CloudNavi: Towards Ubiquitous Indoor Navigation Service with 3D Point Clouds

Xiaoqiang Teng, *Member, IEEE,* Deke Guo, *Member, IEEE,* Yulan Guo, *Member, IEEE,* Xiaolei Zhou, *Member, IEEE,* and Zhong Liu, *Member, IEEE*

Abstract—The rapid development of mobile computing has prompted indoor navigation to be one of the most attractive and promising applications. Conventional designs of indoor navigation systems depend on either infrastructures or indoor floor maps. This paper presents CloudNavi, a ubiquitous indoor navigation solution, which only relies on the point clouds acquired by 3D camera embedded in a mobile device. It pushes the design of indoor navigation to the extreme on five dimensions: accurate, easy-to-deploy, infrastructure-free, robust to environment and crowds, and universal. CloudNavi conducts a first but significant step towards realizing this vision by fully exploiting the advantages of point clouds. Particularly, CloudNavi first efficiently infers the walking trace of each user from captured point clouds. Many shared walking traces and associated point clouds are combined to generate the point cloud traces, which are then used to generate a 3D path-map. Accordingly, CloudNavi can accurately estimate the location of a user using a limited number of point clouds, and then guide the user to its destination from its current location. Extensive experiments are conducted on office building and shopping mall datasets. Experimental results indicate that CloudNavi exhibits outstanding navigation performance in both office building and shopping mall.

Index Terms—Indoor navigation, point cloud processing, mobile crowdsourcing, 3D path-map, indoor localization

1 Introduction

Indoor navigation is highly attractive for a user to obtain the most convenient and shortest walking path to the destination. For instance, the user could pose the query "How can I arrive at the meeting room from my current location in the conference building?" or "How can I find the walking path towards a store in a shopping mall?" Despite the strong demand, ubiquitous navigation service remains a great challenge for indoor environments, since it is expected to realize the following goals.

- **Accurate:** It should be able to accurately guide a user to its destination and track the walking progress.
- **Easy-to-deploy:** It should not require any prior indoor map or dedicated indoor localization system deployment.
- **Infrastructure-free:** It should be independent of any infrastructure, such as WiFi access point, bluetooth, and UWB. Actually, it is unnecessary for indoor environment to offer such infrastructures.
- **Robust:** It should be robust to environment variations and crowds, and can be able to detect deviation events to notify the users.
- **Universal:** It should be able to guide a user to any destination from its current location, rather than from a few predefined locations.

- *X. Teng, D. Guo, X. Zhou, and Z. Liu are with the College of Information System and Management, National University of Defense Technology, Changsha, Hunan,* 410073*, P. R. China.*
- *Y. Guo is with the College of Electronic Science and Engineering, National University of Defense Technology, Changsha, Hunan,* 410073*, P. R. China.*
- *E-mail: {tengxiaoqiang13, dekeguo, yulan.guo, xl.zhou, liuzhong} @nudt.edu.cn.*

Manuscript received April, 2016.

Conventional indoor navigation methods fall into the following two categories. The first category significantly relies on indoor localization infrastructures. Thus, the navigation scenarios are restricted to WiFi based [1], [2], [3], [4] and Visible Light Communication (VLC) based [5], [6]) environment. The second category highly depends on the available indoor maps, e.g., the dead reckoning based [7], [8] and the images based [9], [10] systems. It is clear that such indoor navigation methods are neither easy-to-deploy nor infrastructure-free.

Recent works demonstrate that indoor navigation can be easily bootstrapped and deployed for motivated users without comprehensive indoor localization systems or even floor maps [11], [12], [13]. Motivated by the prospective of self-deployable indoor navigation, we propose a ubiquitous approach, CloudNavi. It pushes the design of indoor navigation to the extreme on the five dimensions for wide deployment. This extreme vision, if realized, can give significant benefits for ubiquitous indoor navigation.

Although the Travi-Navi approach makes a first step towards self-deployable indoor navigation [11], it does not tackle the last three design goals. It jointly utilizes multiple sensors (e.g., camera, inertial sensor, magnetometer) to guide a user to its destination. However, its application has been restricted by several limitations. First, it still requires particular infrastructures (e.g., WiFi access point) for practical use. Second, images are not robust to environmental variations and crowds as they are sensitive to scale, rotation, illumination, and moving crowds. Finally, users can be guided from only a limited number of predefined locations, such as the building entrances. FollowMe [12] shares similar idea with Travi-Navi, while replacing images with geomagnetic data. Furthermore, iMoon [13] uses the Structure-from-Motion (SfM) results of large-scale images to

guide users. SfM, however, is also not reliable and feasible in dynamic and cluttered indoor environments insufficient features. The camera location inferred by SfM is not accurate and hence decreases the navigation accuracy [14] .

Signals (e.g., WiFi, geomagnetic data, and images) used by existing navigation systems are inherently unstable and may vary significantly [11], [12], [13]. However, the indoor structural information (e.g., windows, doors, logos, and decorations) is invariant and robust to crowds within a long duration. This motivates us to design ubiquitous navigation systems using the structural information of indoor space.

To achieve this goal, this paper presents a point cloud based indoor navigation system, named CloudNavi. In particular, a pathway navigation system is designed since people usually move along pathways [8]. Methods from the computer vision community are used to solve this problem. **First**, point cloud can provide rich structural information of an indoor space. In addition, features extracted from point clouds are commonly not affected by variations in scale, rotation and illumination [15]. Therefore, it is more robust to environment variations and is more stable than other signals (e.g., RF, magnetic data, and images). **Second**, during the tracking process, using point clouds only are sufficient for navigation. In contrast, most existing works rely on data from multiple types of sensors in a mobile device. **Third**, with the rapid development of low-cost 3D perception mobile devices (e.g., Google project tango mobile device[1]) and techniques (e.g., [16], [17]), point cloud can be easily obtained and is becoming a type of ubiquitous resources.

Despite the above benefits, CloudNavi faces several major challenges. **First**, since CloudNavi takes crowdsourced point cloud traces as its input and each point cloud trace can be acquired under different pose of a mobile device (e.g., holding in hand or with swinging arms), varying number of features might be included in these point clouds. It is critical for a mobile device to accurately model the point cloud trace, including the acquisition of high-quality point cloud and the accurate estimation of walking trace. To collect high-quality point cloud, the pose of a mobile device is estimated to trigger the acquisition process under desired conditions. To accurately estimate the walking trace of a user, the Visual Inertial Odometry (VIO) method is used to address the hand pose variation problem.

Second, it is unknown how to accurately construct a 3D path-map using point clouds only. To tackle this problem, a set of turning points are detected to partition a trace into several segments. Point clouds are then used to measure the similarity between two walking traces. Next, large-scale walking traces are systematically integrated to a 3D path-map. To improve the accuracy of a 3D path-map, false match detection is considered as an optimization problem and a heuristic solution is proposed. In addition, a marking task is designed to semantically label the 3D path-map.

Third, given the resultant 3D path-map, it remains a great challenge to locate a user and immediately guide the user along an optimal path without the information of initial localization. In this paper, a novel localization method is proposed to accurately locate a user by calculating the similarity of input point cloud against the 3D path-map. Besides, CloudNavi can be used to automatically plan the navigation path, track the walking progress of a user, and detect the deviation of walking progress.

1. http://www.google.com/atap/project-tango/

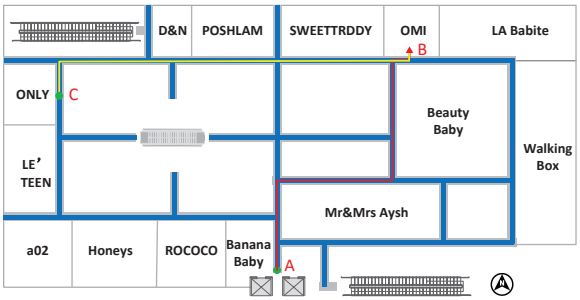


Fig. 1: An indoor navigation example.

A CloudNavi prototype system is implemented and comprehensive experiments have been conducted in two typical buildings under various conditions (e.g., during daytime and night). Experimental results demonstrate that CloudNavi exhibits outstanding navigation performance in both office building and shopping mall. The major contributions of this paper are summarized as follows:

- Five design goals are defined for a ubiquitous indoor navigation system, including accurate, easy-to-deploy, infrastructure-free, robust, and universal. CloudNavi is proposed to satisfy these goals by exploiting the indoor structural information only.
- A novel method is proposed to construct a 3D path-map using point clouds by measuring the similarity between two walking traces. Different from radio map, 3D path-map is robust to environment variations and crowds.
- A novel localization method is designed by calculating the similarity between point clouds. Accordingly, CloudNavi can automatically plan and track the navigation path.
- A prototype of CloudNavi is implemented. Its feasibility and performance is tested in two typical indoor environments. Extensive experimental results demonstrate that CloudNavi makes a great progress towards ubiquitous and widely deployed indoor navigation service.

The rest of this paper is organized as follows. Section 2 presents an overview. Section 3 provides the preliminary techniques used in CloudNavi. Section 4 describes the details of CloudNavi. Section 5 reports the evaluation results, followed by technical discussions in Section 6. The related work is summarized in Section 7 and this paper is concluded in Section 8.

2 Overview

2.1 An Indoor Navigation Example

A navigation example of our CloudNavi system is presented in Fig. 1. A self-motivated user (e.g., an OMI store owner) collects a point cloud trace (the red line in Fig. 1) using a mobile device with CloudNavi system. As the user holds the mobile device in an upright position and walks to the OMI store (i.e., point B in Fig. 1) from an entrance (i.e., point A in

Fig. 1), pathway point clouds are acquired and a point cloud trace is automatically generated by the CloudNavi System. If there are many self-motivated users, multiple point cloud traces will be collected. A point cloud map is then generated by CloudNavi using these point cloud traces (the blue lines in Fig. 1).

