Indoor Navigation With Virtual Graph Representation: Exploiting Peak Intensities of Unmodulated Luminaries

Wenping Liu, *Senior Member, IEEE*, Hongbo Jiang, *Senior Member, IEEE*, Guoyin Jiang, *Member, IEEE*, Jiangchuan Liu, *Fellow, IEEE*, Xiaoqiang Ma, *Member, IEEE*, Yufu Jia, and Fu Xiao

***Abstract*— The ubiquitous luminaries provide a new dimension for indoor navigation, as they are often well-structured and the visible light is reliable for its multipath-free nature. However, existing visible light-based technologies, which are generally frequency-based, require the modulation on light sources, modification to the device, or mounting extra devices. The combination of the cost-extensive floor map and the localization system with constraints on customized hardwares for capturing the flashing frequencies, no doubt, hinders the deployment of indoor navigation systems at scale in, nowadays, smart cities. In this paper, we provide a new perspective of indoor navigation on top of the virtual graph representation. The main idea of our proposed navigation system, named PILOT, stems from exploiting the peak intensities of ubiquitous unmodulated luminaries. In PILOT, the pedestrian paths with enriched sensory data are organically integrated to derive a meaningful graph, where each vertex corresponds to a light source and pairwise adjacent vertices (or light sources) form an edge with a computed length and direction. The graph, then, serves as a global reference frame for indoor navigation while avoiding the usage of pre-deployed floor maps, localization systems, or additional hardwares. We have implemented a prototype of PILOT on the Android platform, and extensive experiments in typical indoor environments demonstrate its effectiveness and efficiency.**

***Index Terms*— Human computer interaction, computational and artificial intelligence, pervasive computing, ubiquitous computing.**

I. Introduction

OFFERING indoor navigation services is of crucial importance, which is a common welfare in complex indoor environments for many involved users such as shop owners and shoppers in smart cities with ubiquitous smart urban infrastructures. Recently, visible light has become a key enabler for indoor navigation, owing to its high reliability immune from multi-path and the light sources are ubiquitous and well-structured in indoor environments. For instance, Carrefour supermarket in Lille, France, uses the Philips connected lighting system to facilitate customers' navigation and triggers aisle-specific special offers (e.g., discounts/promotions) around them, which indeed helps shoppers save shopping time and expenditure while in return increases the sales volume.[1] The potential huge market of mobile industry naturally attracts extensive studies on location-based services (LBS) [1] and LBS related applications such as navigation, location-aware marketing [2], social recommendation [3], [4], and privacy-protecting [5].

The indoor navigation systems with the prerequisite of map information and accurate localization have been well studied [6], [7]. One notorious problem is that the floor map is labor-extensive to acquire or keep up-to-date. An alternative solution could be via crowdsourcing [8]–[10], yet it is time-consuming [6], lack of details [11] and difficult to guarantee the navigation success owing to the complex structure. For indoor localization, techniques exploiting such ambient information as WiFi/magnetic signal [12]–[15] usually suffer from low accuracy for navigation, while visible light localization technologies usually require modulating flashing frequency of light-emitting diode (LED) lights and modifications to COTS mobile devices (or mounting extra devices) [16]–[21], or is energy inefficient [22] to capture the characteristic frequency of unmodified fluorescent lights via the back-end camera. Different from above approaches, NaviLight [23] exploits light intensities from unmodulated light sources as fingerprints, yet potentially it suffers from many practical issues such as room temperature, device diversity, altitude, etc. iLAMP [24]

[1] http://www.lighting.philips.com/main/systems/connected-lighting.html.

Manuscript received June 19, 2017; revised June 6, 2018 and September 19, 2018; accepted November 18, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor B. Krishnamachari. Date of publication December 20, 2018; date of current version February 14, 2019. This work was supported in part by the National Natural Science Foundation of China under Grants 61672213, 61732017, 61572219, 71671060, and 61702204, in part by the Excellent Young and Middle-Aged Scientific and Technological Innovation Team for Universities in Hubei Province under Grant T201714, in part by the Science and Technology Plan Projects of Wuhan City under Grant 2017050304010325, in part by the Fundamental Research Funds for the Central Universities, in part by the Research Project of Science and Technology Department of Hubei Education Department under Grant B2013033, and in part by the Youth Foundation Project of Hubei University of Economics under Grant XJ201316. The work of J. Liu was supported in part by by an Industrial Canada Technology Demonstration Program (TDP) Grant and in part by an NSERC Discovery Grant. *(Wenping Liu, Hongbo Jiang, and Guoyin Jiang contributed equally to this work.) (Corresponding authors: Hongbo Jiang; Guoyin Jiang.)*

W. Liu is with the School of Information Management and Statistics, Hubei University of Economics, Wuhan 430205, China, also with the CICCPREA-Hubei Sub-Center, Wuhan 430205, China, and also with Hunan University, Changsha 410082, China.

H. Jiang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: hongbojiang2004@gmail.com).

G. Jiang is with the School of Public Administration, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: jiangguoyin@uestc.edu.cn).

J. Liu is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.

X. Ma is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China.

Y. Jia is with the School of Information Management and Statistics, Hubei University of Economics, Wuhan 430205, China, and also with the CICCPREA-Hubei Sub-Center, China.

F. Xiao is with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China.

Digital Object Identifier 10.1109/TNET.2018.2884088

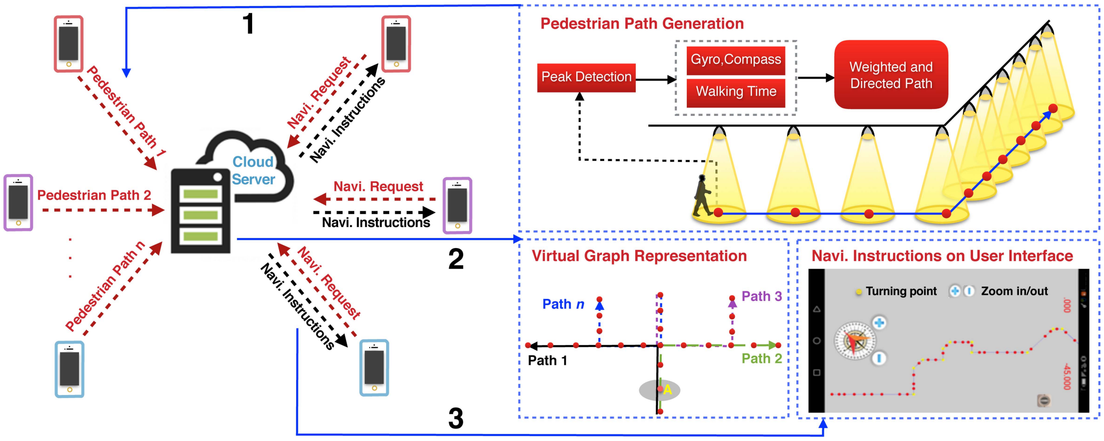


Fig. 1. Illustrative diagram of PILOT. Each participant uploads a pedestrian path which consists of vertices (corresponding to peaks) and weighted and directed edges between consecutive peaks. The server integrates uploaded paths by detecting overlapped segments and generates a virtual graph. During navigation, each user at any point of an affiliated space (indicated by the shaded region where a peak intensity from the nearest light source is detected) is affiliated with the nearest vertex (e.g., Vertex A) of the graph for progress tracking. The navigation path with instructions is displayed on the user's mobile device.

explores the energy-unfriendly camera for indoor localization, which will no doubt raise the issue of privacy leakage. A recent study named RETRO [25] exploits the retroreflected optical signal as location signature for indoor localization; the undesirability is that it requires multiple photodiodes mounted on light sources and other special devices (e.g., retroreflector and LCD shutter) mounted on passive IoT devices. These practical issues entail great challenges for the deployment of indoor navigation.

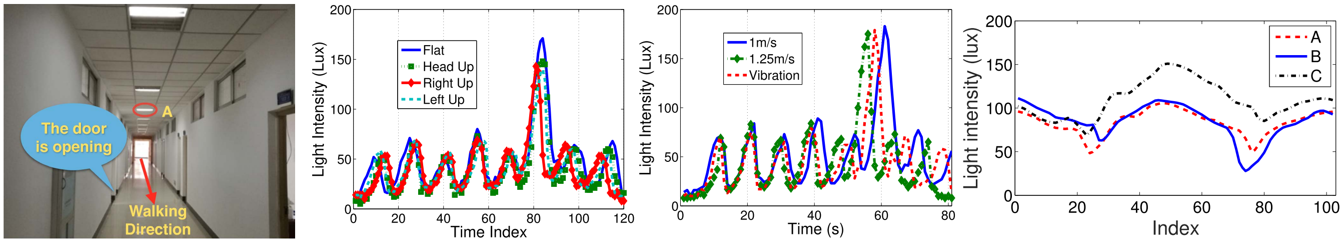
In this paper, we present and answer the following questions with respect to indoor navigation:

- *Are the floor map and the localization system necessary for indoor navigation?* No. The objective of navigation is simply to find a reference path to the destination, including the instruction for next movement. We envision that, with a mapping from the walkable space to a virtual graph, the navigation path can be computed on top of this graph. As such a navigation user only needs to calculate the nearest vertex and it does not matter where s/he is.
- *Is it practical to find the suitable mapping for indoor navigation?* The answer is affirmative. The light fixtures can be found everywhere in a typical indoor space.[2] Our primary experimental evidences show a one-to-one correspondence between *luminaries* and peak intensities (or simply *peaks*) of lights, detected by light sensors of commercial off-the-shelf (COTS) mobile devices when people walk under luminaries. A virtual graph of peaks is thus derived wherein each peak (or light source) corresponds to a vertex and two consecutive peaks form an edge. Such a graph can be used for successful navigation since it correlates with the underlying walkable space.
- *Is a weighted and directed graph helpful for indoor navigation?* Yes. To find an optimal navigation path (e.g, with the minimum walking time), each edge of the graph should have a length as the weight. For the visual navigation, the edge direction can be used for prompting when and where to change the walking direction.
- *Should each vertex be named (e.g., with manual tags or the frequencies of luminaries) for navigation?* Not a necessity. Capturing the flashing frequency of LED lights usually requires modulated luminaries and the modified devices, while the characteristic frequency of fluorescent lights tends to be dynamic with time and temperature, making the frequency-based naming scheme impractical. Another solution is through the manual tag, which is also too costly to be practical; by naming the points of interests in indoor space, the user can be affiliated with the exact location of a nearby vertex such that indoor navigation is enabled with the floor map. By contrast, our system affiliates each user with a vertex in the generated virtual graph; when a user (e.g., Alice) requires the navigation service to reach the destination (e.g., Alice's friend Bob) which is also affiliated with a vertex in the graph, the system can simply track the progress by counting the number of detected peaks on the computed navigation path, and manually tagging the vertices is not necessary in this case. Note that when Alice wants to visit a place (e.g, a shop) instead of a friend, the vertex corresponding to the destination/place should be manually tagged for offering accurate navigation service, which can be readily done by self-motivated shop owners to increase their sales volume, while other vertices are not needed to be tagged.

