Indoor Floor Plan Construction through Sensing Data Collected from Smartphones

Zhe Peng, Shang Gao, Bin Xiao, *Senior Member, IEEE,* Guiyi Wei, *Senior Member, IEEE,* Songtao Guo, *Member, IEEE,* and Yuanyuan Yang, *Fellow, IEEE*

Abstract—With the development of sensing technology, smartphones can provide various kinds of data, including inertial sensing data, WiFi data, depth data and images. These data make it possible to construct accurate indoor floor plans that are the critical foundations of flourishing indoor location-based services for smartphone. However, even with the popular crowdsourcing approach, the wide construction of indoor floor plans has not yet to be realized due to the intensive time consumption. In this paper, we utilize deep learning techniques to build PlanSketcher, a system that enables one user to construct fine-grained and facility-labelled indoor floor plans accurately. First, the proposed system extracts novel integrated features to recognize diverse landmarks. Second, traverse-independent hallway topologies are constructed based on the sensing data, depth data and images through the proposed hallway construction algorithms. Finally, PlanSketcher constructs the room shape and labels recognized facilities in their corresponding positions to generate a complete indoor floor plan. Because PlanSketcher exploits different kinds of data collected from smartphones with new feature extraction method, it can obtain accurate indoor floor plan topology and facility labels. We implement PlanSketcher and conduct extensive experiments in 3 large indoor settings. The evaluation results show that the 90th percentile accuracy of positions and orientations of facilities are $1m \sim 2.5m$ and $4° \sim 6°$, while $85\% \sim 95\%$ facilities are recognized and labelled precisely.

Index Terms—Indoor floor plan construction, sensing data, facility label, energy consumption, smartphone.

1 Introduction

With the development of mobile communication networks and Internet of Things (IoT), a large amount of data are generated on a daily basis. Such huge data make the network systems super-complex and very difficult to model and manage. Benefitted from the advanced data analysis techniques (such as the artificial intelligence), the novel data-driven network management will enable us to dynamically and adaptively meet with the spatio-temporal network demands in the most resource-aware and resource-smart manner. In various spatio-temporal network demands, location based service (LBS) [1], [2] is one of the most important network service applications. Abundant outdoor location based services [3], such as finding nearby point-of-interests (POIs), localization, and navigation, have been provided through the online digital maps (e.g., Google Maps). However, for indoor environments where people spend over 80% of the time, similar indoor location based services are still not widely supported.This is because the fundamental infrastructures, i.e., indoor digital maps, are extremely scarce and unavailable in most buildings. In order to construct indoor digital maps, service providers have to conduct effort-intensive and time-consuming business negotiations with building owners or operators to collect the floor plans, or wait for them to voluntarily upload such data. Neither is conducive to large scale coverage in short time. Therefore, under many resource constraints such as user privacy and computation power of mobile devices, accurate and scalable indoor floor plan construction method at low costs is urgently needed.

Driven by the flourishing of smartphones and their capability of processing diverse data, we are motivated to construct indoor floor plans by these new techniques. Through analysing sensing data gathered from smartphones, we can capture crucial geographic features of indoor environments [4]. Many efforts have been devoted to construct indoor floor plans using smartphones by crowdsourcing [5], [6], [7]. CrowdInside [8] first utilized inertial sensing data to automatically generate user traces and construct indoor pathways. Walkie-Markie [9] used locations where the trend of WiFi signal strength changed as the landmarks to construct indoor floor plans. Jigsaw [10] inferred the approximate structures and shapes of hallways and rooms from the inertial data and images. CrowdMap [11] utilized sensor-rich video data from mobile users for indoor floor plan construction. Shopprofiler [12] resorted to SSIDs collected in each shop to infer the categories and names of shops.

However, there are some limitations in current indoor floor plan construction systems. First, most of the existing systems utilize crowdsourcing to collect data, which will encounter the slow-start problem. These systems often need users to install applications in their smartphones which is not preferred in consideration of privacy and security. If not enough users participate, the collected data are insufficient to construct accurate indoor floor plans. Collecting a large amount of sensing data will consume much energy in smartphone, which is a critical constraint for these methods. Second, to construct accurate indoor floor plans, users

- *Z. Peng, S. Gao and B. Xiao are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. E-mail: {cszpeng, cssgao, csbxiao}@comp.polyu.edu.hk.*
- *G. Wei is with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310000, China. E-mail: weigy@zjgsu.edu.cn.*
- *S. Guo is with the College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China. E-mail: guosongtao@cqu.edu.cn.*
- *Y. Yang is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook NY 11794, USA. E-mail: yuanyuan.yang@stonybrook.edu.*

Manuscript received October 30, 2017; revised April 30, 2018; revised July 4, 2018; accepted July 24, 2018.

need to traverse all hallways and corners in indoor environment. However, hallways are often blocked by furniture or other objects. Third, the constructed indoor floor plans with existing methods often lack valuable labels to guide users. Users may not find their destinations or know the surroundings with a floor plan without facility labels. Although Shopprofiler [12] recognizes the categories and names of shops and labels them in the floor plans, the accuracy of recognition can still be improved. The reason is that some areas are unreachable and the WiFi access service is usually provided by the shopping mall which results in the same WiFi SSID in all shops. It is extremely crucial to provide a user-friendly floor plan with facility labels for location-based services. As a consequence, we ask the following question: *Is it feasible to enable a user to easily construct his own indoor floor plans by himself without compromising on location accuracy and facility labels?*

In this paper, we provide an affirmative answer through the systematic design and implementation of *PlanSketcher*, which utilizes deep learning technique to construct facility-labelled and highly fine-grained indoor floor plans using smartphone. Different from current schemes, the proposed PlanSketcher constructs indoor floor plans with various data collected by a single person. This manner can save extensive manpower and energy consumption to effectively construct indoor floor plans and protect the user's privacy. Utilizing the mature object detection technique, the proposed system can also provide abundant valuable labels to distinguish various facilities. Moreover, the constructed indoor floor plans with PlanSketcher are highly fine-grained (e.g., accuracy within $1m \sim 2.5m$), thus users can locate their positions more precisely and receive much better location-based services.

Compared with previous works, PlanSketcher has the following advantages: (1) Recognizing more landmarks with high accuracy from collected data to improve the accuracy of the constructed indoor floor plans. (2) Constructing more accurate hallway topology from the landmarks, depth data and images with less energy consumption. (3) Constructing more accurate room shape with less energy consumption and labeling recognized facilities precisely in the indoor floor plans.

In this paper, we make following contributions:

- We propose the *PlanSketcher* system architecture to construct fine-grained and facility-labelled indoor floor plans with less energy consumption in smartphone.
- We develop novel *landmark recognition* approaches to detecting various landmarks from the inertial sensing data, WiFi signals and images. Adequate landmarks are recognized such that our system is reasonably accurate.
- We present *hallway construction* algorithms to construct the topology of hallways with high fine-granularity. A new method is proposed to construct hallways that a user has traversed by fusing the depth data, inertial sensing data and images. Considering the robustness of system, we also propose the corresponding measurement calibration method. In addition, a novel Non-Traversed Hallway Construction method is designed to construct the hallways which are not traversed by a user.
- We design a *labelled room construction* method to construct the room shape from depth data and label facilities in their corresponding positions to generate a complete indoor floor plan. Different from most previous works, the proposed method does not need user to traverse in the room and can achieve a high accuracy and provide users valuable room information.
- In addition, we develop the prototype and conduct extensive evaluations across 3 large complex indoor settings on a variety of mobile devices. The results illustrate that our proposed PlanSketcher outperforms the state-of-the-art methods with smaller position and orientation error, more recognized facilities and less energy consumption.

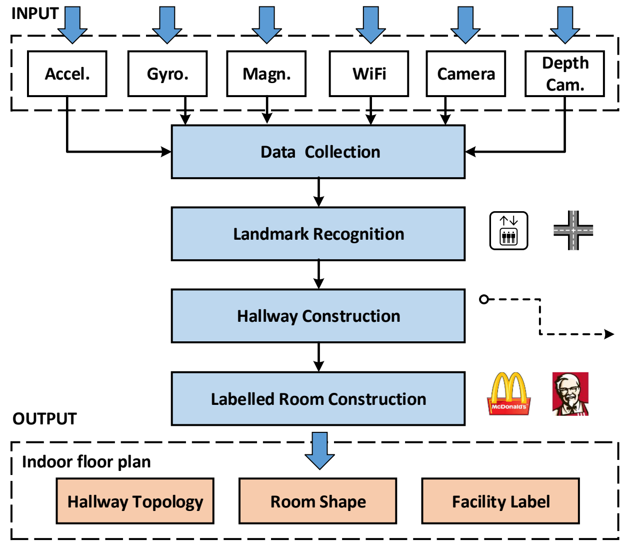


Fig. 1: PlanSketcher System.