If a user wants to reach the OMI store (i.e., point B in Fig. 1), CloudNavi system first locates him in the 3D path-map (i.e., point C in Fig. 1) and then generates an optimal navigation trace plan (i.e., the golden line in Fig. 1). As the user naturally holds its mobile device in hand and moves forward, CloudNavi system tracks its walking progress in the 3D path-map by matching instant point clouds. If the user is off the correct path, the deviation event will be detected and a notification will then be given. In that case, a new optimal trace will be calculated based on the current location of the user.

2.2 System Architecture

Figure 2 illustrates the architecture of the proposed CloudNavi system. It consists of three components: mobile client, server, and navigation user.

Mobile Client. Crowdsourced sensor data from collectors are recorded in indoor space. Specifically, the walking trace of a data collectors is obtained using the VIO algorithm. Meanwhile, the point clouds are associated with walking traces to generate point cloud traces. More importantly, the pose of a mobile device is estimated to acquire high-quality point clouds and reduce computational cost. The generated point cloud traces are then uploaded to the server for further processing.

Server. This part includes the 3D path-map construction module and the navigation module. In the 3D path-map construction module, crowdsourced traces are merged to generate a 3D path-map. In order to track the walking progress of a user, its localization on the 3D path-map is estimated by the navigation module in real time during the whole navigation time. Based on the localization results, a notification is given to the user and the recommended navigation trace is updated if the user is off the correct trace. In addition, the optimal plan can automatically selected if multiple traces are available from the start location to the destination

Navigation Users. The only task for a user is to give CloudNavi its destination and to start the point cloud acquisition process.

3 Preliminaries and Measurements

In this section, the major techniques behind our system are presented, primary measurements are also conducted to understand the variation of point clouds.

3.1 Point Cloud

A point cloud is a set of vertices defined in a three-dimensional coordinate system with X, Y, and Z values. Point clouds are usually generated by a 3D scanner, such as a Google tango tablet. A large number of points on the surface of an object can be acquired by a 3D scanner. Recently, the open-sourced Point Cloud Library (PCL) [18] has been released to include a number of existing point cloud processing algorithms. In our work, the XYZ data type of a point cloud is used.

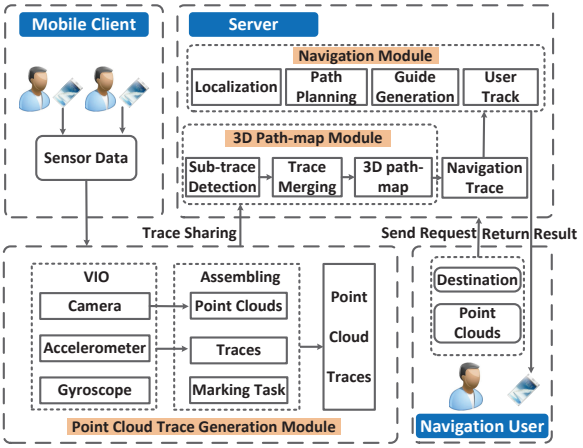


Fig. 2: System architecture.

3.2 Point Cloud Properties

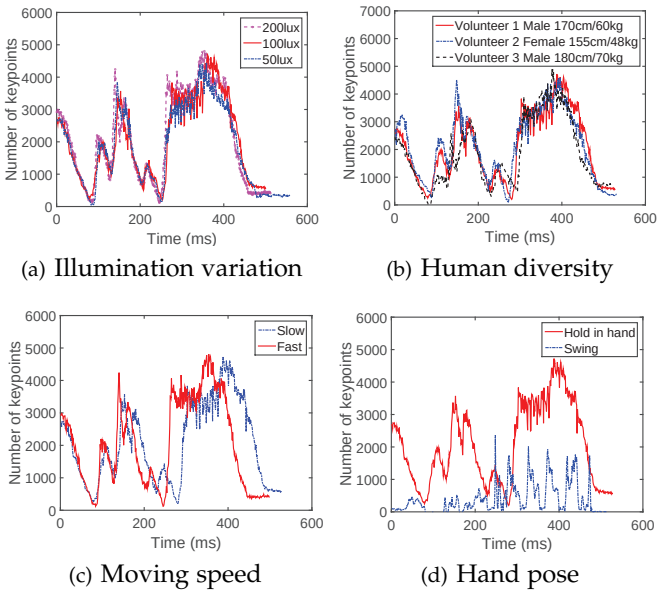
In this subsection, a systematic study is conducted to investigate the effects of various factors (temporal changes, illumination variation, color variation, human diversity, moving speed, and hand pose) on the properties of point clouds. 10 volunteers were invited to conduct experiments, these volunteers are different in genders, heights and weights. 7889 point clouds along 288 walking traces were collected in our office building across nine months using the Google project tango tablet and the Xtion PRO Live camera[2]. These traces covered most areas of the experimental field and were different in length. In order to calculate the difference between two point clouds, the number of keypoints is used. The Scale-Invariant Feature Transform (SIFT) algorithm is used to detect the keypoints for a point cloud [19].

Temporal Changes. The layout of a building may be changed because of the refurbishing of buildings and rooms over time, such as room facades. In order to test the influence of temporal changes, the Wanda shopping mall was selected, which is located in China. It has four storeys and more than 250 stores. The floorplan of Wanda mall contains rich information including store names, store location, promotion information, widths of the corridors, and store facades. The number and proportion of changed rooms on different dates are counted (i.e., three months, six months, and nine months). Table 1 shows that the layout of the indoor space remains unchanged and only the locations of a small number of indoor general objects (e.g., chairs) are changed. Besides, an updated method is proposed to reduce the effect of temporal changes in Sec. 4.3.4.

2. http://www.asus.com

TABLE 1: The number and proportion of changed rooms in a large shopping mall with more than 250 stores.

	3 months	6 months	9 months
Number	8	14	20
Proportion	0.040	0.056	0.080



(a) Illumination variation (b) Human diversity

(c) Moving speed (d) Hand pose

Fig. 3: The number of keypoint detected in point clouds acquired under different situations.

Illumination Variation. We tested the influence of illumination on point clouds. The point clouds were collected under three different lighting conditions (i.e., 200lux, 100lux, and 50lux) along the same path by the same volunteer. Figure 3(a) shows that detected keypoints are similar for point clouds acquired under different lighting conditions. There are several reasons for this result. First, the XYZ data type of a point cloud is used in our work. Point clouds are not influenced by illumination variations. Second, the depth data is recorded by a 3D camera (e.g., Google project tango tablet) using structured light technique, which is roust to illumination variations.

Human Diversity and Moving Speed. Furthermore, we tested the effect of human diversity on point clouds. Volunteers with different genders, heights, and weights were asked to collect point clouds along the same walking trace. The keypoints of each point cloud were generated using the SIFT algorithm. Figure 3(b) shows that the keypoint detection results are highly similar for different people. For the same volunteer, the detected keypoints vary with respect to different moving speed of the volunteer (as shown in Fig. 3(c)). Therefore, point clouds are robust to human diversity and moving speed , which provides opportunities for navigation.

Hand Pose. We tested the effect of hand pose for point cloud collection. The point clouds were collected by two holding poses along the same path for the same volunteer, i.e., the device was held in a static hand and a swinging hand, respectively. Figure 3(d) shows that swinging pose can affect the number of keypoints detected in a point cloud. This hand pose variation problem will be further analyzed in Sec. 4.1.

3.3 Iterative Closest Point

An Iterative Closest Point (ICP) algorithm is used to match two point clouds. It first select some sample points from one or both point clouds, and matches these points to sample points in the other set. The alignment error is minimized by an iterative process. Given two sets of corresponding points P and Q from two point clouds with partial overlap, the least square method is used to calculate the translation vector T and the rotation matrix R. The two point clouds are registered by minimizing the error function:

$$E(R,T)=\frac{1}{n}\sum_{i=1}^{n}\|q_i-(Rp_i+T)\|^2, \quad (1)$$

where $q_i \in Q$, $p_i \in P$, n is the number of corresponding points between two point clouds. The performance of the ICP algorithm is affected by the overlap between a pair of point clouds. Here, the overlap is defined as the common part between two point clouds. A higher overlap leads to a more accurate registration. In CloudNavi, the ICP algorithm is mainly used for 3D path-map construction (Sec. 4.2) and localization (Sec. 4.3).

Note that, the ICP algorithm is significantly affected by the overlap rate between two point clouds. A higher overlap rate results in more accurate registration. However, if the overlap rate is too high, more point clouds are needed to construct the 3D path-map. In this paper, the overlap rate is set to be at least 60% [20]. The overlap rate is calculated using the translation and rotation values of a mobile device. The translation value of a mobile device represents the walking distance of a user along a straight line, and the rotation value represents the rotation angle. First, since the points within a depth range (i.e., $0.5\sim4$m for a Google project tango tablet) can be recorded by a mobile device, we select point clouds using a distance interval $\Delta(d)$, where $\Delta(d)$ is set to one step distance (typically 0.75m). $\Delta(d)$ is sufficient to accurately calculate the walking trace using the VIO algorithm [21], [22]. Generally, the pose of the mobile device is stable within one step [11]. Second, since the points within the field of view (i.e., 57.5×45 for a Google project tango tablet) can be recorded by a mobile device, we set the rotation value (Φ) based on two considerations. If the user is static, we set Φ=20° according to the field of view. If the user walks one step, we set Φ=10°.

