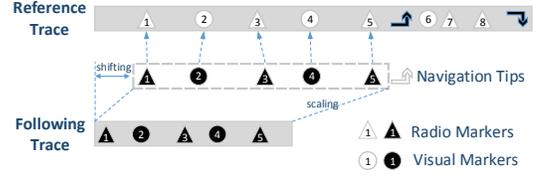


Fig. 7. Trace alignment by fingeram markers via scaling and shifting

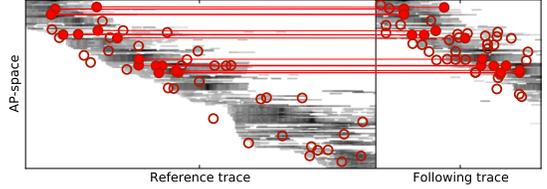


Fig. 8. Visual markers filtered by AP space (markers denoted by hollow circles are omitted)

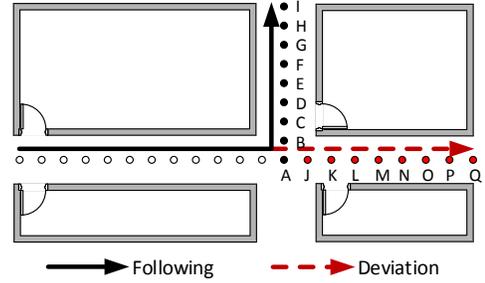


Fig. 9. Experimental scenario of fingerprint distance profile

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. As for motion event detection, various advanced techniques have been developed and can be used [6], [14], [15]. Hence in this section we mainly focus on the navigation design and leave a brief introduction on motion event detection in the implementation section.

B. Walking Progress Estimation

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

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. Suppose a user is locked on to a reference trace $J_R = \langle F_R, E_R \rangle$ and denote his own navigation trace as $J_N = \langle F_N, E_N \rangle$. 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 the i th sample in F_N and the k th 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 i th and k th column in F_N and F_R , respectively. Since no rotation is involved in the mapping, there are basically scaling and translation operations during

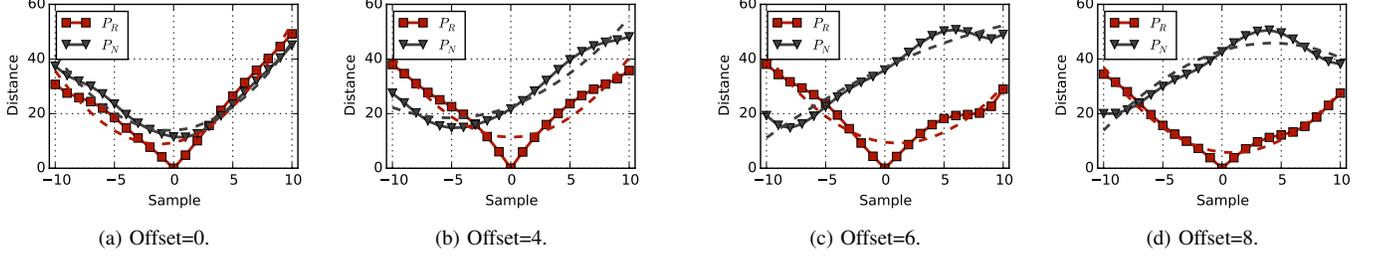


Fig. 10. Fingerprint distance profiles of different samples from deviation.

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. 7 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. 8, 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. 8 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 $V_N = \{\mathbf{v}_N^{(1)}, \mathbf{v}_N^{(2)}, \dots, \mathbf{v}_N^{(n_v)}\}$ and $V_R = \{\mathbf{v}_R^{(1)}, \mathbf{v}_R^{(2)}, \dots, \mathbf{v}_R^{(n_v)}\}$ 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 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 $R_N = \{\mathbf{r}_N^{(1)}, \mathbf{r}_N^{(2)}, \dots, \mathbf{r}_N^{(n_r)}\}$ where $\mathbf{r}_N^{(k)} = (r_N^{(k,x)}, r_N^{(k,y)})$ 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 $R_R = \{\mathbf{r}_R^{(1)}, \mathbf{r}_R^{(2)}, \dots, \mathbf{r}_R^{(n_r)}\}$. 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 [16]:

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

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 may deviate 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 \mathbf{f} against a series of nearby fingerprints observed on its opposite sites along the path, i.e., preceded and posterior fingerprints of \mathbf{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:

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

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 $P_R = \{\phi(\mathbf{f}_R^{(k')}, \mathbf{f}_R^{(k'+i)}), i \in [-d, d]\}$.