The rest of this paper is organized as follows. We first provide the system overview of PlanSketcher in Section II. Then we detail each module of our system in Section III, IV and V. The implementations and evaluation are illustrated in Section VI. We further review the related works in Section VII and conclude our work in Section VIII.

2 Design Overview

PlanSketcher system leverages inertial sensing data, depth data and images collected from smartphones to construct fine-grained and facility-labelled indoor floor plans with less energy consumption. High-performance sensors integrated in modern smartphones provide abundant information to depict user motion patterns and indoor architectural structures. With the rise of Augmented Reality (AR) technique, the time-of-flight (TOF) depth camera has been equipped within more and more smartphones. The TOF depth camera can obtain the distance between the camera and the subject for each point in the image through measuring the time-of-flight of a laser light. Images captured by cameras equipped with smartphones offer luxuriant visible and valuable description about surroundings. Fig. 1 sketches the PlanSketcher system architecture. In the following, we describe four modules of the system briefly.

Data collection. Various inertial sensing data, WiFi data, depth data and images are gathered from smartphone while users move around in the indoor environment. Smartphone sensors such as accelerometer, gyroscope and magnetometer could measure acceleration, rotational velocity and magnetic field respectively. TOF depth camera obtains the distance between the camera and the subject for each point in the image. In addition, users leverage cameras to capture images of corners and facilities.

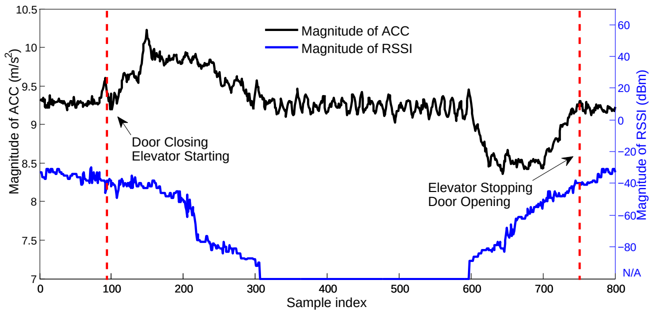


Fig. 2: RSSI value drops and reverses when the user enters and leaves the elevator.

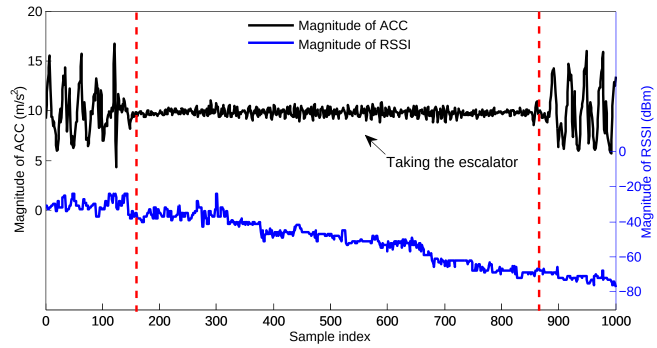


Fig. 3: RSSI and magnetic field changes when the user takes the escalator.

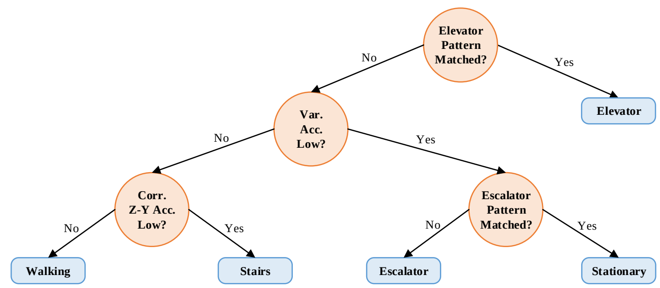


Fig. 4: A generalized framework for real landmark recognition.

Landmark recognition. PlanSketcher recognizes real landmarks and virtual landmarks existing in a building. We first extract novel efficient integrated features (inertial sensing data and WiFi signals) to recognize the real landmarks, which involve elevators, escalators and stairs etc. Then two photographing manners: Corner Photographing Manner (CPM) and Facility Photographing Manner (FPM) are proposed to gather various information to recognize the virtual landmarks (hallway corners, stores and restrooms etc). Moreover, to distinguish various hallway corners, we define the novel and unique fingerprint for each corner through recognizing the store or room names from images with advanced deep learning technique.

Hallway construction. With the recognized landmarks, the topology of hallways is constructed based on the inertial sensing data, depth data and images. We divide the hallway construction into two types: user traversed hallways and user non-traversed hallways. A new method is proposed to construct hallways that a user has traversed by fusing the depth data, inertial sensing data and images. Considering the robustness of system, we also proposed the corresponding measurement calibration method. We propose the novel Non-traversed Hallway Construction method to construct complete floor plans through matching the fingerprints of corners, even hallways are not traversed by the user.

Labelled room construction. To generate a relatively complete indoor floor plan, PlanSketcher constructs room shape from depth data and label recognized facilities in their corresponding positions. Different from most previous works, the user collects 3D depth data of the ceiling at the entrance instead of traversing in the room to construct room shape. Utilizing the projection of 3D depth data in the horizontal plane, PlanSketcher detects the edges and infer the geometric vertices. To label the recognized facilities (such as elevators, escalators, shops, etc), PlanSketcher calculates the position of facility entrance and its orientation.

Different from the data collection approaches adopted in prior works [8], [12], [13], our proposed system could construct accurate labelled indoor floor plans by a single person from a small amount of data. It is assumed that the participate user collects data though our defined photographing manners: Corner Photographing Manner (CPM) and Facility Photographing Manner (FPM).

3 Landmark Recognition

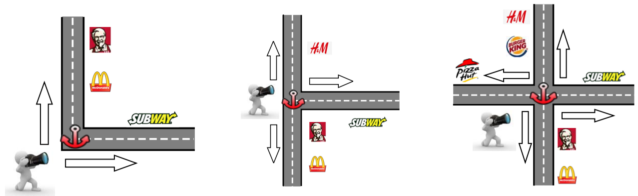
In this section, we describe the method to extract novel integrated features from human motion patterns and images to recognize various landmarks in a building. Special architectural structures, such as elevators, hallway corners and rooms, are critical to construct indoor floor plans because they describe the principal architectural features of a building. The floor plans could be constructed more accurately if these structures are recognized and located precisely.

We consider these structures as landmarks which are divided into two categories: real landmarks and virtual landmarks. Real landmarks compel people to move in the expectable motion patterns. Virtual landmarks are detected through matching two proposed photographing manners and recognizing the captured images.

3.1 Real Landmark

When people are moving within the elevators, escalators and stairs, sensing data collected by acceleration and magnetometer will present distinct features. These real landmarks are distinguished by recognizing unique features extracted from the magnitude of accelerometer and magnetometer [8], [14]. However, the recognition is erroneous (false positive or false negative) because the sensing data are easily disturbed by the unstable environment or the body's moving vibration.

To solve this problem, we combine the Received Signal Strength Indicator (RSSI) value of WiFi connection with the sensing data to recognize three common landmarks: elevators, escalators, and stairs. This idea is inspired by the observation of correlation between the RSSI value and human position. Fig. 2 plots the differentials of RSSI value against accelerometer magnitude when the user takes an elevator. This figure shows that when the user enters the elevator and the door is closing, the RSSI value drops to an undetectable level due to the shielding effect of metal materials. When the elevator stops to open the door, the RSSI value reverses to the normal level. Similarly, we can also separate the escalator from the human stationary case utilizing the variance of RSSI field and magnetic field. In both cases, the accelerometer magnitude tends to be stable, but the RSSI field and



(a) A corner. (b) A T-junction. (c) A crossroad.

Fig. 5: Corner Photographing Manner (CPM).

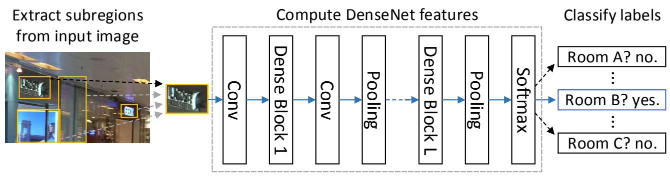


Fig. 6: Illustration of multiple rooms recognition via deep learning.