4 System Design

In this section, we first describe the point cloud trace modeling and the 3D path-map construction methods, and then introduce the methods for user localization, walking progress tracking, and deviation event notification.

4.1 Point Cloud Trace Modeling

Key Point Cloud Acquisition. One bottleneck for CloudNavi is point cloud processing, especially for the ICP algorithm. Therefore, we performed a quantitative analysis using 1275 point clouds acquired in office and mall. Several interesting observations are listed as follows.

1) Point clouds in two consecutive frames are highly similar, as shown in Figs. 4 (a-b).
2) The number of keypoints detected from a plain background (e.g., floorboard, ceilings) is smaller than a textured background (e.g., logos, decorations) using the SIFT algorithm. For instance, 30, 26, 738, 1361, 643, and 16 keypoints are detected in the point clouds

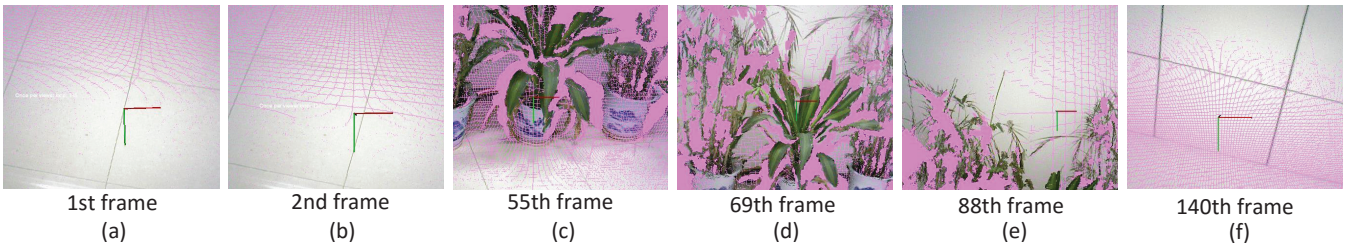


Fig. 4: Point clouds acquired with different poses of a mobile device.

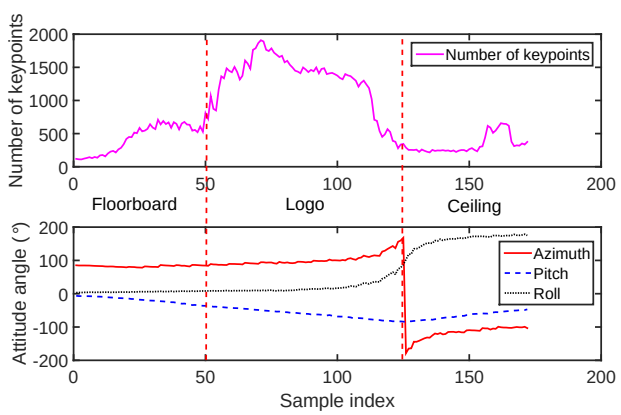


Fig. 5: The number of keypoints detected in a point cloud acquired under different poses of a mobile device.

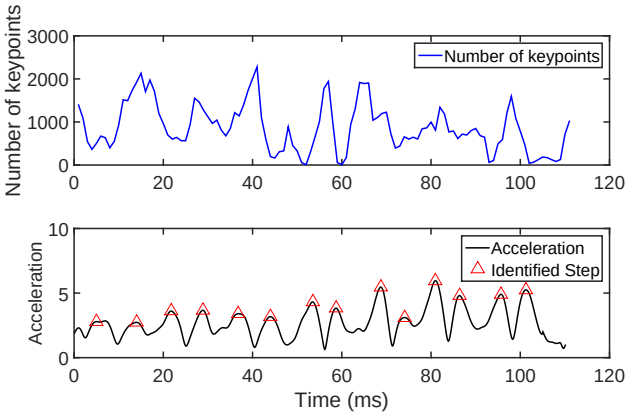


Fig. 6: The number of keypoints detected in point clouds acquired with different hand poses.

acquired from different backgrounds, as shown in Figs. 4(a-f), respectively. Moreover, the number of keypoints detected from a point cloud is related to the pose of a mobile device, as shown in Fig. 5. For example, when a mobile device is directed towards a textured background (e.g., a logo), the number of keypoints is significantly larger than the point clouds acquired from floorboard or ceiling.

Motivated by these observations, an efficient point cloud acquisition method is proposed using the Logistic Regression (LR) algorithm [23].

First, the point cloud acquisition event f is defined as a binary classification problem, i.e.,

$$f_i = \begin{cases} 1 & \text{start acquisition} \\ 0 & \text{no acquisition,} \end{cases} \tag{2}$$

where f_i defines acquisition event at ith time. Then, a predict model is defined using three pose angles (i.e., a, p, r), the walking distance, and the data from light and proximity sensors. For t and t+1, $\Delta(d)$ denotes the walking distance, $\Delta(d)$=$d(t+1)$-$d(t)$. Let $\phi(t)$=$[a(t), p(t), r(t)]$, we have Φ=$||\phi(t+1)$-$\phi(t)||$. The parameter settings for $\Delta(d)$ and Φ have been discussed in Sec. 3.3. The data from the light and proximity sensors (s_l, s_p) are used to determine whether the mobile device is in a bag or pocket. Let X denote the set of variables, i.e., X=$[\Phi, s_l, s_p, \Delta(d)]$, Θ denotes the parameter set. The LR problem is defined as

$$f = \Theta^T X. \tag{3}$$

The cost function $J(\Theta)$ is

$$J(\Theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log(f_\Theta(x^{(i)})) + (1-y^{(i)}) log(1-f_\Theta(x^{(i)}))]. \tag{4}$$

The gradient descent algorithm is used to derive the optimal parameters:

$$\theta = \theta - \alpha \bigtriangledown_\theta J, \tag{5}$$

where α is the length for gradient descent, and $\bigtriangledown_\theta J^T = [\frac{\partial}{\partial \theta_0} J, \ldots, \frac{\partial}{\partial \theta_n} J]$.

In addition, if no point cloud is acquired within $\Delta(d)$, an acquisition event is repeated until a point cloud is acquired. Thus, point cloud acquisition can be automatically started.

Hand Pose. The variation of hand poses significantly affects the quality of captured point clouds. Since the 3D camera can be blocked if the mobile device is hold with a swinging hand, the quality of point clouds acquired in this case is very low. To solve this problem, an effective approach is proposed to capture high-quality point clouds. It is observed that the quality of point clouds is highly related to hand swinging. Figure 6 plots the number of keypoints detected in each frame against the differential of accelerometer magnitude. We found that high-quality point clouds can be acquired when the 3D camera is placed at the highest point during swinging.

Usage of Key Point Clouds. In this paper, the key point clouds are used for three purposes.

(1) Key point clouds are used to estimate walking traces of a user based on the VIO algorithm [21], [22]. Although dead reckoning method has been widely applied in mobile computing, it has several limitations. First, it cannot accurately estimate the pose of a mobile device [21], [22]. Second,

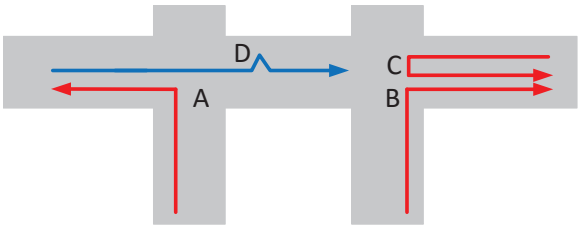


Fig. 7: The turning points detected using the VIO algorithm.

its accuracy is lower than the VIO algorithm for walking trace estimation, even for a short distance [24]. Thus, the VIO algorithm is used in this paper to estimate walking traces. Point cloud traces are generated by combining walking traces with key point clouds.

(2) Key point clouds are used to measure the similarity between two traces, as described in Sec. 4.2.1.

4.2 3D Path-map Construction

In this section, the crowdsourced point cloud traces are used to generate a 3D path-map by merging similar traces. A marking task is then designed to semantically label the 3D path-map based on the current location of the data collector.

4.2.1 3D Path-map Generation

Once a large number of point cloud traces are collected using the method proposed in Sec. 4.1, the next step is to calculate the similarity between two point cloud traces. These similar point cloud traces are merged based on the ICP algorithm.

Trace partitioning. When crowdsourced traces are uploaded to the server, the turning points are used to partition the traces into segments. In practice, turning point detection is very challenging when the user walks in indoor space. For example, a data collector may look at a poster on the wall in a straight corridor and turn around. Such false turning points should be detected and removed. Therefore, the key challenge is to accurately detect the turning points. Generally, the turning pattern of a user is divided into left turn (i.e., point A in Fig. 7), right turn (i.e., point B in Fig. 7) and U-turn (i.e., point C in Fig. 7). A correct turning is that the direction of walking of a user is changed, while a false turning (i.e., point D in Fig. 7) is that the the direction of walking of a user is not changed. Therefore, false turning points are accurately detected by checking the pose of a mobile device produced using the VIO algorithm [24].

Similar Segment Detection. Given two segments under their local coordinate systems, γ^1 and γ^2, where γ=$\{x_i, y_i, pc_i\}$, the trace is firstly divided into several segments using turning points, i.e., γ^1=$\{s_1^1, s_2^1, \ldots, s_n^1\}$, γ^2=$\{s_1^2, s_2^2, \ldots, s_m^2\}$, where n and m are the numbers of turning points for γ^1 and γ^2, respectively. The similarity between a pair of segments is then calculated using the following method.