With these motivations, in this paper, we provide a new perspective of indoor navigation on top of virtual graph (VG) representation by exploiting peak intensities of ubiquitous lights (i.e. unmodulated luminaries). We develop PILOT, as shown in Fig. 1, which is a novel indoor navigation system based on Peak Intensity of Light induced netwOrk Topology. Specifically, in the VG (shown as "Virtual Graph Representation" in Fig. 1), each vertex (i.e., a peak) corresponds to a luminary, and two adjacent vertices form an edge. Besides, to offer a short navigation path with visual instructions, we assign a length and a direction to each edge. Just like the maps for GPS-based outdoor navigation systems, the virtual graph serves as a global reference frame (GRF) [26] for indoor navigation, a particular perspective for motion planing [27].

The proposed PILOT system consists of three components: the participants, cloud server and navigation users (see Fig. 1). Each participant/pedestrian (e.g., a shop owners/worker) turns

[2] Note that the luminary layout is not a good option for navigation since, it is costly to acquire, and most importantly, it does not correlate with the walkable space: two adjacent luminaries may not be directly reachable by people if there is an obstacle (e.g., wall, furniture, etc.) between them, leading to a failed navigation.



(a) (b) (c) (d)

Fig. 2. An Illustrative example. (a) A corridor where the door near light A is opening; (b) The impact of device orientations; (c) The impact of walking speeds; (d) The impact of relative positions under luminaries.

on the PILOT client side application on his/her COTS mobile device while walking in the indoor space of interests. The device periodically collects light intensity, as well as readings form rich built-in sensors such as barometer, gyroscope and compass, thereby uploading the pedestrian path with sensory data to the cloud server when the participant reaches the destination (shown as "Pedestrian Path Generation" in Fig. 1). The cloud server then merges the pedestrian paths to form a virtual graph representing the (unknown) luminary layout by computing the overlapped segment of pairwise paths. When receiving a navigation request from a user, the server first computes the vertex (in the virtual graph) nearest to the user, and then generates a navigation path with visual instructions on the user's device. During navigation progress, PILOT tracks the user (i.e., which vertex on the virtual graph) and prompts visual navigation instructions (shown as "Navi. Instructions on User Interface" in Fig. 1).

We highlight that several salient features of PILOT lead to its wider applicability as compared with existing indoor navigation systems [6], [7], [28], [29]. First, it is zero-effort as we exploit peak intensities of unmodulated luminaries in a crowdsensing fashion. Second, it has no reliance on any floor map or localization system. Instead, it collects readings form rich built-in sensors to assist virtual graph generation. Third, it provides location-free navigation services as there are no constrains on start/end locations for navigation. Fourth, it has no constrains on people's walking patterns, and does not require modulated luminaries or modified mobile devices.

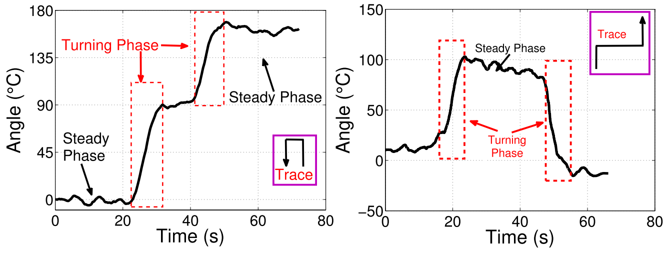
The reminder of this paper is organized as follows. We introduce the background and motivations in Section II. In Section III we detail the system design of PILOT, and evaluate the performance in Section IV. We introduce related work in Section V and conclude this paper in Section VI.

II. Exploiting Peak Intensity Topology as Virtual Graph Representation for Indoor Navigation

As mentioned above, in this paper we exploit the peaks of visible light from luminaries for virtual graph construction. Note that it is not new to leverage peaks or trend of such ambient signals as WiFi [10], [30], [31], geomagnetism [6], [7], etc., for indoor localization and location-based services (LBS). For instance, Walkie-Markie [10] explores the fact that when walking along a pathway, there is a trend of the received signal strength (RSS) from a master WiFi AP changing from increasing to decreasing, and the location corresponding to the trend serves as a WiFi-Mark for inferring the pathway map based on the crowdsourced trajectories. However, WiFi scanning is energy-hungry, while reducing scanning period may lead to less WiFi-Marks being identified, thereby lowering the quality of the pathway map; if each user is tasked to sense the WiFi signals for only a few minutes, more users will be involved for generating sufficient trajectories. Furthermore, if WiFi-Marks are used for indoor navigation, the power consumption for tracking users during navigation process, especially in a giant shopping center, can be very large. When the geomagnetic field measurements are used for indoor LBS [6], [7], the computation complexity of dynamic time warping (DTW) for trace synchronization can be very high.

In indoor environments, one key observation is that the luminaries are usually fixed on ceilings, which can be linked properly to topologically form a graph based on light sources, while the light sensors for sensing intensities are much more energy-efficient than WiFi modules. Just like the electronic maps in GPS-based navigation systems, luminaries, together with their adjacency relationship, hold potential of serving as a global reference frame, thereby being a good candidate for designing indoor navigation systems. As two geographically close luminaries may not be directly reachable from each other (e.g., due to the wall between them), the walkable space between adjacent luminaries does not necessarily match with the luminary layout (i.e., the Euclidean space formed by the luminaries). Consequently, a naive way to exploit the layout of the real luminary placement may result in an unreachable navigation. In this paper, our primary objectivities are to find a one-to-one correspondence between luminaries and peak intensities, and identify two consecutive peaks corresponding two adjacent luminaries such that people can directly walk from one luminary to the other.

To that end, accurately finding unmodulated luminaries, under which the user is currently walking, no doubt lies as the first step before exploring them for navigation. As mentioned before, the frequency capturing techniques suffer from many practical issues (e.g., modulation on LED light). Fortunately, the propagation model of illumination [32] allows us to exploit the light intensity for luminary detection. Specifically, the illumination follows an inverse square distance law of the radiant intensity, such that when a person approaches a luminary and moves away, an increasing trend, a peak, and a decreasing trend of light intensity will be observed sequentially by the light sensor of the COTS mobile device. Please see Fig. 2 as an illustrative example. Even though the absolute light intensity varies when the device is held with different orientations (e.g., flat, head up, left-side up, and right-side up in Fig. 2 (b)), when people walk at different speeds or the device is shaking during walking (please see Fig. 2 (c)), we observe that the eight peaks occurring roughly at the same time, and the peak does not suffer from ambient noise (e.g., the open door near light A in Fig. 2 (a)). In addition, when walking under a luminary with different positions relative to the location right below the luminary, there is always one (and only one) detected peak corresponding to the luminary (please see Fig. 2 (d)).



(a) (b)

Fig. 3. Illustrative examples. (a) The walking directions of a U-turn shaped trace; (b) The walking directions of a right-and-left-turn shaped trace.

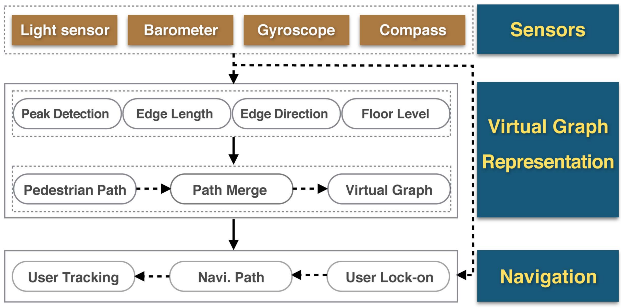


Fig. 4. The architecture of PILOT.

These advantages show that *the peak intensity, detected by the user-carried mobile device, is a robust indicator of the corresponding luminary*.

After finding the one-to-one correspondence between peaks and luminaries, we now connect these peaks to yield a pedestrian path for each participant. Specifically, we draw an edge between consecutive peaks/vertices with the walking time as the weight (i.e., edge length). The edge direction is computed based on the gyroscope and compass readings. Fig. 3 depicts the gyroscope data over time with different walking traces. We observe a sudden change of gyroscope readings when there is a sharp turn while remaining relatively steady when walking straight, showing the feasibility of gyroscope readings for edge direction computation. By incorporating compass readings, we can derive the absolute direction of the edges, which is used in overlapped segment identification for merging multiple paths. Eventually, *the virtual graph of peaks, which indicates the neighboring relationship between luminaries, can be organically grown*, noting that this process does rely on the floor map, localization system, or luminary placement.

As will be shown in Section III, the sharp peak of light intensity helps to affiliate users with the virtual graph during navigation progress. At the same time, the constructed virtual graph well represents the walkable space such that two luminaries, geographically nearby but separated by obstacles and thus not directly reachable, are far away in the graph. Desirably, on top of the virtual graph, a path for detouring obstacles can be computed to guarantee navigation success.

To implement PILOT in practice, there are many challenges to be addressed: 1) The fluctuations of light intensity, e.g., caused by users' sway, such reflectors as walls and mirrors, interference from nearby luminaries, and so on, can incur false-positive results (i.e., false peaks), potentially leading to an inaccurate navigation path/instruction; 2) The noisy gyroscope and compass readings, especially the presence of accumulative errors, make it highly non-trivial to compute the edge direction; 3) Without a vertex identifier or fingerprint, it is rather difficult to compute the overlapped segments for merging multiple pedestrian paths, lock on users, and detect deviation during navigation. We will describe how PILOT tackles all these challenges in Section III.