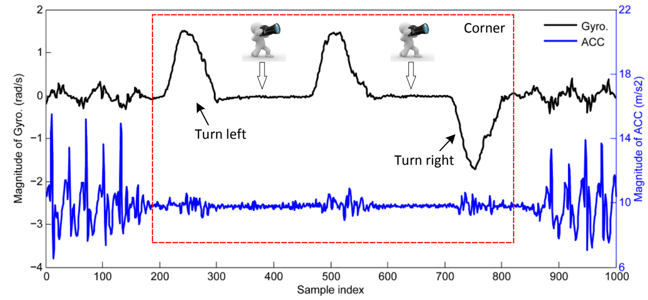


Fig. 7: Gyroscope and accelerometer measurements when the user photographs at the corner.

Fig. 8: Two stores are detected from the image captured at the corner.

magnetic field will change obviously when the user moves within the escalator (shown in Fig. 3).

In Fig. 4, we give a generalized framework to recognize different real landmarks and movements, including elevators, escalators, stairs, walking, and stationary. In the decision tree, the top level can recognize the elevator from the proposed pattern (as shown in Fig. 2). Then the second level distinguishes the variable velocity classes (walking and stairs) from the constant velocity classes (escalator and stationary) based on the variance of the acceleration. The last level separates walking and stairs by the correlation of accelerations in Z and Y axes, and separates escalator and stationary by the proposed pattern (as shown in Fig. 3). Thus, by integrating the RSSI value with the sensing data, PlanSketcher is inherently more robust than existing methods which merely look into the acceleration and magnetometer reading.

3.2 Virtual Landmark

Hallway corners and rooms depict the basic characters of indoor environments. Because images could provide more diverse and reliable descriptions about indoor settings than the sensing data, we recognize corners and rooms from images and model them as the virtual landmarks. Two photographing manners: Corner Photographing Manner (CPM) and Facility Photographing Manner (FPM) are proposed to gather data to recognize virtual landmarks. Moreover, to distinguish various corners, we define the novel and unique fingerprint for each corner through recognizing the room names from images with advanced deep learning technique.

Corner Photographing Manner (CPM). In PlanSketcher, we design Corner Photographing Manner (CPM) which defines a few actions for the user to collect the inertial data and images of corner. As illustrated in Fig. 5, when the user is traversing a corner, he takes a photo of the hallway with some rooms in one direction and then spins his body to photograph in another direction. During this process, the gyroscope and accelerometer offer the angular velocity and acceleration measurements, and the photographing behaviors are also recorded.

Detecting the corners from the CPM. As shown in Fig. 7, the bumps (upward or downward) of gyroscope readings show that the user is turning left or right. During the two bumps, the user is capturing a photo of the hallway. Because the user just rotates his body during the photographing, the accelerometer magnitudes stay stable with less jitters. Triggered by the photographing behavior, the special characteristics emerged in the inertial data can indicate that the user is at the corner.

Distinguishing the corners with the fingerprints. Leveraging the advanced deep learning technique, we propose a method to precisely recognize multiple room names in a hallway image based on their various logos. In order to distinguish various corners, we further define the *fingerprint* of corner utilizing the recognized room names.

Convolutional neural network (CNN) is a dominant deep learning approach for object recognition in images. It can predict the label of a given image after training the network. However, because more than one room may be captured in a hallway image, directly using CNN to classify the image cannot recognize multiple rooms simultaneously. Thus, in PlanSketcher, we precisely recognize multiple rooms in a hallway image from their logos based on the object detection technique [15] and the latest CNN architecture, i.e., deep dense convolutional neural network (DenseNet) [16]. The process of multiple rooms recognition via DenseNet is illustrated in Fig. 6. First, about 3000 subregions are extracted from the input image utilizing the selective search [17]. Then for each extracted subregion, we use a trained DenseNet to recognize the room logo and classify its label. Because extracted subregions may have overlap, many different subregions can be classified into the same room. We further remove redundant subregions of a room by calculating and comparing both the intersection-over-union (IoU) overlap and classification scoring. In addition, to conduct the room recognition by the DenseNet, a training data set is built in advance. The training images are collected from two parts, which contains 2,000 images about 100 categories of facilities, alphabet and numbers. The first part images (300 out of 2000 samples) are collected via photographing the facilities from various viewpoints in the real situation (different from the test buildings). The second part images (1700 out of

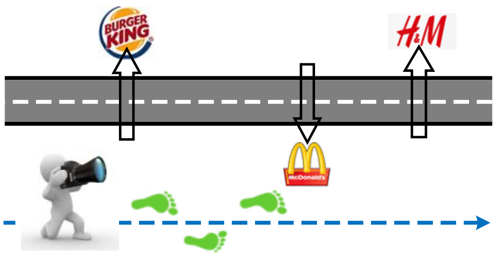


Fig. 9: Facility Photographing Manner (FPM).

Fig. 10: Facilities are recognized from the photos.

2000 samples) are downloaded from the internet to enable the generality of the images. Thus, given a set of images of the target objects (e.g., the logos of various stores, alphabet and numbers) from different viewpoints, our method can recognize: 1) the target rooms contained in each image; and 2) the spatial relations between these rooms. Fig. 8 shows that two rooms are detected in the image captured at the corner.

Once the room is detected, we obtain the corresponding name which is associated with the room. The closer the detected room is to the center of image, the farther it is to the photographer's location (the corner). After recognizing the rooms in each direction of the corner, we define the *fingerprint* of corner C_i as

$$F_i = [f_1^i, f_2^i, \cdots, f_n^i]^T = \begin{bmatrix} N_{11}^i & N_{12}^i & \cdots & N_{1m}^i \\ N_{21}^i & N_{22}^i & \cdots & N_{2m}^i \\ \vdots & \vdots & \ddots & \vdots \\ N_{n1}^i & N_{n2}^i & \cdots & N_{nm}^i \end{bmatrix} \quad (1)$$

where f_n^i is sequence of room names detected in the direction n of corner C_i, and N_{nm}^i is the name of the mth room along the direction n. Thus, the generated fingerprint of corner can be viewed as a unique identification to distinguish various corners.

Facility Photographing Manner (FPM). In PlanSketcher, we design Facility Photographing Manner (FPM) for the user to gather the inertial data and images of facilities. The facilities involve various kinds of rooms in the buildings, such as stores and restrooms etc. As illustrated in Fig. 9, when the user is traversing a facility (such as a store), he stops and turns to face the facility to take a photo which contains the facility logo. At the same time, the gyroscope and accelerometer record the angular velocity and acceleration measurements.

Recognizing facilities from the FPM and image. Similar to the hallway corner, the facility is detected through the proposed FPM. Leveraging the object detection technique, PlanSketcher obtains the names and categories of facilities from the captured photos. During the object detection process, the training examples

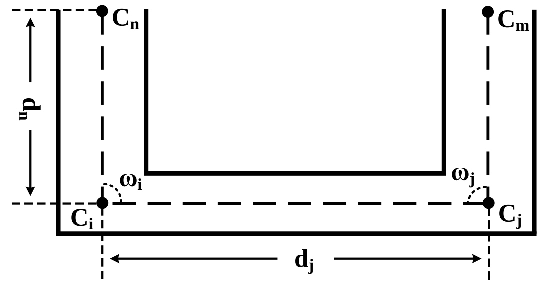


Fig. 11: The model of hallway and corner.

are given as the triples $< name, category, images >$, which indicate the correspondences between the names and categories of target facilities and their images. The training images are gathered via photographing the facilities from different viewpoints or downloading from the Internet. Based on the observations of multiple indoor environments, we distribute the facilities into seven classes {fashion}, {supermarket}, {accessories}, {personal care}, {electronics}, {food} and {others}. Thus, when a facility is recognized from the image, the corresponding name and category are obtained. Fig. 10 shows that facilities are recognized from the captured photos.

4 Hallway Construction

After recognizing various landmarks, we construct the topology of hallways based on the inertial sensing data, depth data and images. Especially, we identify two scenarios in hallway construction: 1) constructing hallways which are traversed by the user with high fine-granularity, and 2) constructing hallways which the user has not traversed. We illustrate these two scenarios as follows.

4.1 Constructing Traversed Hallways

It is not trivial to construct the hallway topology with high fine-granularity. Although the hallway topology could be derived from the user's trajectories tracked by the inertial sensors [18], the accuracy of construction is relatively coarse. Thus, for hallways traversed by a user, we propose a novel method to construct the hallway utilizing the inertial sensing data, depth data and images.