Step 1: The SIFT algorithm is used to generate SIFT descriptors for a point cloud (I). Let D_p represent the number of SIFT descriptors, corresponding points between a pair of point clouds are then estimated using these SIFT descriptors [14]. Let C_p represent the number of corresponding points, the similarity between a pair of point clouds is calculated as:

$$S(I_1, I_2)=\frac{|C_p|}{|D_p^{I_1} U D_p^{I_2}|}. \tag{6}$$

Step 2: To calculate the similarity between two segments, the longest common subsequence (LCS) method is used. Let s_a and s_b be two segments with their lengths of m and n, respectively. The LCS metric is defined as:

$$L(s_{a,m}, s_{b,n})=\begin{cases} 0, \text{if } m{=}0 \text{ or } n{=}0; \\ 1{+}L(s_{a,m-1}, s_{b,n-1}), \\ \quad \text{if } S(\overrightarrow{s}_{a,m}, \overrightarrow{s}_{b,n}) \le \epsilon \text{ and } |m\text{-}n| < \delta; \\ \max(L(s_{a,m}, s_{b,n\text{-}1}), L(s_{a,m\text{-}1}, s_{b,n})), \\ \quad \text{otherwise.} \end{cases} \tag{7}$$

where δ is the length threshold for two point cloud traces and ϵ is the similarity threshold. The similar score S_s is defined as:

$$S_s=\max_{f \in F} \frac{L(s_a, f(s_b))}{\min(m, n)}, \tag{8}$$

where F represents a set of sliding windows. If S_s is larger than the threshold S_H, the two point cloud traces are considered as similar.

Trace Merging. Before trace merging, false matches should be detected. Given n segments and a set of l_i similar segments for segment i, the task of the False Match Detection Problem (FMDP) is to find the set of similar segments H to minimize the number of false matching. Let S denote the set of the similar segments, H=$\{h_{ij} | h_i, h_j \in S, 1 \le i, j \le n, i \ne j\}$. When the matching of segments i and j is false, a segment k is matched to the segment i or j, as $P_{i,j,k}$. Therefore, the objective is to find the set H to minimize the number of false matching:

$$\min_H \sum_{i=1}^{n} \sum_{j=1}^{n} P_{i,j,h_{ij}} \tag{9}$$

$$s.t. \quad i \ne j. \tag{10}$$

FMDP is NP-complete [9]. Here, a heuristic algorithm is proposed to solve this problem. First, let m_{ij} denote the similarity between segments i and j in H. If segments i and j are selected, m_{ij}=0; otherwise, m_{ij}=c (c is a constant). We then take M=$\sum_{i \ne j} m_{ij}$. Intuitively, the best H will produce the smallest M for all possible H. The details are described as follows:

1) Initialization: The segments with $|l_i|$=1 are merged and the smallest M is calculated.
2) Iteration: When a new trace containing similar segments in S is given, in order to merge these new segments, some merged segments may be split based on relative position constraints in traces. We compute M in each changing state and select these similar segments with the smallest M. Furthermore, these new segments are removed from S.
3) Termination: S=$\varnothing$.

Note that, the trace merging process can be implemented in parallel using a graph-subgraph isomorphism algorithm (i.e., the $VF2$ algorithm [25]). In addition, since the point cloud traces are represented in their local coordinate systems, the Bursa Wolf model [26] is used to transform these point

cloud traces into an indoor coordinate system. Finally, these similar point clouds from different traces are merged using the ICP algorithm. (x_i, y_i, pc_i) represents the coordinate of i-th point cloud in the merged point cloud traces.

4.2.2 Marking Task Design

In order to obtain the semantics along a walking path, the data collectors are asked to semantically label its current location, such as store names in a shopping mall and room numbers in an office building. These semantics can be used as navigation destinations in CloudNavi. Similar to Jigsaw [27], we assume that several incentive mechanisms [28] will be further developed for our practical system.

4.3 3D Path-map Based Navigation

To guide a user to its destination, it is important to locate the user on the 3D path-map. First, the current location of the user is considered as the start point of the navigation trace. Second, the location is used to track the user along the navigation trace. If there are multiple traces, a path planning algorithm is used to select the optimal trace. In addition, if the user is off the navigation trace, the user will be notified and the trace will be updated based on its current location.

4.3.1 Localization

Given a captured point cloud, the location is estimated using point clouds. Since each point cloud is acquired under the coordinate system of camera, the distance between the camera and each point of the point cloud is known [22]. Therefore, a triangulation method is proposed to calculate the relative location and pose of a camera from the point cloud. Furthermore, feature matching based method is used to register the point cloud with the 3D path-map to locate the mobile device.

The time for localization is increased with the number of point clouds in the 3D path-map used for matching. For start location detection, the point cloud is matched against the global 3D path-map, its time cost is relatively high. However, the time cost is acceptable in the initialization process (which is similar to GPS initialization). Then, the previous location can be used to define a local 3D path-map for point cloud matching, which significantly reduces the time consumption.

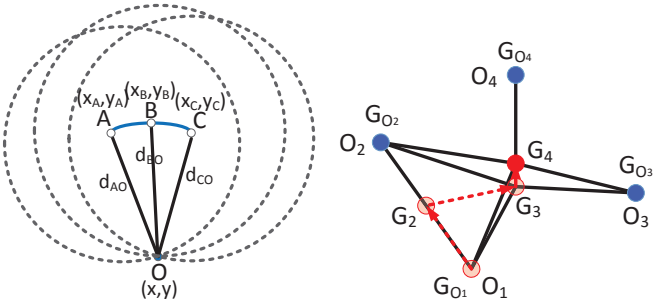
Triangulation Method. Three points are randomly selected from the point clouds (e.g., A, B, and C), as shown in Fig. 8(a). We have

$$O{=}T^{-1}B, \tag{11}$$

where $O{=}[x, y]$, $T{=}\begin{bmatrix} 2(x_A{-}x_C) & 2(y_A{-}y_C) \\ 2(x_B{-}x_C) & 2(y_B{-}y_C) \end{bmatrix}$, and $B{=}\begin{bmatrix} x_A^2{-}x_C^2{+}y_A^2{+}d_{CO}^2{-}d_{AO}^2 \\ x_B^2{-}x_C^2{+}y_B^2{-}y_C^2{+}d_{CO}^2{-}d_{BO}^2 \end{bmatrix}$.

Given three beacons and their coordinates, $C_A(x_A, y_A)$, $C_B(x_B, y_B)$, and $C_C(x_C, y_C)$ (see Fig. 8(a)), the distance between each beacon and the location of a user is calculated as $[d_{AO}, d_{BO}, d_{CO}]{=}[||C_A{-}O||, ||C_B{-}O||, ||C_C{-}O||]$, where $O{=}(0, 0)$.

Due to the complexity of indoor environment, point clouds acquired by different users at the same place may cover different scenes. Besides, two distinct locations may have similar point clouds (e.g., white walls). It is likely



(a) Triangulation method (b) Location estimation

Fig. 8: An illustration for user location estimation. (a) The location (O) of a user is estimated by one point cloud using the triangulation method. (b) The location (G_4) of a user is estimated by multiple point clouds using the center of gravity. O_1, O_2, O_3, and O_4 represent four point clouds. G_{O_1}, G_{O_2}, G_{O_3}, and G_{O_2} represent the estimated location of each point cloud. G_2, G_3, and G_4 represent the estimated location using the center of gravity, respectively.

that insufficient distinguishable features are acquired for localization in these cases. In practice, if the first point cloud acquired by the camera has problem for accurate identification, CloudNavi will continuous to acquire a new point cloud. However, the computation complexity is increased by using more point clouds. The challenge is how to choose appreciate point clouds. To solve this problem, a Point Cloud Localization Algorithm (PCLA) is proposed in this paper.

Point Cloud Localization Algorithm. Given a set of point clouds in a location $N{=}\{n_i|1 \leq i \leq n\}$ (n is the number of acquired point clouds at a location), the objective of PCLA is to minimize the total number of matching errors. We denote the decision variables as $B{=}\{b_k|1 \leq k \leq n\}$. The labels of the chosen point clouds for a location is denoted by $P_{N,B}$ when a set of point clouds N is given. The rotation matrix between two point clouds can be calculated by the ICP algorithm (i.e., $R{=}\{r_{ij}|1 \leq i, j \leq \frac{n(n-1)}{2}\}$) [29]. The objective is to find the set P to maximize the matching performance of point clouds with the minimum number of point clouds, the set P is computed as follows

$$\begin{aligned} P{=}&\arg\max_{P}(P_{N,B}) \\ s.t.\ &n_i{=}r_{ij} \times n_j, n_i, n_j \in N, r_{ij} \in R, i \neq j, \end{aligned} \tag{12}$$

where $i, j{=}1, 2, \ldots, n$. An illustration of PCLA is shown in Fig. 8(b).

Step 1: One point cloud (O_1) is selected to calculate the location (G_1) of a user using the triangulation method.

Step 2: Another point cloud (e.g., O_2) is selected to determine another candidate location (G_{O_2}). We take the center of gravity (G_2) as the final location for point clouds O_1 and O_2. Furthermore, we define the distance between G_i and G_j as $d_{ij}{=}\sqrt{(x_i{-}x_j)^2{+}(y_i{-}y_j)^2}$.

Step 3: *Step* 2 is repeated until $d_{ij} \leq \epsilon$. Considering computation complexity, we empirically tune ϵ accordingly to the environment.

4.3.2 Path Planning and Tracking

The Dijkstra's planning algorithm is used to calculate an optimal trace [30]. If the destination is changed, another

navigation trace will be recommended. That is, CloudNavi tracks the walking progress in real time using the locations.

4.3.3 Deviation Handling

If the user is off the correct path, the deviation event will be detected and the user will be notified. A method is proposed in CloudNavi to handle two cases of deviation events, including point cloud mismatch and real deviation.