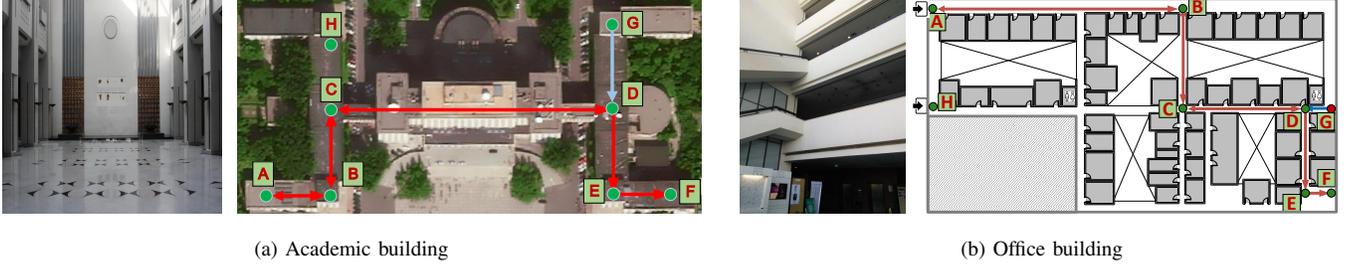


Fig. 11. Experiment environments in (a) academic building and (b) office building

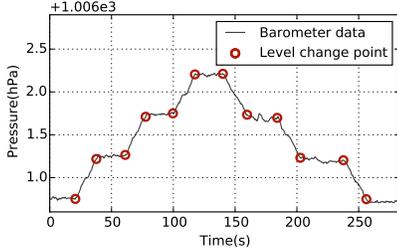


Fig. 12. Level-change detection

We conduct preliminary measurements to validate our observations. As shown in Fig. 9, 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. 10. 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. 10, 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:

$$\tilde{a} = \frac{a_N}{a_R}, \quad (3)$$

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 \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.

V. IMPLEMENTATION AND EXPERIMENTS

A. Experimental Setup

To evaluate *ppNav*, we build a prototype on commodity smartphones (Google Nexus 5 and Nexus 7) running Android

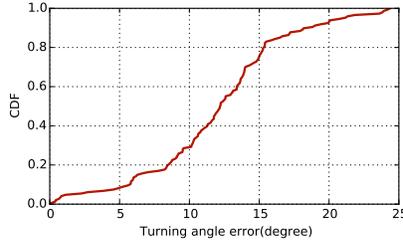


Fig. 13. Turning angle accuracy

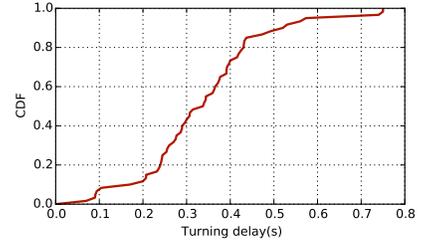


Fig. 14. Turning detection delay

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.11.

We recruit four volunteers to participate in our experiments. As shown in Fig. 11, 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*. Thus, we collect 12 navigation traces in all for evaluation. Note that the users naturally hold their smartphones in hand during walking.

B. Performance of Motion Event Detection

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. 12, the pressure values within the same level are relatively stable (with a maximum variation of 10Pa as measured in 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 [17]. 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 12 traces in all including going upstairs and downstairs in both experimental buildings, which contain 48 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.

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 [18], [19]. We acquire the ground truth turning angles from the floor plans. To extensively understand the turning angle estimation error and turning detection delay, we

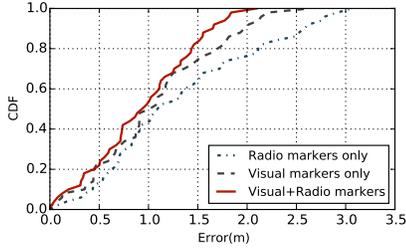


Fig. 15. Impacts of different markers

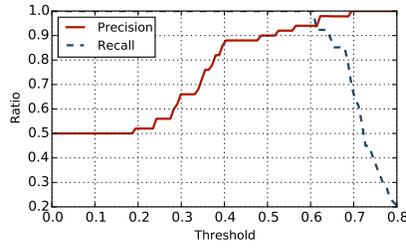


Fig. 16. Deviation detection rate

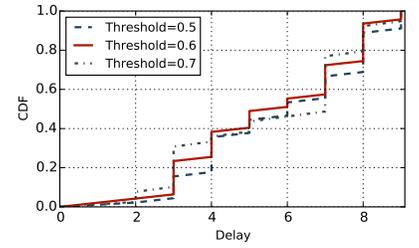


Fig. 17. Deviation detection delay

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. 13, *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. 14 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.

C. Performance of Trace Synchronization

We leverage a robust linear model (RLM) to mitigate the effects of noises, which uses iteratively reweighted least squares method [16] 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. We evaluate the effects of individual type of markers for trace alignment using the same trace data. Fig. 15 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.