To formally define this problem, we represent a hallway segment and a corner as a line segment and a cross point, and denote them as $\overline{C_iC_j}$ and C, respectively (shown in Fig. 11). Thus k corners have a set of cross points $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$. $\mathbf{D} \in \mathbb{R}$ and $\boldsymbol{\omega} \in [-\pi, \pi)$ are the set of the distances of hallways and angles between them. To construct the fine-grained hallway topology, our system calculates the configuration $< \mathbf{D}, \boldsymbol{\omega} >$ of hallways, where $\mathbf{D} = (d_1, d_2, \ldots, d_n)$, $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_m)$.

Measuring and calibrating the distance D: In this part, we propose a new method to calculate the distance of a hallway through the accelerometer and the depth camera. When a user traverses the hallway, he needs to stop and capture images of rooms distributed along the hallway by Facility Photographing Manner (FPM). This gives us a chance to transfer the problem of calculating the distance d of hallway to the problem of calculating the distance p between two photographing positions as follows:

$$d = \sum_{i=1}^{n} p_i, \quad (2)$$

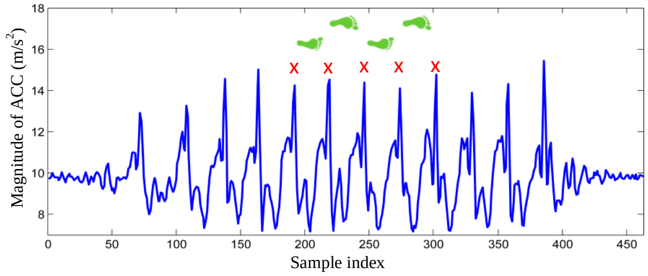
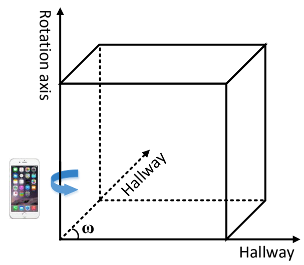
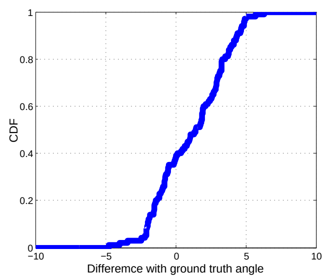


Fig. 12: Measuring the distance through counting the steps from the acceleration data.



(a) Angle measurement.



(b) Measurement error for 90° corner.

Fig. 13: Measuring the angle between two crossed hallways.

where p_i denotes the distance of two adjacent photographing positions.

Depth data: With the rise of Augmented Reality (AR) technique and the improvement of smartphone computing power, many new sensors have been equipped, such as time-of-flight (TOF) depth camera. The TOF depth camera can obtain the distance between the camera and the subject for each point in the image through measuring the time-of-flight of a laser light. Thus, similar with the normal RGB camera, a 3D depth image can be captured by the TOF depth camera with the depth of each point known. Because of the internal property of TOF depth camera, the measured distance can achieve a high accuracy in centimetre level.

We measure the distance between two photographing positions through two manners: accelerometer and depth camera. First, between two adjacent photographing positions, the steps are detected and counted from the acceleration data [14], [19], as shown in Fig. 12. The distance is calculated by multiplying the step count with the step size which could be estimated from the user's weight and height. Second, when the user captures an image in the corner or turns to face the hallway after photographing a room, a 3D depth image of the hallway will be captured by the depth camera. Through the position of the recognized room in the RGB image, the room position can be projected into the depth image. Thus, after projecting the 3D depth image to the horizontal plane, we can obtain the distance between two adjacent photographing positions from the depth image.

The distance between two photographing positions is calibrated by fusing two measurement results. In our system, an effective particle filter [20] is designed to calibrate the measured distances. After activating the data collection, PlanSketcher generates particles spread from the first photographing position. Each particle represents a possible distance between two photographing positions and is updated according to the detected user's steps from the collected acceleration data. To compensate for the differences in user's step length, a zero mean Gaussian noise is added to each particle's step length. The probability density function for particle's step length is:

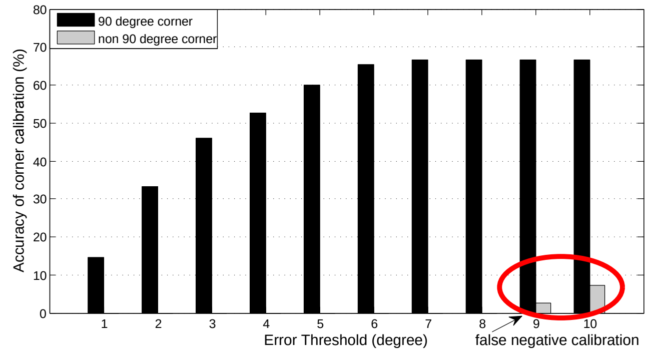


Fig. 14: The tradeoff between error threshold and accuracy of 90° corner calibration.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\|x-\mu\|^2}{2\sigma^2}}, \tag{3}$$

where μ is measured distance from acceleration data, σ is the standard deviation. Then, each particle is weighted by the measurement of depth data. PlanSketcher gives greater weights to the more likely particles and the less likely particles are gradually filtered out. The centroid of weighted particles approximates the actual distance between two photographing positions. The weighted particles are resampled after weight normalization. Such calibration process is repeated until the user finishes data collection.

Particle weighting: Next, we describe how PlanSketcher weights each particle according to the observed depth measurement. With the distance measured from projected depth data, a Euclidian difference Δd_i can be obtained for each particle i. When particles need to be updated, their weights are set according to their corresponding Euclidian difference via a Gaussian kernel. In particular, for the i-th particle, its weight is set as follows:

$$weight_i = e^{-\frac{\Delta d_i}{k}}, \tag{4}$$

where Δd_i is the Euclidian difference and k is a tunable parameter.

Moreover, if the hallway is traversed repetitively, we use the average of multiple estimations to be the distance.

Measuring and calibrating the angle ω: To acquire the spatial relation of hallways, the angle ω between two crossed hallways is measured and calibrated through the photographing manner and depth data.

First, the angle ω is measured by integrating angular velocities captured by the gyroscope. As the prescribed Corner Photographing Manner (CPM), the user holds the smartphone perpendicularly to the earth surface to photograph in the direction of a hallway, and then turns to another direction. So the rotation angle represents the angle between the two hallways as illustrated in Fig. 13 (a).

Second, after measuring the angle, out system calibrates the measurement value to the truth value for the 90° corner. The idea comes from the observation that most of the corners in the building are 90° and there is a distinct gap between 90°

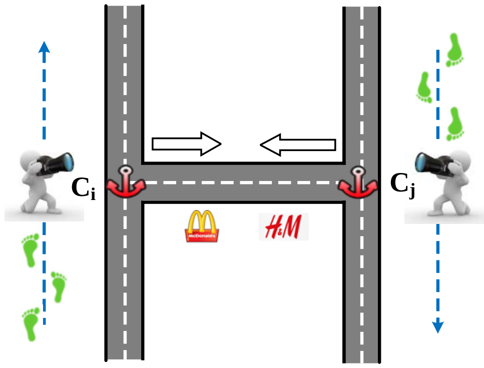


Fig. 15: Constructing a non-traversed hallway through image recognition.

Fig. 16: Two reverse sequences of room names are detected from the photos captured at the corner A and B. (Photos have been zoomed up and clipped to be more clear.)

corner and non $90°$ corner. Fig. 13 (b) shows the cumulative distribution function (CDF) of angular measurement error of $90°$ corner. The measurement error is minor, which is benefited from the prescribed manner, with a fluctuation range from $-4.75°$ to $6.3°$. We define the absolute error $E_i \in [0, 2\pi)$ of the corner C_i as:

$$E_i = |M_i - T_i| \tag{5}$$

where M_i and T_i denote the measurement value and truth value, respectively. In PlanSketcher, we set a threshold for E to calibrate the angle measurement of $90°$ corner. The threshold is equal to $5°$ in our system, indicating that the measurements between $(90° - 5°)$ and $(90° + 5°)$ will be calibrated to $90°$. Although relatively loose threshold improves the measurement accuracy of $90°$ corner, it will introduce false negative calibration. We measure about 300 corners in different buildings and calibrate them with various thresholds. The accuracy of calibration is shown in Fig. 14. We observe that more $90°$ corners can be calibrated correctly with larger threshold. However, when the threshold is more than $9°$, non-$90°$ corner will be wrongly calibrated to $90°$ corner. Thus, in our system, we set the threshold as $5°$ to balance the quantity and accuracy of $90°$ corner calibration.