Given a point cloud trace, the coordinates of a point cloud (x_{pc}, y_{pc}), and the current location (x_2, y_2), it is considered as a mismatch if the distance between the captured point cloud and current location of the user is larger than the maximum working distance of the 3D camera (e.g., 4m for Google project tango tablet). Otherwise, it is considered as a correct match.

Point Cloud Mismatch. If the estimated location is over the distance of previous point cloud acquisition, it is considered as a mismatch. Then, new point cloud is acquired to calculate the location of the user again. Consequently, the location of the user can be corrected in real time.

Real Deviation. If the location of a user determined by CloudNavi does not match the navigation path, it is considered as a real deviation. A notification will then be given to the user and a new optimal trace will be calculated based on the current location of the user.

4.3.4 Robustness to Environmental Variations and Crowds

Environment Variations. To address environment dynamics (e.g., the change of location of indoor objects), an update method is proposed in this paper. First, if the new point clouds provide more details of an area already covered by the latest 3D path-maps, the new point clouds will be registered to the latest 3D path-maps based on feature matching. Second, if the new point clouds represent changes in indoor scenes, or cover an area that has not been included in the latest 3D path-maps, these point clouds cannot be directly registered to the latest 3D path-maps. In this case, CloudNavi system will first create 3D path-maps from the new point clouds and then merge them with the existing ones.

Crowds. The indoor space is sometimes full of crowds, especially in a shopping mall. It is clear that the captured point clouds can be affected by the crowds. Accordingly, to improve the robustness of our CloudNavi system, a lightweight method is proposed. First, crowds are detected from point clouds using the work presented in [22]. Next, the user will be notified by our CloudNavi system to capture the point clouds of space between the head level of crowds and the ceiling. Such method can effectively remove the impacts of crowds.

5 Implementation and Evaluation

In this section, we first describe the implementation details of our CloudNavi system, and introduce the evaluation methodology and setups. We then test the performance of each component of our CloudNavi system.

5.1 Implementation

The CloudNavi prototype consists of two parts: a mobile client running on an Android mobile device and a navigation pipeline working on a server. The mobile client interface can be installed in different Andorid mobile devices which support WiFi and 3D cameras. The navigation pipeline was implemented on a Lenovo computer, which supports WiFi and has 32GB RAM, an i7 CPU processor and a 12GB Titan GPU. For most of our experiments, WiFi networks are used for communication between the mobile devices and the server.

Mobile Client. The mobile client software was used to acquire point clouds with timestamps. Instant point clouds are then automatically compressed and transmitted to the server with WiFi network.

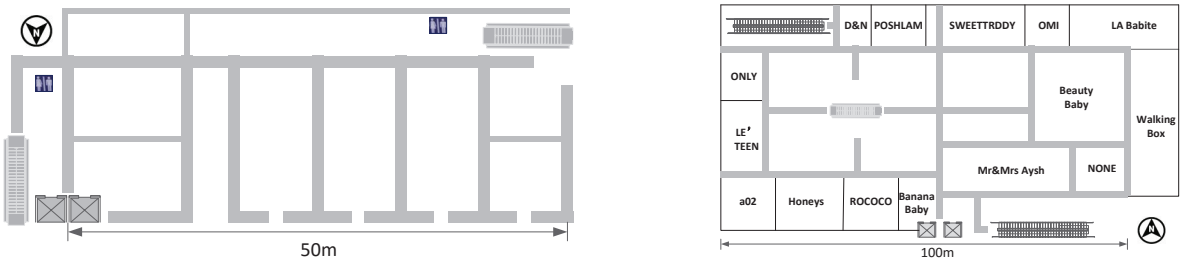
Server Configuration. The server was implemented in a computer with Ubuntu Linux system using two threads. The first thread is used to receive and store incoming point clouds, the second thread is used to process point clouds, create the 3D path-map, and generate navigation instructions. To reduce the running time for point cloud processing, a 12GB Titan GPU was used in our work.

5.2 Evaluation Methodology and Setups

We conducted experiments on one floor of a typical office building and a shopping mall. The office building covers $1000m^2$ with the length of $50m$ and width of $20m$. The shopping mall covers $4000m^2$ with the length of $100m$ and width of $40m$, as shown in Fig. 9. In experiments, the CloudNavi prototype is carried by a user while walking in office and mall buildings. The collected point clouds were transmitted to a server for post-processing using PCL (version $1.6.1$). Specifically, a pass through filter was first used to remove noise in point clouds[3]. The SIFT algorithm was then used to extract keypoints and the Fast Point Feature Histograms (FPFH) algorithm was used to obtain feature descriptors [15], [19]. Point cloud matching was achieved by comparing the corresponding points using the Random Sample Consensus (RANSAC) algorithm [15]. ICP algorithm was used to calculate the rotation and translation between two point clouds [29]. These results were directly sent to CloudNavi to generate instructions for the guidance of indoor walking.

15 volunteers were invited to participate in the data collection procedure. These volunteers consisted of 7 customers in a shopping mall and 8 members in our research group. The walking traces for point cloud acquisition were determined by volunteers themselves. The IMU data were automatically recorded by CloudNavi. For 3D path-map generation, 8556 point clouds along 268 walking traces were collected by these volunteers over a period of 3 days. These traces covered most areas of the experimental field. Each trace was connected to at least 2 rooms, and each room was covered by at least 5 traces. These traces are different in the areas they covered and in their lengths. The dataset was used to construct the 3D path-map. In order to evaluate the performance of navigation, volunteers were asked to collect 150 walking traces in the office building (dataset A) and 200 walking traces (dataset B) in the shopping mall.

3. http://pointclouds.org/



(a) Office building (b) Yuefang mall

Fig. 9: Experiment scenarios of our CloudNavi system.

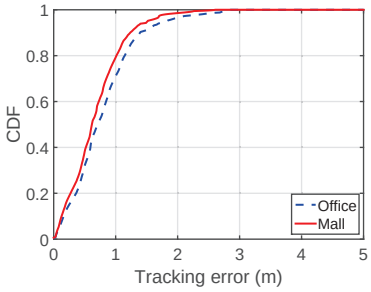


Fig. 10: The tracking errors in office and mall.

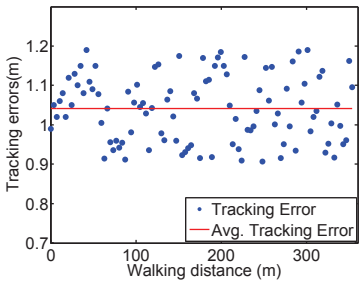


Fig. 11: Tracking errors with respect to different walking lengths along traces.

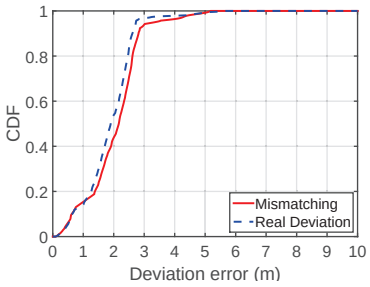


Fig. 12: Deviation errors for point cloud mismatching and real deviation events.

5.3 Performance Evaluation

5.3.1 Navigation Performance

Dataset A was used for the office building. In our experiments, all volunteers can successfully arrive at their destinations along their optimal traces, under the guidance of CloudNavi. Figure 10 shows that 90% tracking errors produced by CloudNavi are less than $1.39m$. As reported in [11], the typical tracking error achieved by Travi-Navi is less than 4 steps. In contrast, CloudNavi achieves a smaller tracking error with an error less than 2 steps (typically 0.75m for one step). That is because the point clouds used by CloudNavi are more robust to environment variations and crowds when compared to images. Moreover, the tracking error of CloudNavi is also less than FollowMe (95% of its spatial errors were smaller than 2m) [12] and iMoon (with a localization error of 2m) [13]. That is because the indoor structural information is invariant and robust to environmental dynamics within a long duration, while signals (e.g., WiFi, geomagnetic data, and images) are inherently unstable and may vary significantly in complex indoor environments.

Further, our CloudNavi system was tested in the Yuefang shopping mall using dataset B. In our experiments, all volunteers can successfully arrive at their destinations. Figure 10 shows that 90% tracking errors produced by CloudNavi are less than $1.20m$. The tracking error of CloudNavi is also smaller than Travi-Navi [11], FollowMe [12], and iMoon [13]. Moreover, the tracking error achieved in the shopping mall is smaller than that achieved in the office building. That is because the point clouds acquired in a shopping mall are rich in texture features.

We further tested the relationship between the tracking error and the walking distance on dataset B. The tracking errors under different walking distances were recorded. The walking distance is increased from 0 to 360m for each experiment. It can be found in Fig. 11 that the tracking error remains stable as the distance of walking traces increases. That is because the drift caused by tracking can be timely corrected by estimating the locations of the user on the 3D path-map.

5.3.2 Deviation Detection

We then tested the deviation detection performance of CloudNavi. If a deviation event is detected, the user is notified by CloudNavi and the type of deviation event is recorded. We recorded the location once a deviation event happened. Therefore, the distance between the start point and the location where the deviation event happened can be calculated. Figure 12 shows that the 90% deviation errors produced by CloudNavi are 2.8m for point cloud mismatch and 2.7m for real deviation, respectively.