III. System Design

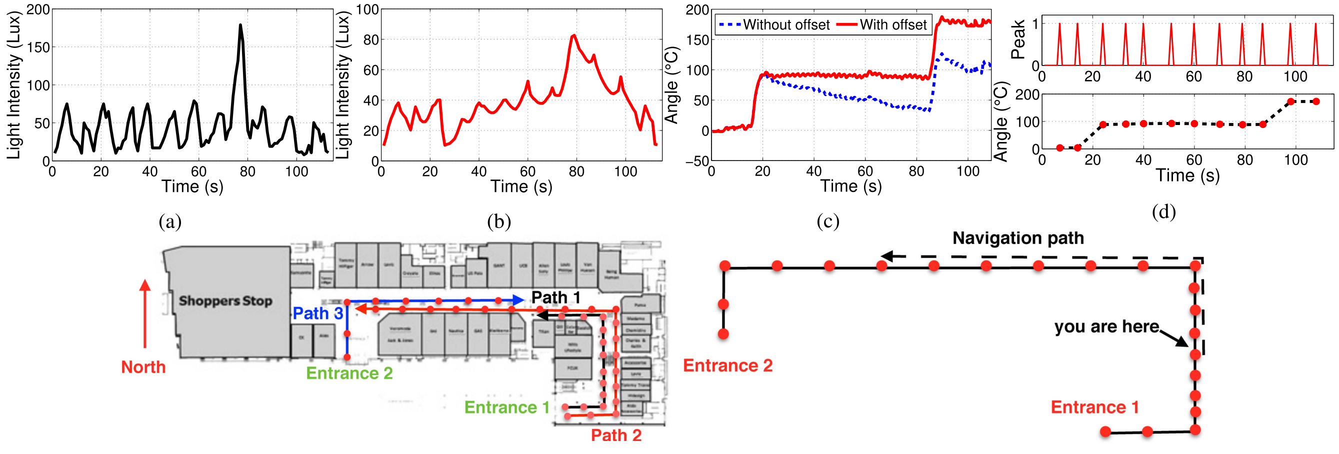
A. An Overview

The prototype implementation of PILOT consists of two main modules, namely *Virtual Graph Generation module* and *Navigation module*, as shown in Fig. 4. In the Virtual Graph Generation module (Section III-B), each crowdsensing participant with a COTS mobile device collects the necessary sensory data, forms a pedestrian path which consists of vertices (i.e., peaks) and directed edges, and uploads it to a cloud server. The server then merges paths from different participants by computing the overlapped segments to form a virtual graph. In the navigation module (Section III-C), PILOT affiliates the user requesting navigation services with the virtual graph, thereby computing a directed and weighted navigation path on the graph, in order to guide the user to the destination. During the walking progress, PILOT keeps tracking the user (i.e., which vertex on the virtual graph) and prompts visual navigation instructions. If the user veers off the navigation path, PILOT will send an alert to the user, and guide her/him back to the original path or computes a new navigation path, depending on which one is shorter to the destination.

Let us take Fig. 5 as an example to describe how PILOT works. The mobile device of each crowdsensing participant continuously measures the light intensity, gyroscope and compass readings. PILOT exploits such techniques as IIR filter and moving average technique to smooth noisy light intensities (see Fig. 5 (b)), and identify peaks (Section III-B1) based on normalization process (shown in Fig. 5 (d)). A pedestrian path is then constructed where each peak corresponds to a vertex, and for the edge between two consecutive peaks (i.e., vertices), we assign the walking time between the two peaks as its length, and relative direction (shown in Fig. 5 (d)) based on gyroscope readings (shown in Fig. 5 (c)) and absolute direction based on compass readings (Section III-B2). Based on the barometer readings of the vertices on the path, we detect the floor level change for cross-level navigation (Section III-B3).

To compute the overlapped segment between two paths, PILOT exploits the absolute direction of edges, and the number of vertices to determine whether and where two paths overlap, and then applies the dynamic time warping (DTW) technique [33] on the compass readings for further verification (Section III-B4). For instance, in Fig. 5 (e), from Path 1 and Path 2, we can infer that both of Participant 1 and Participant 2 first walk toward East until detecting five peaks, and then make a left turn to the direction of North. After that, they walk towards West after detecting five peaks and making another left turn. As such, we can compute the overlapped segments between Path 1 and Path 2. With more paths information, we can compute the overlapped segments (if any) between pairwise paths (e.g., Path 2 and Path 3). Eventually, multiple paths with overlapped segments are merged to construct the virtual graph, as shown in Fig. 5 (f).

When another user requests navigation services, PILOT first affiliates the user with the virtual graph by matching the compass readings with the edge direction in the virtual graph. With the given destination (e.g., the user's friend) on the graph



(a) (b) (c) (d) (e) (f)

Fig. 5. An overview of PILOT. (a) The raw data of light intensity; (b) The smoothed data; (c) The curves of gyroscope readings vs. time with/without offset; (d) The intensity normalization curve (upper) and the adjusted angle curve when peaks (in red solid circles) are detected (lower); (e) The scenario where three pedestrian paths are merged; (f) The constructed virtual graph and a navigation path (indicated by the dashed line with an arrow).

as the input, a navigation path with the minimum walking time is derived, and a visual guiding route is displayed on the device's screen. After that, PILOT shows the instruction for the next movement and keeps track of the user by detecting peaks and walking direction until s/he arrives at the destination.

B. Virtual Graph Generation

In PILOT, a number of participants help to generate the virtual graph. Each participant turns on the PILOT system while holding the mobile device at his/her hand heading toward the walking direction.[3] The mobile device then periodically senses the light intensity from the light bulbs above the walkable space, and collects the gyroscope and compass data. When detecting a peak intensity, it computes the walking direction based on the gyroscope and compass data, and tracks the floor level change via the barometer. For each pedestrian path, PILOT first detects peak intensities corresponding to the vertices, and then computes the length of edges that link neighboring vertices and the difference of gyroscope readings between neighboring vertices to indicate the relative direction of each edge. In this way, a path by each participant is established. Multiple paths from participants are merged to form the virtual graph on top of the detected peaks, where overlapped segments need to be found out. Furthermore, considering multi-floor buildings, the floor level change should be identified.

1) Peak Intensity Detection: Note that the light sensor readings inevitably exhibit noises (e.g, caused by surface reflection, movement sway, etc.). Our peak intensity detection includes the following three steps:

IIR Filter: For the collected intensity sequence $\{x_1, x_2, \cdots, x_n\}$, we first apply the one-order IIR (Infinite Impulse Response) filter to remove the high-frequency components and derive the filtered sequence $\{\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_n\}$.

$$\overline{x}_k = (1.0 - \delta)\overline{x}_{k-1} + \delta x_k \tag{1}$$

where $\delta \in (0, 1)$ is the filtering coefficient, and $\overline{x}_0 = 0$. In our experiments we set $\delta = 0.8$.

Weighted Moving Average: We compute the l-item weighted moving average of the sequence $\{\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_n\}$ as follows.

$$\tilde{x}^l_k = \sum_{i=k-(l-1)/2}^{i=k+(l-1)/2} \alpha_i \overline{x}_i \tag{2}$$

where $\alpha_1, \alpha_2, \cdots, \alpha_l \in (0, 1)$ are the coefficients such that $\alpha_1 + \alpha_2 + \cdots + \alpha_l = 1$. In this paper, we simply set $l = 5$, and $\alpha_1 = \alpha_2 = \cdots = \alpha_5 = 0.2$.

Intensity Normalization: Define the forward difference operator as $\triangleleft\tilde{x}^l_k = \tilde{x}^l_k - \tilde{x}^l_{k-1}$, and the backward difference operator as $\triangleright\tilde{x}^l_k = \tilde{x}^l_k - \tilde{x}^l_{k+1} (k < n)$. Let $I_{\mathcal{A}}(x)$ be an indicator function such that for a set $\mathcal{A}$, $I_{\mathcal{A}}(x) = 1$ if and only if $x \in \mathcal{A}$. For any k, if the following equation holds:

$$I_{[0,+\infty]}(\triangleleft\tilde{x}^l_k) I_{[0,+\infty]}(\triangleleft\tilde{x}^l_k \times \triangleright\tilde{x}^l_k) = 1 \tag{3}$$

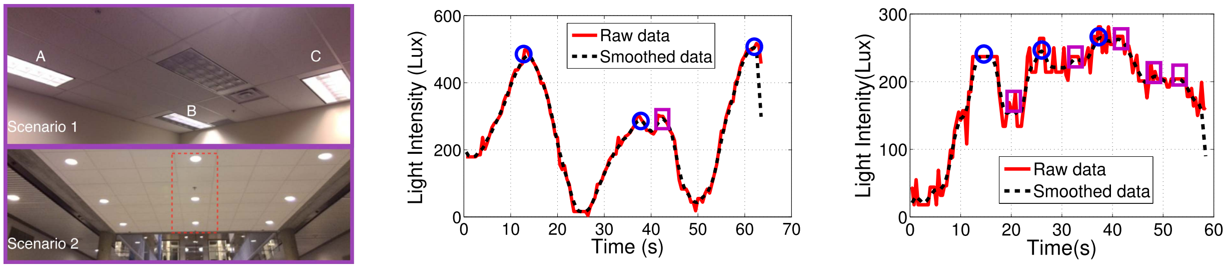
then $\tilde{x}^l_k$ corresponds to a peak, and time k is a peak time.

In most cases, the above-mentioned approach can correctly identify the luminaries; an exception may occur if the participants make a turn right below a light source, when two luminaries are very close or the participant walks very quickly. As such, there might be a violation of the one-to-one correspondence between peaks and light sources. Please see Fig. 6 as an illustrative example. Fig. 6 (b) depicts that a false peak (the third peak) occurs after the participant makes a right turn under the light B in Scenario 1. From Fig. 6 (c) we also observe that when there are multiple rows of luminaries in Scenario 2, the false positive (FP) results become even more. As such, we first compute the valley of light intensity before a peak occurs, and then compare the difference between the value of peak intensity and the valley intensity with a given threshold (e.g., 50 Lux). If the former is larger, then the peak is true, otherwise it's not. we can see from Fig. 6 (b) and Fig. 6 (c) that after this process, only true peaks are identified.