Moreover, considering some corners are not $90°$, we calibrate the measured angle by the 3D depth image. After projecting the 3D depth image to the horizontal plane, we use the line detection algorithm [21] to extract the outer contour of the hallway. By detecting the cross point of the outer contour, the angle of the corner can be obtained from the 3D depth image. Similar with the hallway distance calibration, we also calibrate the corner angle based on the particle filter. The weight of each particle is decided through the measured angle from the 3D depth data.

Algorithm 1: Non-traversed Hallway Construction

Input:
adjacent corner pair (C_i, C_j);
corresponding images set pair $(\mathbf{I}_i, \mathbf{I}_j)$;
Output:
hallway indicator vector Λ;

for *all* (C_i, C_j) **do**
  initialize Λ with 0;
  calculate fingerprint pair (F_i, F_j);
  for *all* (p, q) **do**
    if f_p^i == *reverse* (f_q^j) **then**
      $\Lambda \leftarrow 1$;
      break;
    end
  end
end

4.2 Constructing Non-traversed Hallways

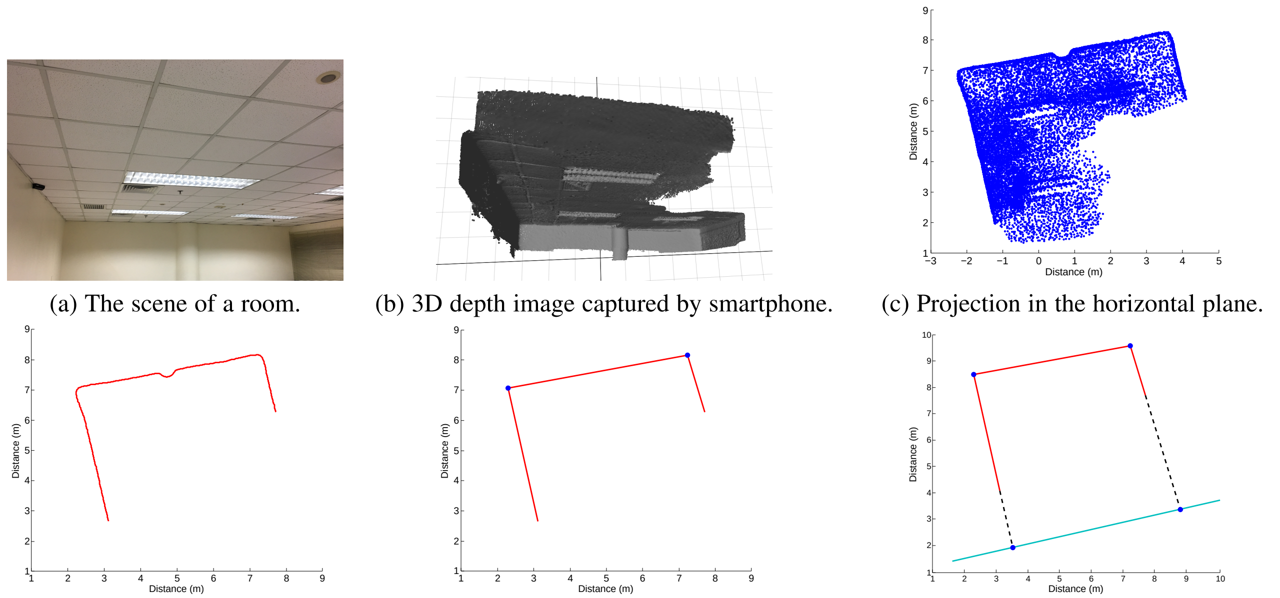
When the hallways are interlaced in the building, it is effort-intensive and time-consuming for the user to traverse every hallway. If the user traverses the hallway without repetition, a fraction of hallways could be constructed with the above method. Thus, the Traversed Hallways Construction method is necessary but not sufficient to construct the integrated hallway topology.

To construct the hallways which the user has not traversed, we design a novel Non-traversed Hallways Construction method by comparing *the sequences f of room names* in various fingerprints F of corners. The method is motivated by the observation that the sequences of room names (f_i and f_j) are reserve if they are detected from the photos captured at the two corners of the hallway (C_i and C_j). As illustrated in Fig. 15, although the user does not traverse the hallway $\overline{C_iC_j}$, the photos are captured in each direction of the two corners as the prescribed Corner Photographing Manner (CPM). And two reverse sequences of room names (such as $[LACOSTE, GEOX]$ and $[GEOX, LACOSTE]$) could be detected from the photos, as shown in Fig. 16. (Photos have been zoomed up and clipped to be more clear.)

The algorithm inputs are the adjacent corner pair which are selected from the previous detected corners and their corresponding images set pair. We assume that for each adjacent corner pair there is an indicator $\Lambda \in \{0, 1\}$ telling whether there exists a hallway between them. Suppose t corner pairs are selected from the previous detected corners. First, for each corner pair (C_i, C_j), the fingerprints of two corners F_i and F_j are calculated through object detection from the images as the above definition Equation (1). Second, we compare every row of room names f_p^i and f_q^j in the fingerprints F_i and F_j. If two reverse sequences are observed, the system infers that the two corners belong to the same hallway and the hallway is constructed to connect them as a linear segment. The whole process is described in Algorithm 1. Thus, PlanSketcher provides an opportunity for the user to construct the non-traversed hallway utilizing the images captured in the corner.

5 Labelled Room Construction

After obtaining the topology of hallways, we construct a relatively complete indoor floor plan with detailed shape and room information by: 1) room construction, and 2) facility labeling. We illustrate these two steps as follows.



(a) The scene of a room.

(b) 3D depth image captured by smartphone.

(c) Projection in the horizontal plane.

(d) Extracting the outer contour of the room.

(e) Inferring line segments and major geometric vertices of the room.

(f) Constructing completely room shape.

Fig. 17: Illustration of room shape construction.

5.1 Room Construction

The shape of room is a crucial component in indoor floor plan. Most existing works [8], [10], [12] on room shape construction need users to traverse in the room to collect various data (e.g, inertial sensing data, wireless data or images). However, these approaches are labor-intensive and the accuracy of constructed room shape is still limited. Thus, we propose a novel method to construct room shape from depth data with high accuracy but without traversing in the room.

In our system, we leverage the TOF depth camera equipped in smartphone to construct the room shape. After the user takes a photo of a room using FPM, he the captures a 3D depth image of the scene in the room. Considering the problem of obstacles in the room, we ask the user to capture a depth image including the ceiling and the connecting line of wall segments. The insight of this manner is that the ceiling can effectively represent the room shape with less obstacles. Fig. 17 (a) shows the scene of a room and Fig. 17 (b) is the 3D depth image captured by smartphone. When the user captures a 3D depth image, the pose of smartphone can be derived from sensor readings, i.e. yaw, pitch and roll. Then we obtain a rough room shape through projecting the 3D depth image to the horizontal plane, as shown in Fig. 17 (c).

To construct completely room shape, we model the room by major geometric vertices and edges. First, we use the line detection algorithm [21] to extract the outer contour of the rough room shape (shown in Fig. 17 (d)) and infer the line segments of the room (shown in Fig. 17 (e)). Next, we extend the line segments to find the major geometric vertices in the room, as the two blue points shown in Fig. 17 (e). Then we transfer the room shape constructed in the local coordinate system to the global coordinate system of the indoor floor plan, according to the position of photographing. Moreover, to derive the connecting point between the room and the hallway, we also extend the line segments in two sides to produce the two cross points. Fig. 17 (f) gives an example, where the blue line is the hallway. Especially, for some irregular room, simply capturing one depth image may not cover all boundaries of walls in the room. The user can make a scanning of boundaries of the ceiling to construct the room.

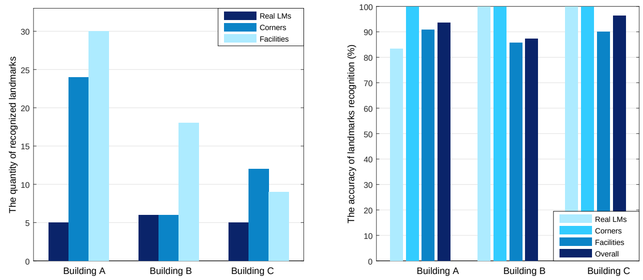
Different from previous works, the room construction method we proposed do not need user to traverse in the room and can achieve a high accuracy. This non-traversed method can save much labor, time and energy consumption in the smartphone. And because of the internal fine-grained property of laser measurement in the TOF depth camera, the accuracy of constructed room shape is improved.

5.2 Facility Labeling

In this subsection, we calculate the entrance positions and orientations of facilities and label them in the constructed hallway plans and rooms. The names and categories of facilities are extremely valuable information for user's reference. It is difficult for users to find their destinations and know the surroundings with merely the topology of hallways and rooms. Thus the complete indoor floor plans should provide the names, categories and positions of facilities.