D. Performance of Deviation Detection

Now we evaluate the delay of deviation detection according to the checkpoints marked by leaders and followers. Fig. 16 depicts the precision and recall of deviation detection all the 12 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. 17 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.

VI. DISCUSSION

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 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.

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 [8], our design is actually significantly more energy efficient.

Evolving to Generic Navigation Services. Rather than disposable navigation, *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 a accessible roadmap. To achieve this, we need to design path planning algorithms to find the shortest path for friendly navigation [8].

VII. 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 [20], ultrasound sensors [21], RFID [22], etc. Various approaches have been proposed to leverage the pervasive WiFi infrastructure for localization based on either ranging [23], [24] or fingerprinting [9], [11]. Other ubiquitously available radio signals such as FM [10], GSM [2] are also exploited as fingerprints. Thanks to the prosperity of mobile sensing, location fingerprints are recently further extended to magnetism [25], sound [4], and multi-modal ambient features [11], [26]. These systems either rely on specialized hardware or require labor-intensive site survey. Recent ranging-based localization based on sound [3] or WiFi [27] achieve high accuracy, but they need cooperation between phones or are not readily applicable on commercial phones.

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], [28]. 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], [29]. Appropriate mechanisms need to be designed to guarantee the precision of resulted maps as well as encourage user participation [30]. 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], [25]. [25] navigates blind users using customized device for magnetic sensing. Two recent system, Travi-Navi [8] and FOLLOWME [31], also employs 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, 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 the user just need to be asked to hold the smartphone in hands during navigation. Second, 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.

VIII. 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.

IX. ACKNOWLEDGEMENT

This work is supported in part by NSFC under grant 61522110,61572366, and Beijing Nova Program under grant Z151100000315090.

REFERENCES

- [1] M. Youssef and A. Agrawala, "Handling samples correlation in the horus system," in *Proceedings of IEEE INFOCOM*, 2004.
- [2] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-id sequence matching," in *Proceedings of ACM MobiSys*, 2011.
- [3] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained indoor localization via smartphone," in *Proceedings of ACM MobiSys*, 2013.
- [4] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proceedings of ACM MobiSys*, 2011.
- [5] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *Mobile Computing, IEEE Transactions on*, vol. 14, no. 2, pp. 444–457, 2015.
- [6] Y. Jiang, Y. Xiang, X. Pan, K. Li, Q. Lv, R. P. Dick, L. Shang, and M. Hannigan, "Hallway Based Automatic Indoor Floorplan Construction Using Room Fingerprints," in *Proceedings of ACM UbiComp*, 2013.
- [7] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: Indoor pathway mapping made easy," in *Proceedings of USENIX NSDI*, 2013.
- [8] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-navi: Self-deployable indoor navigation system," in *Proceedings of ACM MobiCom*, 2014.
- [9] 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 *Proceedings of ACM MobiCom*, 2012.
- [10] S. Yoon, K. Lee, and I. Rhee, "Fm-based indoor localization via automatic fingerprint db construction and matching," in *Proceedings of ACM MobiSys*, 2013.
- [11] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu, "Enhancing wifi-based localization with visual clues," in *Proceedings of ACM UbiComp*, 2015.
- [12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [14] A. Brajdic and R. Harle, "Walk Detection and Step Counting on Unconstrained Smartphones," in *Proceedings of ACM UbiComp*, 2013.
- [15] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors," in *Proceedings of ACM UbiComp*, 2012.
- [16] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics-theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.
- [17] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, "Barometric phone sensors: More hype than hope!" in *Proceedings of ACM HotMobile*, 2014.
- [18] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proceedings of ACM MobiCom*, 2014.
- [19] 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 Computing Surveys*, vol. 47, no. 3, pp. 54:1–54:34, Apr. 2015.
- [20] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.
- [21] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of ACM MobiCom*, 2000.
- [22] J. Wang and D. Katabi, "Dude, where's my card?: Rfid positioning that works with multipath and non-line of sight," in *Proceedings of ACM SIGCOMM*, 2013.
- [23] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of ACM MobiCom*, 2010.
- [24] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *Proceedings of ACM MobiSys*, 2013.
- [25] T. H. Riehle, S. M. Anderson, P. A. Lichter, N. A. Giudice, S. I. Sheikh, R. J. Knuesel, D. T. Kollmann, and D. S. Hedin, "Indoor magnetic navigation for the blind," in *Proceedings of IEEE EMBC*, 2012.
- [26] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proceedings of ACM MobiCom*, 2009.
- [27] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proceedings of ACM SIGCOMM*, 2015.
- [28] 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 *Proceedings of IEEE INFOCOM*, April 2015.
- [29] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.
- [30] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.
- [31] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-mile navigation using smartphones," in *Proceedings of ACM MobiCom*, 2015.