2) Edge Computation: To derive a global reference frame for indoor navigation, we build a weighted peak intensity topology (virtual graph) by identifying edges, and computing the length and direction for each edge.

Edge Length Computation: To construct a global reference frame for visual navigation, we propose to draw edges between pairwise vertices corresponding to two consecutive peaks, and

[3] When people step into an unfamiliar indoor space and use the navigation service, the smart device is often held at hand such that people can get prompt instructions to avoid veering off the course. This is a reasonable assumption in visible light based localization systems such as [17], [18], [20], [21], and [34].



(a)

(b)

(c)

Fig. 6. Illustrative examples on false positive (FP) and false negative (FN) results. (a) Two investigated scenarios with a right turn (upper) and multi-row of luminaries (lower), respectively; (b) The presence of false peaks (indicated by the rectangle) after making a turning under light B in Scenario 1, and eventually the other three peaks (indicated by the circles) are identified; (c) There are many false peaks (indicated by the rectangles) while only three true peaks are identified (indicated by the circles), corresponding to the three luminaries indicated by the dashed rectangle of the middle row in Scenario 2.

Algorithm 1 The Pseudo-Code for the Shift Offset

```
**Input:** the gryoscope readings sequence $\{g_0, g_1, g_2, \cdots, g_m\}$, $C_{offset}$,dt (time interval);
**Output:** relative angle sequence $\{A_1, \cdots, A_m\}$;
**Initialize:**$S_{offset} = 0.00$;
for i=1:m do
$S_{offset} = C_{offset} \times g_i + (1 - C_{offset}) \times g_{offset}$;
$v_i = g_i - S_{offset}$;
$a_i = a_i + v_i \times dt$;
$A_i = a_i$;
end for
```

derive a weighted and directed topology. To this end, we first compute the edge length. One intuitive way is to compute the distance from one vertex to another. This can be done by calculating the approximated displacement between consecutive peaks, based on the multiplication of step number and the participant's step size as described in [35]. One drawback of the method is that it requires participants' input of such private information as participants' height or gender for estimating step size [36], and participants may be reluctant to disclose such information due to privacy issues. An alternative way is to compute the double integration of the accelerometer readings [26], whereas there will be a significant accumulative error. In this paper, as our objectivity is to provide an optimal path with the minimum walking time, we thus simply regard the walking time from one peak to the next as the edge length. One challenge is that different participants may have different walking speeds and stride lengths, making the computed length less valuable. We address this issue in Section III-B4.

The Computation of Edge Directions: To promptly provide correct instructions for next movements, we need to compute the relative direction of a vertex to its precedent, i.e., the direction of an edge, which is crucial for the visual navigation instructions. To that end, we use the gyroscope readings collected by the users during walking. As the gyroscope sensor suffers from accumulative shift errors, we introduce a mechanic to offset the shift as described in Algorithm 1. Fig. 7 shows the evolution of the angle with time t where the smart device which is equipped with the LSM330 Gyroscope sensor and placed unmoved on the desk. We observe that there is an accumulative error with or without offset, while the error after an offset procedure where $C_{offset} = 0.005$ becomes stable very quickly. Even though there is still an error of 2 degrees, it will not affect the edge direction since generally a turning action will happen within 2 seconds (please see Fig. 8) such that the shift can be ignored. With the new mechanism, we continuously compute the walking directions based on the gyroscope readings. Generally speaking, when the users make a left/right turn, the gyroscope readings may vary greatly. Figs. 3 (a) (b) describe two different traces (indicated by the curves with arrows in the rectangles) and the corresponding walking directions based on the gyroscope readings. Clearly, we observe that the walking traces consist of a set of turning phases (indicating a left/right turn) and steady phases (i.e., going straight without turning actions).

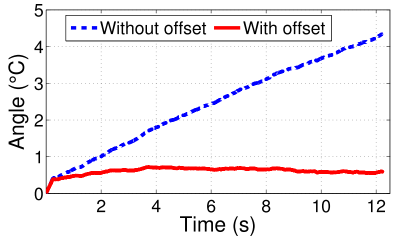


Fig. 7. The gyroscope readings over time with/without offset.

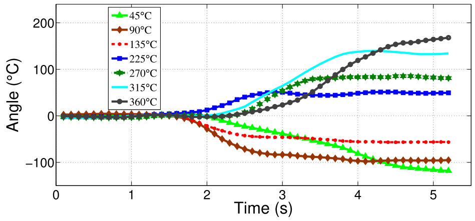


Fig. 8. The turning time for different turning angles.

However, we also notice from Fig. 3 (a) (b) that, the sway of the mobile devices during users' walking, the reading noise of gyroscope, and the turning actions, etc., will lead to a highly fluctuated gyroscope readings even when the user is walking straight. Without considering these issues, we may obtain a *squiggly* topology when the users actually walk in a straight corridor, and derive a very complex graph which does not correctly reflect the true topology of the walkable space. To address this issue, we observe that usually the distance between consecutive peaks/lights is usually 3-5 meters such that the gyroscope readings will not vary significantly, and thus propose to identify the *turning edge* in the following way.

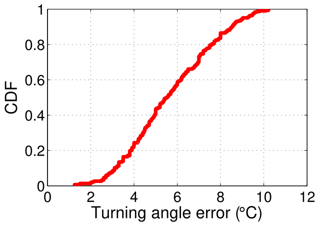


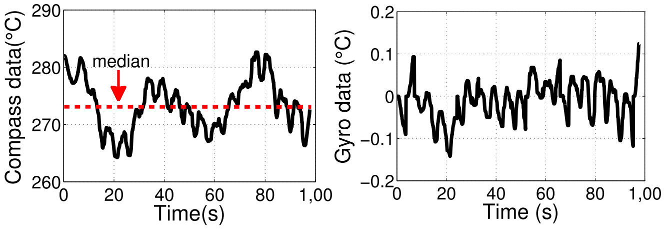
Fig. 9. The turning angle errors.

Definition 1: For two neighboring vertices v_i and v_j, let the angles corresponding to the peak time t_i and t_j ($t_i < t_j$) be $\angle v_i$ and $\angle v_j$, respectively. For a given threshold $\delta_\angle$, if $|\angle v_i - \angle v_j| > \delta_\angle$, then the edge $\overline{v_i v_j}$ is said to be a turning edge, and vertex v_i and v_j are turning vertices; otherwise, it is a non-turning edge.

We observe from Fig. 9 that the angle difference between adjacent vertices after and before a turning has the maximum angle error about 10 degrees. As such, in our experiments we set $\delta_\angle = 15$ which can guarantee accurately detected turning action. Clearly, due to noisy gyroscope data, $\angle v_i$ and $\angle v_j$ can be different even though two peaks occur at a short interval in a straight corridor. To guarantee that the graph is not squiggly, for any non-turning edge, we define an adjusted angle of v_j as $\angle_{adj} v_j \triangleq \angle v_i$, and the adjusted angle at the initial time is zero, such that when the user is walking in a straight corridor without turning actions, we can yield a straight topology based on the adjusted angle sequence. The turning angle between turning vertex v_i and the vertex v_j after turning is computed as $|\angle v_i - \angle v_j|$. This way, we can successfully eliminate the accumulative error of gyroscope readings.

Note that the computed turning angle may vary among different pedestrian paths sharing the same pairwise turning vertices, and thus hardly feasible for generating a virtual graph. One naive way to derive a fixed turning angle $||\angle v_i - \angle v_j|$ is to conform the following rules: if $|\angle v_i - \angle v_j|$ nearest to $45°$ (or $90°, 135°, \cdots$), then we set $|\angle v_i - \angle v_j| = 45°$ (or $90°, 135°, \cdots$). The undesirability is that it might deviate from the truth that the turning angle can be other degrees. We will present our solutions of exploiting multiple pedestrian paths with overlapped segments in Section III-B4.

With the obtained adjusted angles, we can easily determine relative direction of each edge. To compute the absolute direction, which will be used in this paper for path merge (where we also leverage the vertex number between peaks of interest as the edge direction is not sufficient to detect overlapped segments, detailed in Section III-B4), user lock-on, and progress estimation, we further incorporate the compass readings for edge direction computation. Note that the compass readings are usually more unstable[4] than the gyroscope data. For instance, while walking in a straight corridor for tens of seconds, we observe that the compass readings can have a fluctuation over 15 degree, as shown in Fig. 10 (a), while the gyroscope readings are relatively stable (see Fig. 10 (a)). As such, the compass readings are not suitable for directly computing the relative direction of an edge. Since the gyroscope readings are stable, especially after alignment with the identified peaks described above, we can use these data for correcting the compass readings: when there is no left/right turn is detected based on the gyroscope data while the compass data vary greatly, the compass readings are not accurate. In PILOT, when walking in a straight corridor, we eliminate such inaccurate compass readings and regard the median of the sequence with the left compass data as the absolute direction of the edges on the straight corridor.



(a) (b)

Fig. 10. The compass and gyroscope readings collected when walking along a straight corridor. (a) The compass readings; (b) The gyroscope data.

3) Floor Level Change Detection: Note that the participant may go upstairs/downstairs by means of an elevator/escalator, or even stairs. In this case, we will observe that the barometer readings vary greatly. Existing indoor navigation systems like FollowMe [6] can only tell the level changing, but not the number of levels changed, which may confuse the users when taking an elevator.