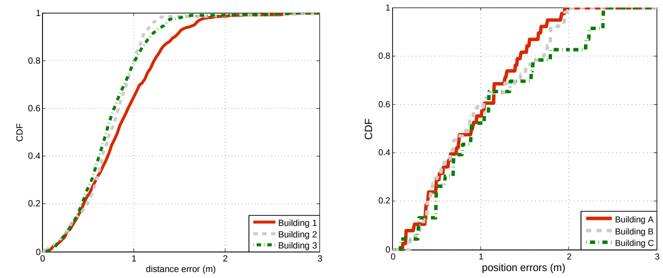
The positions of elevators, escalators, stairs (viewed as the real landmarks) and restrooms are directly derived from the recorded inertial sensing data and the positions of photographing. The positions and orientations of rooms are estimated from the Facility Photographing Manner (FPM).

The proposed FPM provides a cue to estimate the positions of entrances $\mathbf{E} = (E_1, E_2, \ldots, E_n)$. From our practical observations, we find that the logos are usually placed on the top of or beside the entrances. Because the user takes photos in front of rooms, we set the photograph's locations as the positions of entrances, which are derived from the acceleration data. The orientations of facilities are inferred from rotation angles when the user photographs. The rotation angle is calibrated as the hallway angle measurement through the threshold-based approach. After calculating the entrance positions and orientations of facilities, PlanSketcher labels the facility icons or names in the hallway plans. The spatial relations of facilities are ensured from the FPM.



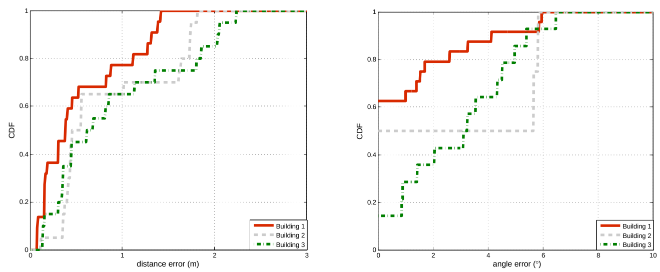
(a) The quantity of recognized landmarks. (b) The accuracy of landmarks recognition.

Fig. 18: The Performance of Landmark Recognition.



(a) Distance errors of constructed rooms. (b) Position errors of facility labeling.

Fig. 20: The Performance of Labelled room Construction.



(a) Distance errors of constructed hallways. (b) Angle errors of constructed hallways.

Fig. 19: The Performance of Hallway Construction.

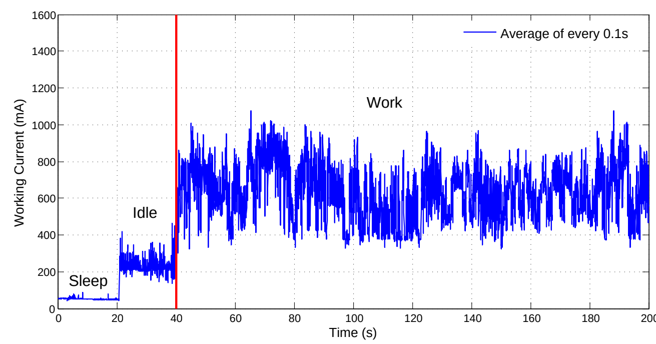


Fig. 21: Energy Consumption.

Thus, it is more accurate to locate facilities than the conventional trajectory-based approaches.

6 Implementations and Evaluation

In this section, we present the evaluation of key functional components of PlanSketcher. We evaluate PlanSketcher in three representative indoor environments for a better understanding of the effectiveness and limitation of system.

6.1 Experimental Methodology

We implement PlanSketcher on the Android platform to collect multiple sensing data and on an Intel core i7 machine with 32GB RAM and NVIDIA TITAN X graphics card to construct the indoor floor plan. Many different types of sensing data are collected through smartphone, including accelerometer data, gyroscope data, magnetometer data, WiFi data, depth data and images. Especially, to recognize room names from images via deep learning, a training data set is built in advance. The training images are collected from two parts, which contains 2,000 images about 100 categories of facilities, alphabet and numbers. The first part images (300 out of 2000 samples) are collected via photographing the facilities from various viewpoints in the real situation (different from the test buildings in the following section). The second part images (1700 out of 2000 samples) are downloaded from the internet to enable the generality of the images. Moreover, to obtain the absolute walking direction in the indoor floor plan construction, we leverage the loss of GPS signal to detect whether a user enters a building. The absolute walking direction can be obtained at the building entrance. After the user enters the building, the walking direction can be derived from inertial sensing data.

6.2 Performance Evaluation

We test PlanSketcher on a variety of Android mobile devices (Lenovo Phab2 Pro, Google Nexus 5, Samsung Galaxy S2 and Samsung Galaxy Note 3). Especially, we use Lenovo Phab2 Pro to gather inertial sensing data, WiFi RSSI values, depth data and images in three large complex indoor environments: one story of a $145m \times 40m$ shopping mall (Building A), one story of a $100m \times 40m$ university building (Building B) and one story of a $40m \times 40m$ exhibition center (Building C). Because other three mobile devices have not been equipped with depth camera, we test their energy consumption of collecting inertial sensing data, WiFi RSSI values and images. The lighting conditions during open hours allow user to capture bright images. The user holds the smartphone in a perpendicular angle to the ground to collect trajectories and images in the buildings. In each environment, we take 38, 16 and 31 photos of different corners in Corner Photographing Manner and collect 33, 27 and 17 photos of different facilities in Facility Photographing Manner. We also gather 12, 6 and 8 user trajectories along the hallways in each building.

Performance of Landmark Recognition. We evaluate the quantity of recognized landmarks and the accuracy of landmarks recognition. Fig. 18 (a) shows the quantity of landmarks recognized from the sensing data and images in each building. For the building A, the composition of the landmarks is: 5 real landmarks, 24 corners and 30 facilities. For the building B, the composition is: 6 real landmarks, 6 corners and 17 facilities. For the building C, the composition is: 5 real landmarks, 12 corners and 9 facilities. Because PlanSketcher recognizes various landmarks, the well-known cumulative error is limited to a low level, which improves the accuracy of whole system. Fig. 18 (b) shows the accuracy of landmarks recognition for all categories of landmarks. By the proposed integrated features, most of the real landmarks are recognized (only 1 missed in building A). The reason for the

TABLE 1: The Energy Consumption on Lenovo Phab2 Pro

Mode	Period	Power	Current	Battery Life
Sleep	$0s \sim 20s$	$190.90mW$	$51.18mA$	$79.13h$
Idle	$20s \sim 40s$	$884.12mW$	$237.03mA$	$17.09h$
Work	$40s \sim 200s$	$2390.30mW$	$640.83mA$	$6.32h$

TABLE 2: Battery Life Measurements in Different Models of Smartphones

Energy	Phab2 Pro	Nexus 5	Galaxy S2	Note 3
Battery Capacity	$4050mAh$	$2300mAh$	$1800mAh$	$3200mAh$
Battery Life (Work)	$6.32h$	$4.47h$	$3.11h$	$4.91h$

TABLE 3: The Performance Comparison

Category	Jigsaw	Shopprofiler	PlanSketcher
Position error	$2m \sim 3m$	$2m \sim 5m$	$1m \sim 2.5m$
Orientation error	$6° \sim 9°$	$8° \sim 15°$	$4° \sim 6°$
Facility quantity	None	$9 \sim 25$	$20 \sim 40$
Recognition rate	None	$30\% \sim 60\%$	$85\% \sim 95\%$
Energy Consumption	$843mAh \sim 1278mAh$	$411mAh \sim 634mAh$	$224mAh \sim 358mAh$

undetected real landmark (an escalator) is the low magnitude of RSSI (less than -80dBm), which cannot help the acceleration readings to present a significant feature. All corners are detected and the recognition rate of facilities is more than 85%. Although some landmarks are missed, the recognition still achieves a high accuracy with the overall rate: 93.65%, 85.29% and 96.30% in three buildings.