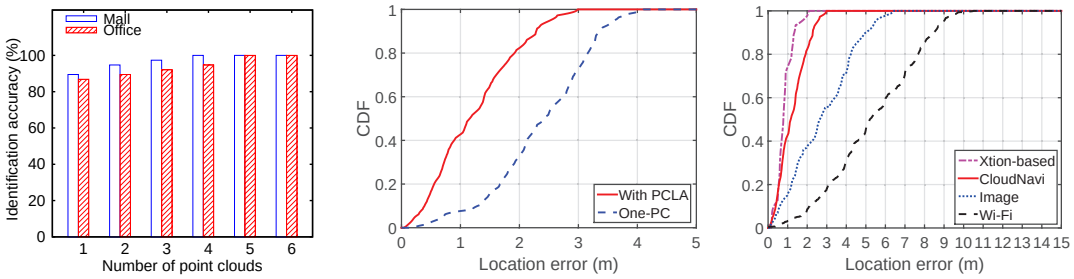
In summary, the tracking errors and the accuracy of deviation detection can satisfy the requirements of an indoor navigation system. Specifically, a user can easily find its destination with 90% tracking error of less than 1.5m (see Fig. 10), that is because the destination is within the range of the visibility.

5.3.3 Accuracy of 3D Path-map Construction

We further evaluated the accuracy of 3D path-map construction using the Root Mean Square Error (RMSE). Given n

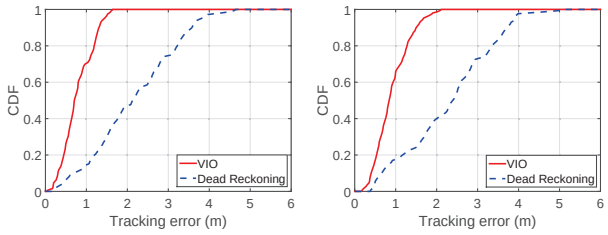
TABLE 2: RMSE of constructed 3D path-map (m).

	Office	Mall
1	0.567	0.489
2	0.520	0.505
3	0.561	0.491
4	0.560	0.499
Avg.	0.552	0.496



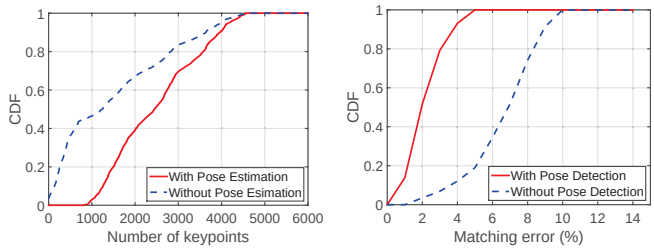
(a) Point cloud identification problem (b) Localization errors (c) Localization errors

Fig. 13: The evaluation of the point cloud identification problem, PCLA method, and the localization errors.



(a) Office building (b) Yuefang mall

Fig. 14: CDF of tracking error for the VIO and dead reckoning algorithms.



(a) Keypoints (b) Matching accuracy

Fig. 15: The evaluation of point clouds quality. (a) Number of keypoint with/without pose detection; (b) Accuracy of matching for point clouds with/without pose detection.

locations with $2D$ coordinates S_i^{PC}=(x_i^{PC}, y_i^{PC}), and their ground truth coordinates S_i^G=(x_i^G, y_i^G) (i=$1, 2, \ldots, n$). The RMSE is calculated as

$$e_{RMS}=\sqrt{\frac{\sum_{i=1}^{n}(S_i^{PC}\text{-}S_i^G)^2}{n}}. \quad (13)$$

The volunteers were asked to construct a 3D path-map in the office building and shopping mall. In total, 8556 point clouds along 268 walking traces were collected. The ground truth was recorded manually. Table 2 shows that the RMSEs of 3D path-map is small (e.g., less than $0.567m$) for both of the office and mall scenarios. It is clear that the accuracy of 3D path-map increases with the number of participated volunteers. This indicates that the proposed system can generate a highly accurate 3D path-map using crowdsourcing.

5.3.4 Localization Accuracy of Mobile Users

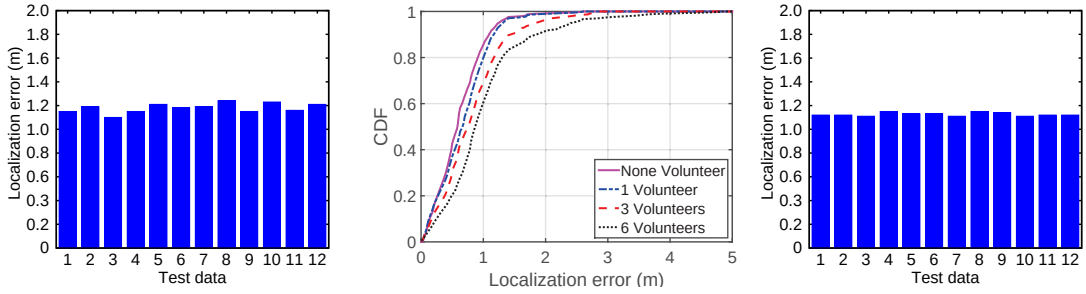
The localization accuracy is crucial for tracking and deviation detection. Due to the complexity of indoor environment, point clouds acquired by different users at a same place may cover different scenes. Besides, similar point clouds may be acquired from two different places. It is likely that sufficient distinguishable features are sometimes unavailable for localization. We first tested the accuracy of point cloud identification in both office and mall buildings. The number of point clouds for localization was increased from 1 to 6. Figure 13(a) shows the experimental results. We find that the accuracy of point cloud identification increases with the number of point clouds. When the number of point clouds is increased to 5, the highest accuracy of point cloud identification is achieved (with an accuracy of 100%).

We further tested the performance of PCLA. We randomly selected 20 sample locations, for each sample location, 5 point clouds were acquired from different directions. We compared the performance of PCLA to the method using only one point cloud (denoted as One-PC). As shown in Fig. 13(b), the accuracy of localization can be doubled using PCLA. Specifically, the 50-percentile error is reduced from 2.43m to 1.17m, and the 80-percentile error is reduced from 3.17m to 1.92m.

In addition, we compared the localization performance of our CloudNavi with images based [9], WiFi fingerprinting based, and point cloud based methods. Note that, the point cloud based method was implemented using Xtion PRO Live. The localization errors are shown in Fig. 13(c). CloudNavi achieves a median location error of 1.17m, which is significantly smaller than the image based (2.4m) and WiFi based (4.8m) methods. It is clear that the localization accuracy of CloudNavi is superior to the image based approaches due to the use of depth information. While, the point cloud based method using Xtion PRO Live (Xtion-based) achieves a smaller localization error than our CloudNavi system. That is because high-precision 3D cameras were used in the point cloud based method while low quality-commodity mobile devices were used in our CloudNavi system.

5.3.5 Accuracy of Walking Traces

In this section, we tested the tracking error of the VIO algorithm and compared it with the dead reckoning method. For this evaluation, the volunteers were asked to walk randomly in office building and Yuefang shopping mall. The CDF of the tracking error for the VIO algorithm and the dead reckoning method is shown in Figs. 14(a-b). The VIO



(a) Impact of environment variations (b) Impact of crowds (c) Impact of color variation

Fig. 16: Location error under the factors of environment variations, crowds, and color variation.

algorithm performs better than dead reckoning. It achieves a median tracking error of 0.7m, as compared to 2.1m for dead reckoning in office and a median tracking error of 0.8m, as compared to 2.4m for dead reckoning in mall. That is because the accuracy of mobile devices' pose calculated by the VIO algorithm is more accurate than that achieved by the dead reckoning method.

5.3.6 Keypoints Detected in Point Cloud

The volunteers were asked to walk along corridors in office and 200 point clouds were captured in two situations, this is, with and without pose estimation. Figure 15(a) shows the number of keypoints detected with/without pose estimation. It can be seen that the number of keypoints detected in point clouds acquired with pose estimation is always larger than that acquired without pose estimation. Thus, pose estimation can be used to improve the feature richness of point clouds. Furthermore, we investigated the accuracy of point cloud matching with/without pose detection. Figure 15(b) shows that better matching accuracy can be obtained when the pose of mobile device is estimated for point cloud acquisition. That is, the average matching error is about 1.9% using pose estimation. In contrast, the average matching error is about 6.8% without pose estimation. It is further demonstrated that the captured point clouds usually have rich features with pose estimation.

5.3.7 Response Delay

We deployed CloudNavi using a Google project tango tablet and a server equipped with a 32GB RAM, an i7 CPU processor, and a 12GB Titan GPU. The response delay were evaluated for two cases: initial phase and navigation phase. The initial phase includes data uploading, global localization, destination query, and path planning. The navigation phase includes data uploading and local localization. Since the WiFi radio transmission speed is about 2MB/s, point cloud uploading takes 0.1s on average. Global localization takes 3.6s on average, and local localization takes 0.4s on average. Destination query costs only 0.002ms on a library with 185 labeled destinations, and path planning takes 0.1s. In total, initial phase costs about 3.8s (0.1+3.6+0.1s), the navigation phase costs about 0.6s (0.1+0.4+0.1s). It is reasonable that the initial localization phase takes relatively long time while the navigation phase is highly efficient. Note that, the response delay can further be reduced using more powerful machines.

5.3.8 Energy Consumption

If point clouds are acquired using cameras and are uploaded using WiFi network, relatively high power consumption can be incurred. We tested the energy consumption of our CloudNavi mobile client software, including those consumed by cameras and WiFi network. The energy is estimated using the PowerTutor profiler [31] installed in the Google project tango tablet. During the experiment, we turned off all background applications and additional hardware components. The energy consumed by camera and WiFi network is 6.9 Joule and 1.6 Joule for a point cloud, respectively. Compared to the battery capacity of 20k Joules, point cloud capturing and uploading do not constitute any signification power consumption for a mobile device [27].

5.3.9 Other Factors

In this section, we tested the performance of our CloudNavi system with respect to environment variations, crowds, and color variation.