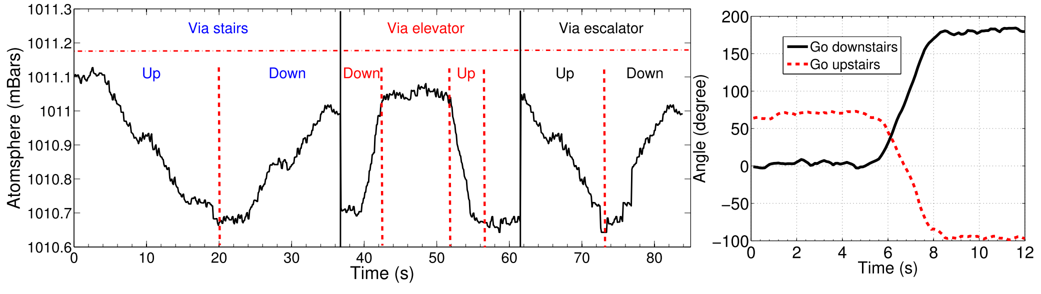
Our approach is motivated by the observation that when taking elevators, escalators or stairs, the atmosphere curves with time have different features (please see Fig. 11 (a)), where the curve by elevators is the steepest, and the curve by going stairs is the flattest. Although we can differentiate taking elevator with going stairs, it seems unreliable to tell the difference by taking escalator and going stairs. Fortunately, we find that by going stairs, the gyroscope readings have a different pattern. That is, there is a sharp turn of 180 degrees, as shown in Fig. 11 (b), which can be used for differentiating going stairs with taking escalators.

Once we detect the level change information, we can further determine the changed level when participants take an elevator and thus the specific level of each participant as follows. Assume that the indoor space of interest has k floors. Let the absolute difference of barometer data of one level change is δ_1, and that of a participant by taking elevators is δ_2. By computing the ceiling of $\delta(l) = \frac{\delta_2}{\delta_1}$ we derive the changed level. Afterwards, for a sub-graph of level $l_i (i \leq k)$ formed by the vertices on the same floor, we can obtain the difference, denoted by $\delta(i,j)$, of l_i and any other floor level, say $l_j (j \leq k)$. If for any $j (j \neq i)$, the vertices (or one of them) on the sub-graph of level $l_j (j \leq k)$ has a higher level than those on the sub-graph of level l_i, then we infer that $l_i = 1$. As a result, we can infer the specific level of other sub-graphs.

Till now, a pedestrian path for participant i is generated, which consists of vertices $v_1^i, v_2^i, \cdots, v_{k_i}^i (k_i > 1)$ and weighted edges $e_1^i, e_2^i, \cdots, e_{l_i}^i$ with directions. Note that we can easily infer the direction of the reversed Edge $e_j^i (j < l_i)$.

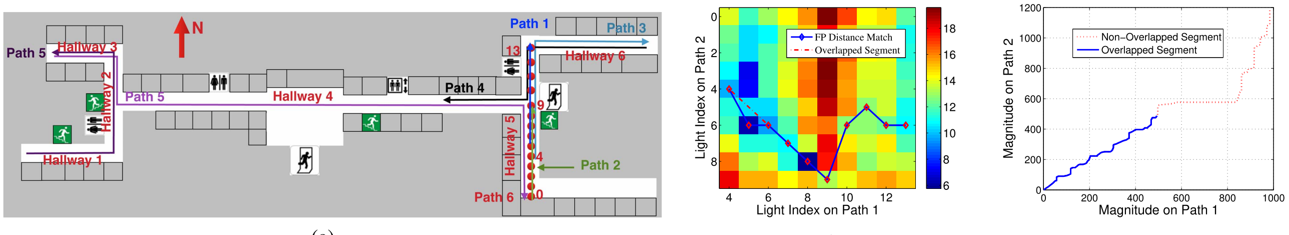
4) Path Merge: When multiple paths are uploaded to the server, PILOT merges them to form the virtual graph, such that the system can provide the user a shorter navigation path, or a shortcut when the user has multiple destinations to visit. To that end, we need to find out whether two paths have an overlapped segment and where they meet. In literature, Travi-Navi [7] exploits WiFi-based fingerprint (FP) distance to compute overlapped segment, yet it requires the device to

[4] Sometimes it can fluctuate even over 90 degrees, e.g., due to the presence of working appliances.



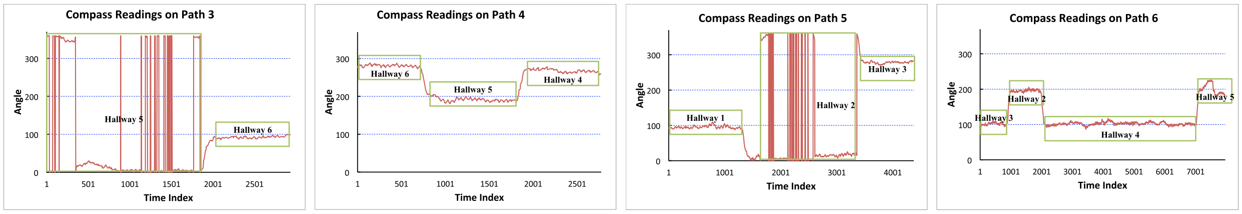
(a) (b)

Fig. 11. Illustrative examples. (a) The barometer reading curves via stairs/elevator/escalator; (b) The walking direction curves of going upstairs/downstairs.



(a) (b) (c)

Fig. 12. An illustrative example on an indoor scenario with 6 hallways. (a) Different pedestrian paths are marked by different colors. (b) WiFi-based overlap detection. (c) DTW-based overlap detection.



(a) (b) (c) (d)

Fig. 13. The compass readings collected when walking along four different paths marked in Fig. 12(a).

frequently scan the surrounding WiFi signals, which is known to be energy-expensive; also the accuracy is not guaranteed.

As an illustrative example, please see Fig. 12 where Path 1 and Path 2 in Fig. 12 (a) are generated when two users walk along the corridor and collect the surrounding WiFi signals under the luminaries starting from Light 0 to Light 10, and Light 4 to Light 13, respectively. For each vertex v_i on Path 1, we compute the FP distance of v_i to the vertices on Path 2, and if the FP distance between v_i on Path 1 to the vertex v_j on Path 2 is the minimal, we infer that Path 1 and Path 2 overlap at v_i (or v_j). Fig. 12 (b) shows the matching results where the numbers of the x-axis and y-axis labels represent the indices of luminaries on Path 1 and Path 2, respectively, and the color bar indicates the FP distance value between pairwise vertices on Path 1 and Path 2 where the blue indicates a small distance and the red indicates a large distance. Clearly, Vertex 5 on Path 1 is mistakenly identified as Vertex 6 on Path 1.

On the other hand, we observe that compass readings are usually discriminative from hallway to hallway (e.g., Hallway 4 and Hallway 6, Hallway 2 and Hallway 5, have significantly different compass readings) and fine-grained as composed to WiFi signals, as shown in Fig. 13, and thus apply the dynamic time warping (DTW) technique [33] on the sequence of compass data for overlapped segment detection. Fig. 12(c) shows that the DTW-based approach derives the correct matching result and detects the accurate overlapped segment. Note that the compass readings alone may not be sufficient for accurate overlapped segment detection, e.g., for two identical hallways with comparable compass readings. As such, we further exploit the number of peaks/vertices on the hallway which reflects the luminary density, as different hallways are likely to have different density of luminaries [23].

Specifically, to narrow down the scope of DTW computation, we define a *corridor branch* as the collection of edges between two adjacent turning vertices. For two paths, namely Path 1 and Path 2, if for any corridor branch of Path 1, there is no corridor branch in Path 2 such that their medians of the compass readings are almost the same, we determine that they have no overlapped segments. Otherwise, they may share common edges and thus we apply DTW technique for further verification of overlapped segments by matching the compass data sequences of Path 1 and Path 2. If the DTW cost is minimized among all corridor branches with similar compass readings and the same number of vertices, then these two paths have shared edges, which can be located based on the warped path. With these strategies, we can merge two paths into a bigger graph based on the detected overlapped segments. Eventually, all paths can be organically integrated into a whole virtual graph in the order of path uploading. Clearly, by further incorporating some fixes such as stairways, elevators, or WiFi routers, etc., can greatly help to anchor the path and thus make our scheme more robust with less computation complexity.

Note that after two paths having overlapped segment(s) are merged, even for the same edge, the length on the two

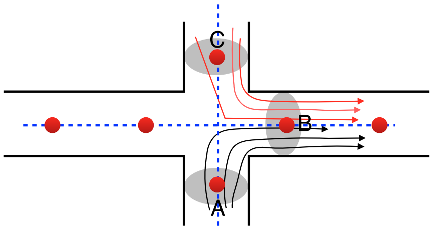


Fig. 14. The turning angle.

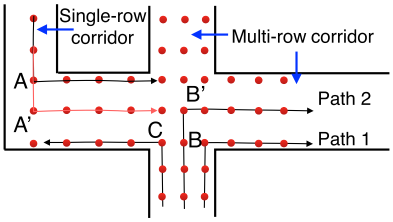


Fig. 15. The multi-row luminaries.

paths may be different as they are uploaded by different users walking at different speeds, and the turning angle for each pedestrian path may also vary as the participants have different trajectories, as mentioned in Section III-B2. Similar with [10], we now address these issues by incorporating pedestrian paths with overlapped segments. Let the average length of edges on the overlapped segments of Path 1 and Path 2 be t_1 and t_2, respectively. We define the length ratio of the shared edges be $\lambda(p_1, p_2) = \frac{t_2}{t_1}$, then the length of any edge on Path 2 will be updated as $\lambda(p_1, p_2)t(v_i, v_j)$ where $t(v_i, v_j)$ denotes the walking time between v_i and v_j on Path 2. Similarly, we sequentially align the edge length on other paths with overlapped segments with Path 1 or Path 2, such that the edge length by different participants are comparable.

To compute a unique turning angle, e.g., between vertices A and B in Fig. 14, we compute the average of the turning angles of n_{AB} paths crossing A and B, and n_{CB} paths crossing B and C. This is because vertex A and vertex B locate at a straight corridor, such that $\frac{|\angle A - \angle B| + \pi - |\angle C - \angle A|}{2} = \frac{\pi}{2}$. As such, we estimate the turning angle $|\angle A - \angle B| = (\frac{\sum_{i=1}^{n_{AB}} |\angle^i A - \angle^i B|}{n_{AB}} + \pi - \frac{\sum_{i=1}^{n_{CB}} |\angle^i C - \angle^i B|}{n_{CB}})/2$.