Performance of Hallway Construction. Precision of the constructed hallway is a critical criteria of PlanSketcher. We use the distance errors and angle errors of conducted hallways (i.e., the difference between the calculation values and the ground truth measurements) to evaluate the performance. As shown in Fig. 19 (a), PlanSketcher generates precise distances of hallways. The constructed hallways achieve average accuracy of $0.51m$, $0.69m$ and $0.83m$ and 90th percentile accuracy of $1.32m$, $1.72m$ and $2.02m$ in each building. Because abundant landmarks are recognized in the building and the accurate depth data, the accumulation error of distance is limited to a low level. As shown in Fig. 19 (b), PlanSketcher produces accurate angles of hallways. PlanSketcher yields the angles of hallways, with average accuracy of $1.15°$, $2.87°$ and $2.92°$ and 90th percentile accuracy of $4.14°$, $5.84°$ and $5.42°$ in each building. The high accuracy is benefitted from the stable performance of Corner Photographing Manner (CPM) and hallway angle calibration. Most angle measurements of corners have been calibrated to the truth values.

Performance of Labelled Room Construction. We evaluate the performance of labelled room construction using the distance errors of constructed rooms and position errors of facility labeling. Fig. 20 (a) shows the distance errors of constructed room edges. The constructed room edges achieve average accuracy of $0.84m$, $0.61m$ and $0.72m$ and 90th percentile accuracy of $1.37m$, $1.07m$ and $1.21m$ in each building. Because of the internal property of high accuracy in TOF depth camera, the distance error of edge is limited to a low level. Fig. 20 (b) shows the position errors of room entrances where are used to label facilities. The average errors of room entrances are $0.94m$, $0.98m$, $1.1m$ and 90th percentile accuracy of $1.68m$, $1.79m$, $2.23m$ in each building. The high accuracy of facility positions is achieved, because abundant recognized landmarks in the buildings help to calibrate the measured distances.

Energy Consumption. PlanSketcher employs inertial sensors, WiFi, camera as well as depth camera to collect various data for indoor floor plans construction. Thus, given the energy bottleneck of smartphones, the energy consumption issues should be taken into consideration and discussion for practical use. We measure the energy consumption on various models of smartphones utilizing the Monsoon power monitor [22]. The power monitor directly supplies power to the smartphone and accurately tracks the current, voltage and power. To precisely measure the energy consumption of PlanSketcher, all background services and applications are turned off. The WiFi module is turned on and the screen brightness is set to auto-adjustment mode. All the sensor modules and WiFi module are sampled and processed in real time. As the smartphone has to be wired with the power monitor for measurement, we keep the smartphone stationary or moving in limited space. We synthetically trigger image capture in PlanSketcher every 5s to 20s.

Fig. 21 shows the working currents measured on the Lenovo Phab2 Pro with a 4050mAh battery in different working modes as an example. The current is sampled at 5KHz utilizing the power monitor and averaged over every window of 0.1s. During the evaluation, the smartphone is in sleep mode during the period from 0s to 20s. We wake up and unlock the smartphone during the period from 20s to 40s. And then we launch PlanSketcher at around 40s and start to collect various sensing data and images. The data collection is finished at 200s. We repeat the experiments 10 times and characterize the power draws in different modes in Table 1. The work mode of PlanSketcher draws power at around 640.83mA and the expected battery life is 6.32h for the Lenovo Phab2 Pro.

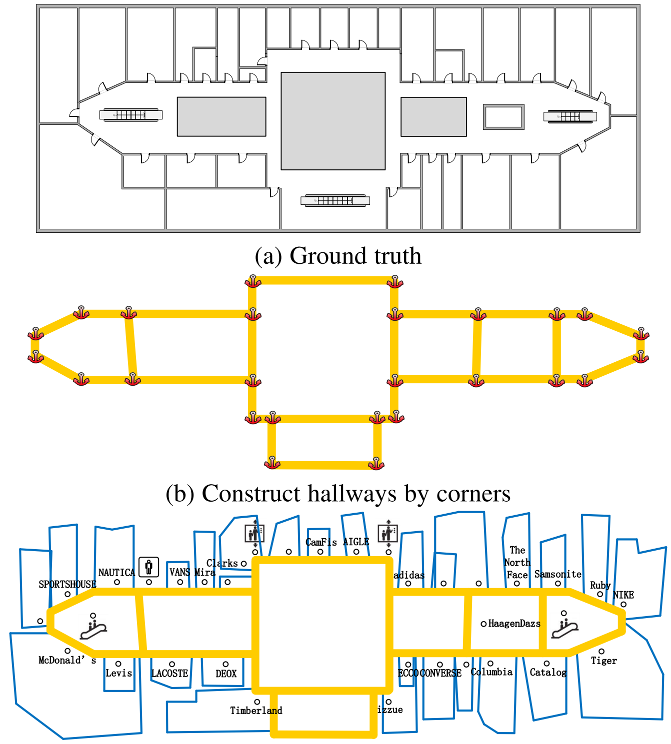
We also measure the expected battery life of the Google Nexus 5 with a 2300mAh battery, the Samsung Galaxy S2 with a 1800mAh battery and Samsung Galaxy Note 3 with a 3200mAh battery. Because these mobile devices have not been equipped with depth camera, we test their energy consumption of collecting inertial sensing datap, WiFi RSSI values and images. The measurement results are presented in Table 2. For the Google Nexus 5, it can continuously work for about 6.32h. The expected battery life of Google Nexus 5 is 4.47h in work mode. Powered by a small battery, the expected life of Samsung Galaxy S2 is around 3.11h in work mode. The expected battery life of Samsung Galaxy Note 3 is 4.91h in work mode. The expected battery life time is longer for the Lenovo Phab2 Pro compared with the others, mainly because of its larger battery.

The current version of PlanSketcher can be further optimized for energy efficiency. PlanSketcher may reduce the image capture quality to a discernible level to save energy and benefit from the energy efficient mobile vision techniques [23]. When the user traverses in the indoor environment, PlanSketcher may dynamically adjust WiFi scanning rate to reduce energy consumption. PlanSketcher can also benefit from energy efficient coprocessor architectures for sensor fusion as well [24].

Performance Comparison. To further evaluate our proposed system, we implement two systems (Jigsaw [10] and Shopprofiler [12]), and compare the construction performance with them.

Jigsaw constructs indoor hallway and room topologies from the inertial sensing data and images. Shopprofiler utilizes the inertial sensing data, acoustic data and WiFi access information to construct indoor floor plans and recognize shops in the buildings. Both of the systems use the crowdsourcing approach to collect data. In the experiment, we recruited 10 volunteers to collect total 100 user traces (including acceleration and gyroscope data) and 250 photos in each building to implement Jigsaw. For Shopprofiler, we also recruited 10 volunteers to collect total 100 user traces (including acceleration data, gyroscope data, acoustic data and WiFi access information) in each building. As shown in Table 3, PlanSketcher outperforms the other two schemes. PlanSketcher achieves a higher accuracy ($1th$ and $2th$ rows in Table 3) of indoor floor plans (including hallways and rooms) than the others, because much landmarks are recognized and most of the measured angles are calibrated to the truth values. PlanSketcher can recognize more facilities ($3th$ and $4th$ rows in Table 3) based on the superior visual object detection technique instead of inaccurate WiFi information. In addition, we test the energy consumption in smartphone of completing the input data collection in each building for each system. The battery of smartphone is recorded when data collection is started and finished. Our system consumes less energy in smartphone ($5th$ row in Table 3). This is because: 1) our system is a non-crowdsourcing-based approach which spends less time collecting less amount of data, and 2) the user is unnecessary to traverse all hallways and rooms. We notice that Shopprofiler fails to recognize most of the facilities in the buildings. This is because a fraction of stores or rooms provide their own WiFi access while the others are provided by the buildings which results in the same WiFi SSID. The above results demonstrate that PlanSketcher can construct the facility-labelled and highly fine-grained indoor floor plans with little energy consumption in smartphone.

Floor Plan Performance. Fig. 22 shows the ground truth and the process of labelled indoor floor plan construction in the shopping mall. The shadow areas represent some inaccessible areas. In Fig. 22 (b), the anchor icons highlight all detected corners and the lines show constructed hallways. When two corners are detected belonging to the same hallway, they are connected with a straight line to form a hallway. In Fig. 22 (c), the blue lines indicate the constructed room shape. Especially, many room shapes are different from the ground truth, because these shop owners restructure the room space to generate extra functional zones (such as storeroom). Facilities are recognized and labelled in the floor plan, which involve elevators, escalators, restrooms and stores etc. The circles indicate the positions of facility entries. Because the training samples may not be collected perfectly sufficient, a small quantity of facilities (1 escalator and 3 stores) are not recognized and labelled. Compared with the ground truth, the spatial relations of hallways and facilities are all correct in the constructed floor plans. Moreover, Fig. 23 and Fig. 24 show the constructed indoor floor plans in the university building and the exhibition center, respectively. Compared with the room shapes constructed in shopping mall, the room shapes are more consistent with the architectural drawings in these two buildings. This is because the room space is not restructured in these buildings, which demonstrates the effectiveness of our proposed methods. The spatial relations of hallways are correct and more than 85% facilities are recognized and labelled successfully.