Impact of Environment Variations. To analyze the effect of environment variations, we conducted a set of tests by moving some objects, such as chairs, desks, posters. 12 point clouds were acquired and CloudNavi was used to locate the camera using the updated 3D path-map. Experimental results are shown in Fig. 16(a). It can be seen that CloudNavi can successfully identify the location with comparable accuracy.

Impact of Crowds. Since the points within a depth range (i.e., 0.5∼4m for a Google project tango tablet) and the field of view (i.e., 57.5×45 for a Google project tango tablet) can be recorded by a mobile device, it is reasonable that we set the maximum number of volunteers to 6. In order to analyze the effect of human existence, 0, 1, 3, and 6 volunteers were asked to keep walking in the 3D camera view in our experimental, respectively. Experimental results are shown in Fig. 16(b). It can be seen that the location can be successfully estimated. The accuracy of localization clearly demonstrates the effectiveness of the point clouds acquired from the space between the head level of crowds and ceilings.

Color Variation. Furthermore, we tested our CloudNavi system under color variations. A set of tests were conducted by placing two electronic display screens on both sides of our camera. 12 point clouds were acquired at the same location, and the camera localization was estimated using these point clouds and the 3D path-map. Experimental results are shown in Fig. 16(c). It can be seen that the location can be

successfully estimated with comparable accuracy. Therefore, our CloudNavi system is robust to the color variation. That is because our CloudNavi system does not rely on any color information and it is therefore robust to color variations.

6 Discussion and Future Work

Ubiquitous Point Clouds. Although point clouds can only be captured by Google project tango mobile device currently, 3D cameras (e.g., Intel realsense[4], TI 3D camera[5]) will be widely equipped with mobile devices in the future. Point clouds will become as ubiquitous as 2D images. The design of CloudNavi is motivated by the development of 3D cameras.

Limitations of the Experiments. We only tested the performance of indoor navigation on point clouds acquired by Google project tango tablet. Thai is because Google project tango tablet is the only available tablet with a 3D sensor in market. In addition, our method provides a general framework and can be used on different hardware platforms. In the future, once more tablets with 3D sensors are available, we can do more experiments.

The Scalability of Navigation Service. It is clear that the number of participated data collectors determines both the navigation performance and the scalability of CloudNavi. If only a few data collectors participate in the service, the collected point clouds would be insufficient. This is a common challenge faced by most crowdsourcing-based systems. At the initial stage, CloudNavi provider should contribute some useful traces to bootstrap the navigation service, then, upcoming users can be navigated to their destinations using CloudNavi. At the same time, a rewarding mechanism can be designed to encourage users to contribute long walking trace and to record their destinations. Through such a simple way, the navigation system can collect sufficient traces to expand its coverage area and to enhance its service capability.

Multiple Floors. A user may walk on multiple floors. Therefore, CloudNavi will be improved for multiple floor navigation in our future work. Inspired by the works [7], [32], the motion states of a user can be recorded by inertial sensors when walking across different floors (e.g., through stairs, escalators, or elevators). CloudNavi can incorporate these motion states to achieve multiple floor navigation.

Building Types. The 3D path-map generated by CloudNavi mainly focus on the connection between corridors and other rooms that lying along both sides of the corridor, such as in office building and shopping mall. The proposed solution may be failed in large open environments, such as hall, gymnasium, museum and lobby, where the movement of a user is difficult to be characterized. Ubiquitous indoor navigation services in large open indoor environments will be our future work.

Using Semantics from Crowdsourced Point Clouds. We plan to extend our system to support automatic semantic labelling of 3D path-map by object identification. Object identification is a very active research topic in computer vision community [15], [33]. It is reasonable to use such technique to decrease the dependency on crowdsourced labeling and to improve its flexibility.

4. http://click.intel.com/realsense.html
5. http://www.ti.com

7 Related Work

Indoor navigation has been extensively studied and can be roughly categorized as follows.

SLAM. The task of Simultaneous localization and mapping (SLAM) is to reconstruct maps in an unexplored environment using different sensors (e.g., laser sensors, cameras, and odemetry) [34], [35], [36], [37]. We share similar goals with SLAM, however, our problem has several significant differences. First, a robot systematically explores all accessible areas in an indoor space using SLAM, while our CloudNavi system is a crowdsourcing system. It is more challenging to construct maps using a crowdsourcing system. Second, a robot is usually equipped with high precision sensors (e.g., laser finders) in SLAM, while our CloudNavi system uses low precision 3D cameras (e.g., Google project tango tablet). Third, the locomotion of humans is more complicated than that of indoor robots. Therefore, the crowdsensed data is not only noisy but also piece-wise, as they are usually collected by unorganized users

Infrastructure Based Navigation Services. Existing indoor navigation systems usually rely on indoor localization systems. Radar [1], LiFs [2], Ubicarse [3], SpotFi [4], and the works presented in [38], [39], [40] use existing WiFi infrastructure to perform indoor localization. Epsilon [6], Luxapose [5], Spinlight [41], and Lightitude [42] achieve high indoor localization accuracy through VLC. Although these systems are different in details, all of them rely on a particular infrastructure. In contrast, our CloudNavi system makes a first step towards a ubiquitous indoor navigation using point clouds only. It has two advantages. First, our method is more applicable to scenes without WiFi or VLC infrastructures. Second, our method is orthogonal to existing WiFi-based and VLC-based methods for indoor navigation systems. They can be combined to further improve the accuracy and stability of navigation systems.

Self-deployable Navigation Service. Recent techniques (including Travi-Navi [11], FollowMe [12] and iMoon [13]) can provide self-deployable indoor navigation service. Travi-Navi [11] uses the images of pathway to track the user's walking progress. FollowMe [12] uses the geomagnetic field to guide a user by providing the "scent" left by the leaders or previous travelers. iMoon [13] uses crowdsourced photos to build a smartphone-based indoor navigation system. CloudNavi is significantly different from and better than these systems in many aspects. **First**, CloudNavi uses structural information of indoor space (i.e., point cloud) to perform navigation, it is more stable, robust, and accurate than signal (e.g., RF, magnetic data, and images) based systems. Therefore, it satisfies the requirements for a ubiquitous indoor navigation system. **Second**, The existing systems rely on IMU sensor (including gyroscope and accelerometer) based dead reckoning method [11], [12], [13]. Since the dead reckoning method is highly sensitive to usage diversity (e.g., different hand holding, various walking speed), it is difficult to accurately estimate the steps, walking directions, and step length. In contrast, CloudNavi uses the VIO algorithm to reduce the effect of usage diversity. **Third**, existing systems can only be launched at some predetermined positions (e.g., the building entrances). In contrast, CloudNavi can be started from any position on a related indoor pathway.

Besides, there are significant amount of literatures for localization using point clouds [43], [44], [45], [46]. These methods used high-precision 3D cameras to acquire point clouds. However, our CloudNavi system used commodity mobile devices equipped with low-quality 3D sensors. Additionally, the traces and point clouds are also highly noisy due to error accumulation. Therefore, our CloudNavi is a ubiquitous system.

Note that, although it is possible to develop a self-deployable indoor navigation system using the dead reckoning method [7], [8], [47], such system extremely depends on indoor maps and suffers from low navigation accuracy. CloudNavi does not require floor maps and can achieve a high navigation accuracy.

Indoor Map Construction. Indoor map based navigation systems are also available. Google Maps started to provide detailed indoor floorplans. However, they only have a small fraction of millions of indoor environments. First, buildings owners may not allow the sharing of their floorplans. Second, manual generation of these maps requires slow, labor-intensive tasks, and they are subject to intentionally incorrect data produced by malicious users [48]. To resolve these problems, the research community has recently embarked to develop automatic construction method for indoor floorplans by exploiting crowdsourced data collected by mobile device users [14], [27], [32], [49]. CrowdInside [32] and the work in [49] are dependent on aggregated user motion traces derived from inertial data. Jigsaw [27] and CrowdMap [14] use both images and inertial data to reconstruct the indoor floorplan. These systems [14], [27], [32], [49] are easy-to-deploy, infrastructure-free, and universal. However, they are not robust and accurate. Therefore, none existing technique is as ubiquitous as our CloudNavi system.

8 Conclusion

Indoor navigation is an emerging research area with attractive and promising applications in daily life. This paper presents a point cloud based indoor navigation solution (namely CloudNavi). It pushes the design of indoor navigation to the extreme on five dimensions: accurate, easy-to-deploy, infrastructure-free, robust, and universal. Extensive experiments were conducted on a large number of point clouds acquired in office building and shopping mall. Experimental results show that CloudNavi achieves promising indoor navigation performance and outperforms several state-of-the-art systems.

Acknowledgments

This work is supported in part by the National Natural Science Foundation for Outstanding Excellent young scholars of China under grant No.61422214, National Basic Research Program (973 program) under Grant No. $2014CB347800$, the Program for New Century Excellent Talents in University and Distinguished Young Scholars of National University of Defense Technology under grant No.JQ14-05-02, the Preliminary Research Funding of National University of Defense Technology under grant NO.ZDYYJCYJ20140601, and National Natural Science Foundation of China under Grants No. 61402513 and No. 61471371.

References

[1] P. Bahl and V. Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *Proc. of IEEE Infocom*, Israel, 2000.

[2] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proc. of ACM MobiCom*, Istanbul, Turkey, 2012.

[3] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. In *Proc. of ACM MobiCom*, Maui, Hawaii, 2014.

[4] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using WiFi. In *Proc. of ACM SIGCOMM*, London, United Kingdom, 2015.

[5] Y. Kuo, P. Pannuto, K. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proc. of ACM MobiCom*, Maui, Hawaii, 2014.

[6] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A visible light based positioning system. In *Proc. of USENIX NSDI*, Seattle, WA, 2014.