In some corridors (e.g., in a shopping mall), there can be multi-row luminaries. We call such a corridor as a *multi-row corridor*. Based on the multiple directed pedestrian paths with overlapped segments, we can easily identify the multi-row corridor. For instance, in Fig. 15, one participant makes a left turn at Vertex A while another participant turns left at Vertex A' with the same turning angle. We can determine that Vertex A and A' belong to different rows. Similarly, after identifying that Path 1 and Path 2 locate at different rows of a multi-row corridor before turning right in above way, we also infer that Path 1 and Path 2 belong to different rows of the luminaries after turning right. This is because that, for Path 1, the participant turns at Vertex B, and another participant make a right turn at Vertex B' with the same turning angle of 90 degrees, but the number of detected peaks before turning right on Path 1 is less than that of Path 2.

Note that during pedestrian path generation, when people walk in the corridor near the windows or under a transparent roof, the light from bulbs might be immersed by the sunlight, such that PILOT may derive false-negative results which poses great challenges for path merge. In addition, there might be some luminaries out of function, or extra luminaries might be deployed. To deal with these dynamic issues, we exploit the relative and absolute direction of edges. Specifically, in Fig. 16, the overlapped segment of Path 1 (generated without the presence of sunlight) and Path 2 (suffered from sunlight effect) in North direction is first identified. As there are a left turn with 90 degrees, we can determine that they have an overlapped segment in the effecting area of sunlight, and the DTW technique is applied for computing the overlapped segment. In a similar way, we can infer that Path 3 and Path 4 have an overlapped segment when there are two luminaries out of function. As such, we randomly select one path with the largest number of vertices to initialize the graph, and then a path having an overlapped segment with a path in the graph will be merged to form an organically grown graph.

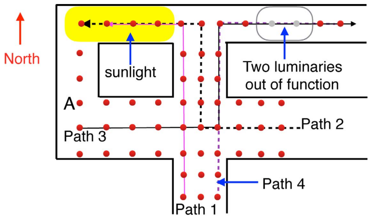
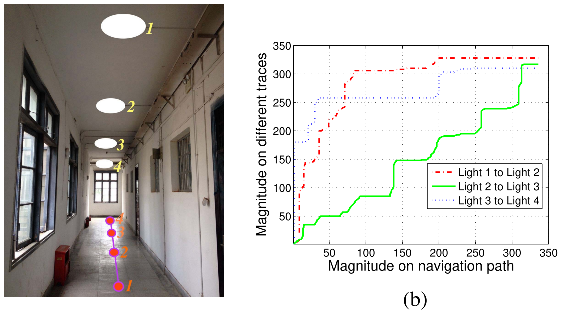


Fig. 16. The sunlight and luminary dynamic.

5) Semantic Labeling: As mentioned in Section I, if the destination of a user is his/her friend, as the friend is also affiliated with a vertex of the virtual graph and the navigation path can be easily computed on the graph with the given starting vertex (corresponding to the luminary under which the user stands) and destination (corresponding to the luminary under which the friend stands), there is no need for manually labeling. If a user wants to visit a place, to anchor the constructed virtual graph to the real world, we need to assign the corresponding vertex a semantic meaning (e.g., the shop name or room number) to facilitate navigation for users with arbitrarily starting locations. To that end, each shop owner (or other involved users) labels the semantics of the starting and destination positions after finishing the crowdsensing task, such that two end vertices are semantically labeled. Note that not every vertex needs to be labeled. Such a labeling process can also be used for verifying overlapped segment detection since, if two paths overlap, there will be some vertices having the same labels on the overlapped segment.

C. Real-Time Navigation

To offer navigation services, PILOT first computes the orientation, e.g., by the methods in [37] and [38] and affiliates the user with the virtual graph based on sensed data from her/his mobile device as follows. When the user starts the navigation service, PILOT needs to locate the starting vertex affiliating with the user based on the compass readings. Specifically, the user first walks along the corridor for a few steps, and the mobile device continuously detects the light intensity, gyroscope and compass readings. After some peak intensity being detected, a short pedestrian path with the computed gyroscope and compass data will be uploaded to the server. The server determines which corridor branch in the virtual graph has overlapped segments with the newly generated path, in the similar way discussed in Section III-B4, such that the starting vertex affiliating with the user is determined. Please see Fig. 17 as an illustration. With the sequence of compass readings, truncated by the peaks, of the navigation user, PILOT applies



(a)

(b)

Fig. 17. The DTW-based user lock-on. The previous location of a user is under light 2, and the current location is under light 3. (a) Luminaries layout; (b) The DTW matching results.

(a) (b) (c)

Fig. 18. The experiment scenarios of PILOT. (a) A supermarket; (b) a giant shopping mall; (c) an office building.

the DTW technique on the compass readings sequence of the new path and those of history traces by participants with similar edge directions. It can be shown that the starting vertex corresponds to the Light 2. In practice, the navigation users may be asked to walk longer (e.g., walk from Light 2 to Light 4, rather than from Light 2 to Light 3) for collecting more compass readings, depending on how discriminative the compass readings are from peak to peak. After inferring the starting vertex, PILOT computes a navigation path from current location (e.g., the vertex corresponding to the Light 3) to the destination, i.e., the vertex of the target label (e.g., a shop name) as an input by the user.

During the navigation progress, the user walks along the derived navigation path that consists of vertices $\{v_1, v_2, \cdots, v_n\}$ and directed edges $\{e_1, e_2, \cdots, e_{n-1}\}$. The mobile device keeps sensing data, such that PILOT can track the user continuously to avoid deviation from the course. Specifically, if the user has just passed by Vertex $v_i(i = 1, 2, \cdots, n-1)$ on the navigation path, and then the light sensor detects a new peak, PILOT updates v_{i+1} as the current vertex, and prompts when to change the walking pattern (i.e., turning or floor-level change) if there is a turning edge ahead on the navigation path. With the gyroscope and barometer data, PILOT can easily detect whether the user deviates from the course or not, and offers an alert of going back to the path if detecting a deviation. Such a tracking process continues until the user reaches the destination, i.e., $v_{i+1} = v_n$.

Note that during the navigation process, some luminaries might be switched-off, or there can be sunlight overwhelming the lights from luminaries, such that false negative results (i.e., some true luminaries/peaks are detected) may occur. To tackle this challenge, PILOT computes the ratio of the walking time $t(v_i, v)$ of the navigation user (i.e., from the last vertex v_i to the new peak v), to the edge length of e_i, denoted by $l(e_i)$ which is measured by the average walking time of participants. If $\frac{t(v_i,v)}{l(e_i)} = \frac{t(v_{i-1},v_i)}{l(e_{i-1})}$ where $t(v_{i-1}, v_i)$ denotes the walking time of the user from v_{i-1} to v_i, PILOT updates $v = v_{i+1}$ as the current vertex. When multiple (say k) luminaries are off, PILOT judges whether $\frac{t(v_{i-1},v_i)}{l(e_{i-1})} = \frac{t(v_i,v)}{l(e_i)+l(e_{i+1})+\cdots+l(e_{i+k-1})}(i + k \leq n)$ holds, and updates $v = v_{i+k}$ as the current vertex if the equality holds. Note that the users may stand still for a while such that the walking time gets longer. In this case, we can use accelerometer readings to infer whether a user is walking or not, and derive the walking time by subtracting the elapsed time of standing still. An alternative of PILOT is to apply DTW technique to compute sequence similarity of the compass readings between v_i and v_{i+k} $(i \leq n - k)$ with the sequence between v_i and the new peak. If the largest similarity is with the sequence from v_i to $v_{i+j}(j \leq k)$, then v_{i+j} is updated as the current vertex v. As such, the navigation progress can be accurately tracked such that the prompt instructions is still guaranteed

At the same time, the user can also judge the direction of next step based on the updated vertex and the navigation map on the mobile device. This double-check strategy greatly reduces the possibility of the user's veering off the course, which is also confirmed by our experiments.

IV. System Evaluation

In this section, we will present the evaluation of the key functional components of PILOT. We then evaluate PILOT in typical indoor environments (please see Fig. 18) for a better understanding of PILOT effectiveness.

A. Implementation

We implement PILOT system on the Android platform. The scanning frequency for light intensity, gyroscope and compass measurement is set to be 50 Hz. The graphic user interface (GUI) displays the navigation instructions. During the walking progress, PILOT tracks user and updates the current vertex in the virtual graph and shows the prompt instructions on when and where to change the walking pattern (e.g., left/right turn with the sign of $\leftarrow$ / $\rightarrow$, go upstairs/downstairs with the sign of $\uparrow$ / $\downarrow$ via elevator/escalator/stairs).

We conducted the experiments in three typical scenarios: a one-storied supermarket with the testing area of 1000 m^2 where the luminaries are about 7 meters high with the separation of 3 meters between adjacent luminaries, a six-storied shopping mall with the testing area of 20000 m^2 where the luminaries are about 4 meters high with the separation of 2 meters between adjacent luminaries, and a four-storied office building with the testing area of 800 m^2 where the luminaries are about 3.5 meters high with the separation of 3.6 meters between adjacent luminaries. We have four participants, carrying a variety of mobile devices (Huawei Mate 8, Huawei P9, Samsung Galaxy S5, and Google Nexus 9) at hand, walk around the indoor spaces covering the main hallways to collect the sensory data and generate pedestrian paths for constructing high-coverage virtual graph.

B. Performance Evaluation

1) Accuracy of Peak Detection: Table I presents peak detection accuracy in the investigated three scenarios. Note that

TABLE I
ACCURACY OF PEAK DETECTION

Scenario	Light Number	TP	FP	FN	Accuracy
Supermarket	125	125	0	0	100%
Shopping mall	2448	2406	12	32	98.3%
Office	80	80	0	0	100%

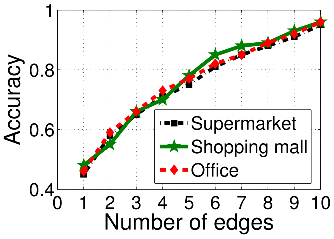


Fig. 19. Overlap detection accuracy.

some peaks correspond to the true lights which we call as *truly positive* (TP), and some are caused by the interference or at turning vertices which we call as *false positive* (FP). Besides, as some lights separate very close, some of the detected peaks are deemed as noises, and we call them as *false negative* (FN). We observe that PILOT can accurately detect peak intensities in the office building and supermarket, since there the luminaries are separated farther or the lights hang from the ceilings with an enough height such that the interference from nearby light sources can be eliminated, and the detection accuracy of PILOT in the shopping mall is 98.3%, which clearly demonstrates the effectiveness of our proposed peak detection method.