[7] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. Choudhury. Unsupervised indoor localization. In *Proc. of ACM MobiSys*, United Kingdom, 2012.

[8] G. Shen, C. Zhuo, P. Zhang, M. Thomas, and Y. Zhang. Walkie-Markie: Indoor pathway mapping made easy. In *Proc. of USENIX NSDI*, Lombard, IL, 2013.

[9] R. Gao, Y. Tian, F. Ye, K. Bian, Y. Wang, T. Wang, and X. Li. Sextant: Towards ubiquitous indoor localization service by phontotaking of the environment. *IEEE Transactions on Mobile Computing*, 13(9):1–14, 2014.

[10] S. Wang, S. Fidler, and R. Urtasun. Lost shopping! monocular localization in large indoor spaces. In *Proc. of IEEE ICCV*, Santiago, Chile, 2015.

[11] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao. TraviNavi: Self-deployable indoor navigation system. In *Proc. of ACM MobiCom*, Maui, Hawaii, 2014.

[12] Y. Shu, K. Shin, T. He, and J. Chen. Last-Mile navigation using smartphones. In *Proc. of ACM MobiCom*, Paris, France, 2015.

[13] J. Dong, Y. Xiao, M. Noreikis, Z. Ou, and A. Yla-Jaaski. iMoon: Using smartphones for image-based indoor navigation. In *Proc. of ACM SenSys*, Seoul, South Korea, 2015.

[14] S. Chen, M. Li, K. Ren, and C. Qiao. Crowdmap: Accurate reconstruction of indoor floor plans from crowdscourced sensor-rich videos. In *Proc. of IEEE ICDCS*, Ohio, USA, 2015.

[15] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wang. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2270–2287, 2014.

[16] T. Petri, K. Kalin, M. Lorenz, C. Federico, S. Olivier, and P. Marc. Live metric 3D reconstruction on mobile phones. In *Proc. of IEEE ICCV*, Sydney, Australia, 2013.

[17] K. Kalin, T. Petri, S. Pablo, and P. Marc. Turning mobile phones into 3D scanners. In *Proc. of IEEE CVPR*, Columbus, Ohio, 2014.

[18] R. Rusu and C. Steve. 3D is here: Point cloud library (pcl). In *Proc. of IEEE ICRA*, Shanghai, China, 2011.

[19] E. Delponte, F. Isgro, F. Odone, and A. Verri. SVD-matching using SIFT features. *Graphical models, Elsevier*, 68(5):415–431, 2006.

[20] Y. Guo, M. Sohel, F.and Bennamoun, J. Wan, and M. Lu. An accurate and robust range image registration algorithm for 3D object modeling. *IEEE Transactions on Multimedia*, 16(5):1377–1390, 2014.

[21] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *Proc. of IEEE ICRA*, Karlsruhe, Germany, 2013.

[22] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze. Segmentation of unknown objects in indoor environments. In *Proc. of IEEE IROS*, Vilamoura, Algarve, 2012.

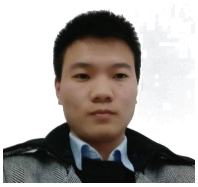
[23] P. McCullagh. Generalized linear models. *European Journal of Operational Research*, 16(3):285–292, 1984.

[24] J. Gui, D. Gu, and H. Hu. Robust direct visual inertial odometry via entropy-based relative pose estimation. In *Proc. of IEEE ICMA*, Beijing, China, 2015.

[25] L. Coedella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372, 2004.

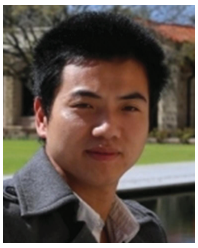
[26] O Akyilmaz. Total least squares solution of coordinate transformation. *Survey Review*, 39(303):68–80, 2007.

[27] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *Proc. of ACM MobiCom*, Maui, Hawaii, 2014.
[28] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *Proc. of ACM MobiCom*, Istanbul, Turkey, 2012.
[29] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proc. of IEEE ICRA*, Sacramento, CA, 1991.
[30] S. Skiena. Dijkstra's algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, pages 225–227, 1990.
[31] M. Gordon. PowerTutor: A power monitor for Android-based mobile platforms. 2013.
[32] M. Alzantot and M. Youssef. Crowdinside: Automatic construction of indoor floorplans. In *Proc. of SIGSPATIAL*, Redondo Beach, California, 2012.
[33] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. An integrated framework for 3-D modeling, object detection, and pose estimation from point-clouds. *IEEE Transactions on Instrumentation and Measurement*, 64(3):684–693, 2015.
[34] S. Fu, H.Y. Liu, L.F. Gao, and Y.X. Gai. SLAM for mobile robots using laser range finder and monocular vision. In *Proc. of IEEE M2VIP*, Xiamen, China, 2007.
[35] J. Engel, T. Schops, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. of Springer ECCV*, Zurich, 2014.
[36] L. Contreras and W. Mayol-Cuevas. Trajectory-driven point cloud compression techniques for visual SLAM. In *Proc. of IEEE/RSJ IROS*, Hamburg, Germany, 2015.
[37] H. Abdelnasser, R. Mohamed, A. Elgohary, M.F. Alzantot, H. Wang, S. Sen, R. Choudhury, and M. Youssef. SemanticSLAM: Using environment landmarks for unsupervised indoor localization. *IEEE Transactions on Mobile Computing*, 15(7):1770–1782, 2016.
[38] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed. Joint indoor localization and radio map construction with limited deployment load. *IEEE Transactions on Mobile Computing*, 14(5):1031–1043, 2015.
[39] S. Nikitaki, G. Tsagkatakis, and P. Tsakalides. Efficient multichannel signal strength based localization via matrix completion and bayesian sparse learning. *IEEE Transactions on Mobile Computing*, 14(11):2244–2256, 2015.
[40] S.H. Fang and T.N. Lin. Principal component localization in indoor WLAN environments. *IEEE Transactions on Mobile Computing*, 11(1):100–110, 2012.
[41] B. Xie, G. Tan, and T. He. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *Proc. of ACM SenSys*, Seoul, Korea, 2015.
[42] Y. Hu, Y. Xiong, W. Huang, X.Y. Li, Y. Zhang, X. Mao, and C. Wang. Lightitude: Indoor iositioning using ubiquitous visible lights and COTS devices. In *Proc. of IEEE ICDCS*, Ohio, USA, 2015.
[43] H. Fukai, J. Takagi, and G. Xu. Robust and fast self localization by 3d point cloud. In *Proc. of IEEE HSI*, Yokohama, Japan, 2011.
[44] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach. TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In *Proc. of IEEE ICIP*, Orlando, Florida, 2012.
[45] B.K. Kim. Indoor localization and point cloud generation for building interior modelings. In *Proc. of IEEE RO-MAN*, Gyeongju, Korea, 2013.
[46] C. Jaramillo, I. Dryanovski, R.G. Valenti, and J. Xiao. 6-DoF pose localization in 3D point-cloud dense maps using a monocular camera. In *Proc. of IEEE ROBIO*, Shenzhen, China, 2013.
[47] Z. Xiao, H. Wen, A. Markham, and N Trigoni. Indoor tracking using undirected graphical models. *IEEE Transactions on Mobile Computing*, 14(11):2286–2301, 2015.
[48] M. Elhamshary, M. Youssef, A. Uchiyama, H. Yamaguchi, and T. Higashino. Transitlabel: A crowd-sensing system for automatic labeling of transit stations semantics. In *Proc. of ACM MobiSys*, Florence, Italy, 2016.
[49] Y. Jiang, Y. Xiang, X. Pan, K. Li, Q. Lv, R. P. Dick, and M. Hannigan. Hallway based automatic indoor floorplan construction using room fingerprints. In *Proc. of ACM UbiComp*, Zurich, Switzerland, 2013.



Xiaoqiang Teng received the B.S. degree from the School of Mechanical Engineering, Shenyang University of Technology, China, in 2013. Currently, he is a Ph.D. student in the College of Information System and Management, National University of Defense Technology, Changsha, China. His research interests include mobile computing, pervasive computing, and computer vision.

Deke Guo received the B.S. degree in industry engineering from Beijing University of Aeronautic and Astronautic, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from National University of Defense Technology, Changsha, China, in 2008. He is a Professor with the College of Information System and Management, National University of Defense Technology, Changsha, China. His research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks.



Yulan Guo received his B.Eng. and Ph.D. degrees from National University of Defense Technology (NUDT) in 2008 and 2015, respectively. He is currently an assistant professor with the College of Electronic Science and Engineering at NUDT. He was a visiting PhD student at the University of Western Australia (UWA) from 2011 to 2014. He authored more than 30 peer reviewed journal and conference publications (including TPAMI and IJCV), and one book chapter. He served as a reviewer for more than 20 international journals (e.g., IJCV). His research interests include computer vision and pattern recognition, particularly on 3D feature extraction, 3D modeling, 3D object recognition, and 3D face recognition.

Xiaolei Zhou received the BA degree from the Information Management Department, Nanjing University, P.R. China, in 2009, and the MS degree in military science from the National University of Defense Technology, P.R. China, in 2011. He is currently working toward the Ph.D. degree in management science and engineering from National University of Defense Technology, Changsha, China. His current research interests include wireless sensor networks, Internet of things, and data center networking.

Zhong Liu received the B.S. degree in Physics from Central China Normal University, Wuhan, Hubei, China, in 1990, and the Ph.D. degree in management science and engineering from National University of Defense Technology, Changsha, China, in 2000. He is a professor with the College of Information System and Management, National University of Defense Technology, Changsha, China. His research interests include information systems, cloud computing, and big data.