2) Overlapped Segment Detection Accuracy: Fig. 19 describes the accuracy of overlapped segment detection vs. different segment lengths in terms of the number of common edges between pairwise pedestrian paths. We find that the detection rate is very low when the paths shared a few edges, but it increases significantly with the increase of the overlapped segment length. When there are 9 or more shared edges, the detection accuracy reaches above 90%. We believe that the major reasons are that the luminary layout usually varies from corridor to corridor, such that the number of vertices between branches tends to vary among all corridors with nearly the same compass readings with which the user is walking along, and also the compass reading sequences are adopted to measure path similarities for further verification.

3) The Time Saving of PILOT: To show the advantage of PILOT in saving navigation time, we manually deploy some landmarks at turning vertices to guide a user's next movement, and compute different between the navigation time by PILOT and by the landmark-based navigation system. Fig. 20 shows the cumulative distribution function of the saving time. We find that under the help of the virtual graph, we have the global information of navigation path which greatly saves the walking time to the destination since the users do not necessarily need to stop and see whether they will reach the destination. Among the investigated three scenarios, the saving time in Shopping mall is the most since the indoor structure is the most complex and thus the users have difficulties in finding their destinations by the landmark-based navigation system.

4) The Delay of User Lock-on: Fig. 21 plots the CDF of user lock-on delay when the users walk straight along the current corridor without making any turns. One can imagine that if a user makes a turn during the lock-on process, s/he can be more quickly located by PILOT. When the time for uploading the sensory data is not considered, which greatly depends on the network condition instead of our system itself, around 50% of users are located to the nearest vertex within 5 seconds since that they have to walk for a while to generate a pedestrian path for lock-on, and around 15% of users can be locked within 3 seconds, showing that PILOT is promising for providing a real-time navigation.

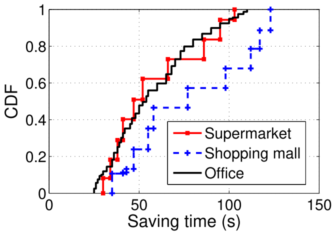


Fig. 20. The saving time.

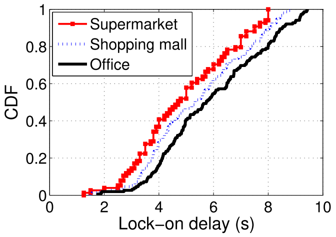


Fig. 21. The lock-on delay.

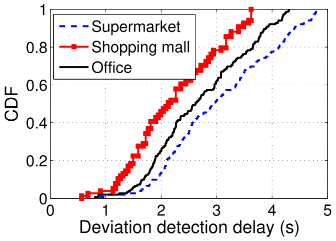


Fig. 22. The deviation detection delay.

5) The Delay of Deviation Detection: An important function of a navigation system is to timely provide for users the prompt deviation alert (if any). PILOT keeps track of a navigation user and sends an alert when the user deviates from the computed navigation path, namely, misses the turning/level change, makes a wrong turning/level change. As the navigation path consists of directed edges between vertices/peaks, based on the compass readings uploaded by a navigation user, PILOT can easily detect such deviations based on DTW matching results and alerts the user to go back to the course. Fig. 22 shows the delay for deviation detection during navigation progress. Desirably, we observe that deviation can be detected within 5 seconds for Supermarket, Shopping mall and Office.

6) The Lead Time of Navigation Instructions: Fig. 23 examines the lead time of navigation instructions (i.e., the ahead-of-time of the instructions before changing walking direcions at the junctions of interest) in the investigated three scenarios. As the navigation progress is tracked by PILOT, when an instruction on change of walking direction or floor level is offered, PILOT records the timestamps at the moment and then record the timestamps after detecting a turning action or floor level change, hereon computes the lead time of instructions

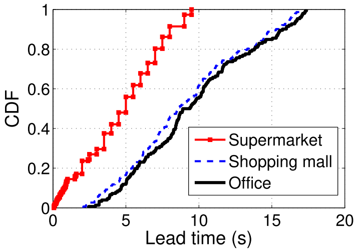


Fig. 23. The lead time of instructions.

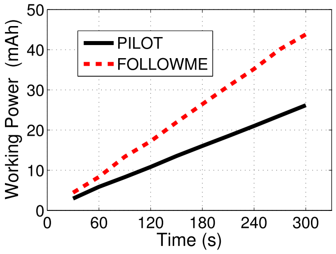


Fig. 24. Power consumption of path generation.

at given checkpoints (i.e., vertices of turning edges which are manually labeled on the virtual graph and navigation paths) by computing the difference of the two timestamps.

In Supermarket, the instructions can be given 10 seconds before reaching turning vertices, while in the Shopping mall and Office, the lead time can be much longer. As a matter of fact, with the help of the weighted and directed virtual graph, when PILOT detects a turning, it can quickly compute the walking time for next turning or to reach the destination.

7) The Power Consumption: To show the power efficiency of PILOT, we let a participant serve as a leader to generate a pedestrian path, and a navigation user (a.k.a, the follower) uses this path as a reference path for navigation, just like the guider-follower based navigation systems [6], [7]. We repeat such an experiment 20 times, and compare PILOT with FOLLOWME [6] since, unlike other navigation systems (e.g., iMoon [11], Travi-Navi [7]) which use such power-hungry sensors as WiFi and Camera, only energy efficient sensors are adopted in PILOT and FOLLOWME. Also it has been shown in [6] that FOLLOWME is more energy-efficient than Travi-Navi. For fair comparison, similar with [6], we assume that the navigation path has the same starting vertex and destination with the pedestrian/reference path. In other words, the starting vertex is assumed to be known. We use the built-in software, named HwSystemManager, of Huawei Mate 8, designed for measuring battery usage information of APPs installed in the smartphone, to compute the power consumption of PILOT and FOLLOWME. During experiments, all other APPs and extra hardware components (e.g., WiFi, GPS, etc.) are turned off.

The power consumption measurements of PILOT and FOLLOWME against the duration time for path generation and navigation are shown in Fig. 24 and Fig. 25, respectively. Fig. 24 depicts the comparison study of FOLLOWME and PILOT on power consumption during pedestrian path generation, and we find that the power curves of FOLLOWME and PILOT are both linear with the duration time, while the slope of the curve by FOLLOWME is larger. Also we observe that PILOT has an energy saving of nearly 50%. Fig. 25 plots the power consumption during navigation. As in this case PILOT only use light sensors, barometer and gyroscope during navigation, and no complicated calculation is involved, while FOLLOWME applies the DTW approach on the geomagnetic measurements, we find that PILOT is much more energy-efficient, especially when the navigation time becomes longer since the computation cost for DTW is larger.

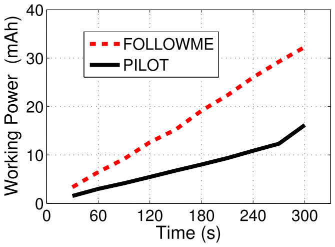


Fig. 25. Power consumption of navigation.

V. Related Work

Indoor localization and navigation have received a lot of attention in recent years, given the demand of LBS and the prevalence of smartphones with rich built-in sensors. We can roughly classify the literature of indoor navigation into two categories: floor map based and guider-follower based.

A. Floor Map Based Indoor Navigation

Many indoor navigation systems are based on given floor maps, and applies localization techniques to guide the users. Recently, a handful of studies on indoor localization have been proposed. Among them, WiFi fingerprinting based methodology is perhaps the most popular one, e.g., [39]–[45], which relies on the availability of such infrastructures as WiFi APs or GSM towers and requires the time-consuming signal strength calibration/site survey process to build radio-frequency (RF) map. Recently, the Inertial Measurement Unit (IMU) sensors on smartphones have been exploited to achieve infrastructure free indoor localization [9], [13], [46], which consists of such components as step counting, stride strength estimation, and a particle filter. Yet the pure IMU sensing-based approach would suffer from cumulative errors, and regular absolute position fixes based on landmarks [38] or beacons [26], are often needed to ensure long-term operation and to cope with unexpected behaviors. The geomagnetic field has been adopted for indoor localization due to its global availability and stability [12], [14], [15] while the accuracy is relative lower.

Visible light has been further incorporated into indoor localization systems. In [16]–[21], taking the advantage of LED lights being capable of rapid on-off keying, the authors propose to modulate the LED lights and modify COTS device (or mount a special device on the COTS device) for achieving high accuracy. However, the requirements on modulating the frequency of LED lights and modification on COTS mobile devices make it prohibitive in reality. Recently, the conventional incandescent/fluorescent lamps have been used for localization [19], [28], [47], without modulation on lights or modification to the device. However, they require that floor maps and/or luminaries placement are given in advance while only achieving room-level localization accuracy.

Zhu and Zhang [24] present iLamp, an indoor localization system based on the observation that each lamp has hidden visual features imperceptible to human eyes but can be extracted by processing the lamp image. Clearly, iLamp requires the power-hungry camera and also raises the concern of privacy leakage. In NaviLight [23], a vector of multiple light intensity from any unmodified light sources are exploited as location fingerprints, yet many factors affect the absolute value of light intensity, including room temperature, starting time, distance (and relative angle) of device to light sources, device diversities, etc. Recently, Shao *et al.* [25] propose RETRO, a retroreflector-based visible light localization system which can achieve centimeter-level accuracy. In RETRO, a light-weight small device, named retroreflector, is mounted on each passive IoT device such that the light can be reflected back to its source, and an LCD shutter is also mounted on the IoT device to ensure a unique signature. In addition, multiple photodiodes (PDs) are also mounted on lamps to establish a low-delay backward channel from the retroreflector to the lamps.

B. Guider-Follower Based Indoor Navigation

Recently, several navigation systems have been developed for indoor environments, especially without the aid of floor maps [6], [7], [26], [29]. Escort [26] guides users to the vicinity of their friends by using crowd encounter information. Yet it relies on pre-deployed audio beacons to correct dead-reckoning drifts due to the noise in the accelerometer and compass sensors, which may limit its practical deployment. Riehle *et al.* [29] developed a magnetic navigation system to navigate blind people. The authors used customized wireless IMUs for collecting magnetic information. Travi-Navi [7], a vision-guided navigation system, enables a user to easily bootstrap and implement indoor navigation services without localization or floor maps. Travi-Navi uses a guider-follower model: during a leader's trip, Travi-Navi records pathway images and samples WiFi fingerprints and IMU sensors, which are packed into a reference trace. A user then downloads the trace data and is guided from the same starting location to the same destination location, by comparing the user's current sensor readings with the reference trace. FollowMe [6] also leverages the leader-user model to address the last-mile navigation problem in indoor and semi-outdoor environments; a reference trace is generated only based on a leader's record of sensory data along a specific trip and her/his walking patterns.

VI. Conclusion

Numerous indoor navigation systems have been developed in recent years, given the prevalence of smart mobile devices which can conveniently collect ambient information through built-in sensors. Yet these systems requires either expensive floor maps or complex indoor localization systems as a prerequisite. In this paper, we exploited the ubiquitous visible light sources in typical indoor environments, and presented PILOT, a novel navigation system which successfully overcomes the above limitations. We implemented a prototype of PILOT on the Android platform, and conducted extensive experiments in typical indoor environments to valid the effectiveness and efficiency of PILOT.

References

[1] I. A. Junglas and R. T. Watson, "Location-based services," *Commun. ACM*, vol. 51, no. 3, pp. 65–69, 2008.

[2] A. Jovicic, J. Li, and T. Richardson, "Visible light communication: Opportunities, challenges and the path to market," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 26–32, Dec. 2013.

[3] X. Chen, X. Gong, L. Yang, and J. Zhang, "Amazon in the white space: Social recommendation aided distributed spectrum access," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 536–549, Feb. 2017.

[4] Z. Yu, H. Xu, Z. Yang, and B. Guo, "Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 151–158, Feb. 2016.

[5] T. Shu, Y. Chen, and J. Yang, "Protecting multi-lateral localization privacy in pervasive environments," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1688–1701, Oct. 2015.

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

[7] Y. Zheng *et al.*, "Travi-Navi: Self-deployable indoor navigation system," in *Proc. ACM MOBICOM*, 2014, pp. 2655–2669.

[8] R. Gao *et al.*, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. ACM MOBICOM*, 2014, pp. 249–260.

[9] 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.

[10] 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.

[11] J. Dong, Y. Xiao, M. Noreikis, Z. Ou, and A. Ylä-Jääski, "iMoon: Using smartphones for image-based indoor navigation," in *Proc. ACM SenSys*, 2015, pp. 85–97.

[12] J. Chung *et al.*, "Indoor location sensing using geo-magnetism," in *Proc. ACM MOBISYS*, 2011, pp. 141–154.

[13] F. Li *et al.*, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM UBICOMP*, 2012, pp. 421–430.

[14] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *Proc. IPIN*, 2012, pp. 1–9.

[15] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lv, "MaLoc: A practical magnetic fingerprinting approach to indoor localization using smartphones," in *Proc. ACM UBICOMP*, 2014, pp. 243–253.

[16] P. Hu, L. Li, C. Peng, G. Shen, and F. Zhao, "Pharos: Enable physical analytics through visible light based indoor localization," in *Proc. ACM HOTNETS*, 2013, Art. no. 5.

[17] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *Proc. USENIX NSDI*, 2014, pp. 331–343.

[18] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proc. ACM MOBICOM*, 2014, pp. 447–458.

[19] Q. Xu, R. Zheng, and S. Hranilovic, "IDyLL: Indoor localization using inertial and light sensors on smartphones," in *Proc. ACM UBICOMP*, 2015, pp. 307–318.

[20] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Wearables can afford: Light-weight indoor positioning with visible light," in *Proc. ACM MOBISYS*, 2015, pp. 317–330.

[21] B. Xie, G. Tan, and T. He, "SpinLight: A high accuracy and robust light positioning system for indoor applications," in *Proc. ACM SENSYS*, 2015, pp. 211–223.

[22] C. Zhang and X. Zhang, "LiTell: Indoor localization using unmodified light fixtures: Demo," in *Proc. ACM MOBICOM*, 2016, pp. 481–482.

[23] Z. Zhao *et al.*, "NaviLight: Indoor localization and navigation under arbitrary lights," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[24] S. Zhu and X. Zhang, "Enabling high-precision visible light localization in today's buildings," in *Proc. ACM MOBISYS*, 2017, pp. 96–108.

[25] S. Shao, A. Khreishah, and I. Khalil, "RETRO: Retroreflector based visible light indoor localization for real-time tracking of IoT devices," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1025–1033.

[26] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see bob?: Human localization using mobile phones," in *Proc. ACM MOBICOM*, 2010, pp. 149–160.

[27] M. Alencastre-Miranda, L. Munoz-Gomez, R. Murrieta-Cid, and R. Monroy, "Local reference frames vs. global reference frame for mobile robot localization and path planning," in *Proc. MICAI*, 2006, pp. 309–318.

[28] A. R. Jiménez, F. Zampella, and F. Seco, "Light-matching: A new signal of opportunity for pedestrian indoor navigation," in *Proc. IPIN*, 2013, pp. 1–10.

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

[30] J.-G. Park, D. Curtis, S. Teller, and J. Ledlie, "Implications of device diversity for organic localization," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 3182–3190.

[31] Y. Shu *et al.*, "Gradient-based fingerprinting for indoor localization and tracking," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2424–2433, Apr. 2016.

[32] *Global Illumination*. Accessed: Nov. 4, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Global_illumination

[33] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.

[34] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou, "Human sensing using visible light communication," in *Proc. ACM MOBICOM*, 2015, pp. 331–344.

[35] I. Constandache, R. Choudhury, and I. Rhee, "Towards mobile phone localization without war-driving," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[36] *How to Measure Stride or Step Length for Your Pedometer*. Accessed: Nov. 4, 2018. [Online]. Available: https://www.walkingwithattitude.com/articles/features/how-to-measure-stride-or-step-length-for-your-pedometer

[37] N. Roy, H. Wang, and R. R. Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Proc. ACM MOBISYS*, 2014, pp. 329–342.

[38] H. Wang *et al.*, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM MOBISYS*, 2012, pp. 197–210.

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

[40] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of RSS fingerprinting based indoor localization," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 2479–2487.

[41] Q. Xu, A. Gerber, Z. M. Mao, and J. Pang, "AccuLoc: Practical localization of performance measurements in 3G networks," in *Proc. ACM MOBISYS*, 2011, pp. 183–196.

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

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

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

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

[46] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, 3rd Quart., 2013.

[47] N. Ravi and L. Iftode, "FiatLux: Fingerprinting rooms using light intensity," in *Proc. PERVASIVE*, 2007.

Wenping Liu (SM'17) received the Ph.D. degree from the Huazhong University of Science and Technology in 2012. He is currently an Associate Professor with the Hubei University of Economics, China. His research interests include mobile computing, Internet of Things, smart city, and sensor networks.

Hongbo Jiang (SM'14) received the Ph.D. degree from Case Western Reserve University in 2008. He ever was a Professor with the Huazhong University of Science and Technology. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University. His research concerns computer networking, especially algorithms and protocols for wireless and mobile networks. He is serving as an Editor for the IEEE/ACM Transactions on Networking, an Associate Editor for the IEEE Transactions on Mobile Computing, and an Associate Technical Editor for the *IEEE Communications Magazine*.

Guoyin Jiang (M'18) received the Ph.D. degree from the Huazhong University of Science and Technology, China, in 2010. He was a Professor with the School of Information Management and Statistics, Hubei University of Economics, China. He is currently a Professor with the University of Electronic Science and Technology of China. His research interests include decision support systems, simulation, smart city, and e-commerce.

Jiangchuan Liu (S'01–M'03–SM'08–F'17) received the B.Eng. degree *(cum laude)* from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He was an Assistant Professor with The Chinese University of Hong Kong, and a Research Fellow at Microsoft Research Asia. He is currently a University Professor with the School of Computing Science, Simon Fraser University, BC, Canada. He is an EMC-Endowed Visiting Chair Professor with Tsinghua University, and also an Adjunct Professor with the Tsinghua-Berkeley Shenzhen Institute. His research interests include multimedia systems and networks, cloud computing, social networking, online gaming, big data computing, RFID, and Internet of Things. He is an NSERC E.W.R. Steacie Memorial Fellow. He is a co-recipient of the inaugural Test of the Time Paper Award of the IEEE INFOCOM (2015), the ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), and the ACM Multimedia Best Paper Award (2012). He is a Steering Committee Member of the IEEE Transactions on Mobile Computing and the Steering Committee Chair of the IEEE/ACM IWQoS (2015–2017). He has served on the editorial boards of the IEEE/ACM Transactions on Networking, the IEEE Transactions on Big Data, the IEEE Transactions on Multimedia, the IEEE Communications Surveys and Tutorials, and the IEEE Internet of Things Journal.

Xiaoqiang Ma received the B.Eng. degree from the Huazhong University of Science and Technology, China, in 2010, and the M.Sc. and Ph.D. degrees from Simon Fraser University, Burnaby, BC, Canada, in 2012 and 2015, respectively. His areas of interest are wireless networks, social networks, and cloud computing.

Yufu Jia received the Ph.D. degree from the Huazhong University of Science and Technology, China. He is currently an Assistant Professor with the School of Information Management and Statistics, Hubei University of Economics, China. His research interests include mobile computing and Internet of Things.

Fu Xiao received the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, China, in 2007. He is currently a Professor and a Ph.D. supervisor with the School of Computer, Nanjing University of Posts and Telecommunications, China. His research papers have been published in many prestigious conferences and journals, such as IEEE INFOCOM, IEEE/ACM ToN, IEEE JSAC, IEEE TMC, ACM TECS, and IEEE TVT. His research interests are mainly in the areas of Internet of Things and mobile computing. He is a member of the IEEE Computer Society.