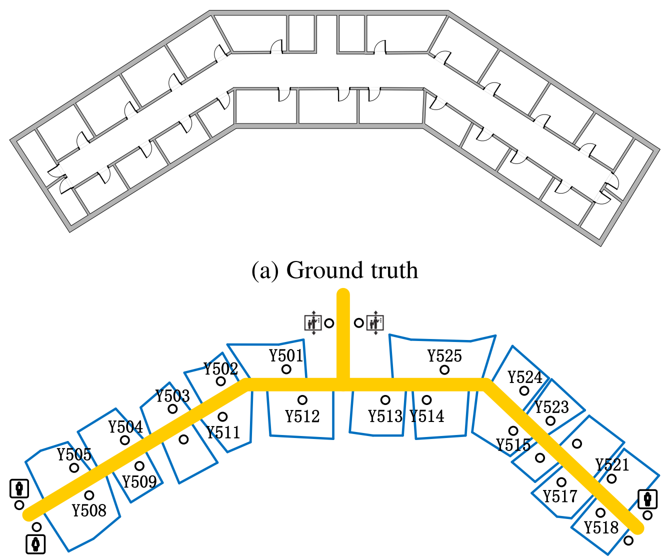


(a) Ground truth

(b) Construct hallways by corners

(c) Construct indoor floor plan with labels

Fig. 22: The Ground Truth and Constructed Indoor Floor Plan with Labels in the Shopping Mall.



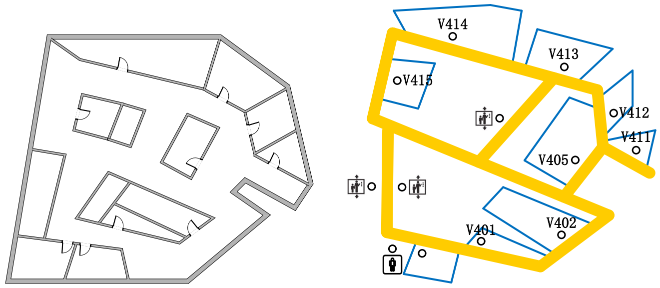
(a) Ground truth

(b) Construct indoor floor plan with labels

Fig. 23: The Ground Truth and Constructed Indoor Floor Plan with Labels in the University Building.

7 RELATED WORK

Indoor floor plan construction. Indoor floor plan construction is an extensively studied field in mobile computing. Most of the existing approaches focus primarily on inertial sensing data aggregations. CrowdInside [8] is a crowdsourcing-based system for automatically constructing indoor floor plans. It leverages various inertial sensing data collected from the smartphones to generate user motion traces. Some landmarks such as elevators, stairs and locations with GPS reception are recognized to calibrate



(a) Ground truth (b) Construct indoor floor plan with labels

Fig. 24: The Ground Truth and Constructed Indoor Floor Plan with Labels in the Exhibition Center.

users' trajectories and accumulative errors. The proposed algorithm is highly dependent on the accuracy of crowdsourced motion traces. Jiang *et al.* [5] leverage WiFi fingerprints and user motion information to obtain the room sizes and hallway orientations to automatically construct indoor floor plans. Walkie-Markie [9] uses the locations at which the trend of WiFi signal strength changes as the landmarks to construct indoor floor plans. However, their systems fully rely on the availability of WiFi fingerprints. Jigsaw [10] combines both image and sensing data to construct the indoor floor plan, which achieves a better performance. However, it only uses images to infer the wall segments of the room entrance and still uses aggregated user motion trace and camera position to infer the shape of room. We cannot assume all edges and corners of the room could be covered with user traces as it may be inaccessible to users (e.g. blocked by desk). Most of the existing works are based on the crowdsourcing method, which need to collect abundant data. Although crowdsourcing-based scheme has its own advantages and is promising for practical applications, it won't be the only way for constructing indoor floor plan. The proposed PlanSketcher avoids the well-known shortcomings caused by crowdsourcing, such as the slow-start problem, privacy issue, high overhead, device heterogeneity, etc. Therefore, the design in this paper presents a reasonable alternative to the existing works.

Mobile big data and crowdsensing. With the development of mobile communication and Internet of Things, analyzing mobile data collected through crowdsensing [25] provides a new research field. Considering the vehicular crowdsensing scenario, the Nash equilibrium of the static vehicular crowdsensing game is derived for various sensing tasks in [26]. A new payment strategy and sensing strategy is proposed for the dynamic vehicular crowdsensing game based on some deep learning techniques. Cheng *et al.* [27] provide a comprehensive survey on the features, sources and applications of mobile big data. Some challenges and opportunities for research and development in this field are discussed with an emphasis on the user modeling, infrastructure supporting, data management, and knowledge discovery aspects. In addition, with a huge amount of data collected in mobile networks, recent research on mobile big data mining show great potential for various purposes including traffic management improvement, providing personal and contextual service, and city dynamics monitoring. The authors in [28] analyze many research opportunities and challenges in the application of mobile big data. Various scenarios are discussed including communication and networking infrastructure, data security and privacy, as well as data mining and knowledge discovery.

8 Conclusion

In this paper, we utilize machine learning techniques to propose PlanSketcher, a system that enables a user to construct indoor floor plans by himself. Compared with previous works, PlanSketcher can construct fine-grained and facility-labelled indoor floor plan with less energy consumption. To realize this system, various sensing data are collected from smartphone and novel landmarks recognition approaches are presented. Then novel hallways construction algorithms are proposed to construct traverse-independent hallway topologies. With the object detection technique, PlanSketcher also constructs the room shape and labels recognized facilities in their corresponding positions. We implement PlanSketcher and conduct abundant experiments. The evaluation results illustrate that PlanSketcher outperforms the state-of-the-art methods by showing smaller position and orientation error, more recognized facilities and less energy consumption.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China under Grant No. 61772446 and No. 61472365.

References

[1] A. Goswami, L. E. Ortiz, and S. R. Das, "Wigem: a learning-based approach for indoor localization," in *Proc. of ACM CoNEXT*, 2011, pp. 1–12.
[2] C. Wu, Z. Yang, C. Xiao, C. Yang, Y. Liu, and M. Liu, "Static power of mobile devices: Self-updating radio maps for wireless indoor localization," in *Proc. of IEEE INFOCOM*, 2015, pp. 2497–2505.
[3] Z. Peng, S. Gao, B. Xiao, S. Guo, and Y. Yang, "Crowdgis: Updating digital maps via mobile crowdsensing," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 369–380, 2018.
[4] M. Li, P. Zhou, Y. Zheng, Z. Li, and G. Shen, "Iodetector: A generic service for indoor/outdoor detection," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 2, pp. 28:1–28:29, 2015.
[5] Y. Jiang, Y. Xiang, X. Pan, K. Li, Q. Lv, R. P. Dick, L. Shang, and M. Hannigan, "Hallway based automatic indoor floorplan construction using room fingerprints," in *Proc. of ACM Ubicomp*, 2013, pp. 315–324.
[6] D. Philipp, P. Baier, C. Dibak, F. Durr, K. Rothermel, S. Becker, M. Peter, and D. Fritsch, "Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data," in *Proc. of IEEE Percom*, 2014, pp. 139–147.
[7] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.
[8] M. Alzantot and M. Youssef, "Crowdinside: automatic construction of indoor floorplans," in *Proc. of ACM SIGSPATIAL*, 2012, pp. 99–108.
[9] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: indoor pathway mapping made easy," in *Proc. of USENIX NSDI*, 2013, pp. 85–98.
[10] 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*, 2014, pp. 249–260.
[11] S. Chen, M. Li, K. Ren, and C. Qiao, "Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos," in *Proc. of IEEE ICDCS*, 2015, pp. 1–10.
[12] X. Guo, E. C. Chan, C. Liu, K. Wu, S. Liu, and L. M. Ni, "Shopprofiler: Profiling shops with crowdsourcing data," in *Proc. of IEEE INFOCOM*, 2014, pp. 1240–1248.
[13] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-navi: Self-deployable indoor navigation system," in *Proc. of ACM MobiCom*, 2014, pp. 2655–2669.
[14] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *Proc. of ACM MobiSys*, 2012, pp. 197–210.
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of IEEE CVPR*, 2014, pp. 580–587.

[16] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, “Densely connected convolutional networks,” in *Proc. of IEEE CVPR*, 2017, pp. 4700–4708.

[17] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

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

[19] C. Wu, Z. Yang, Y. Xu, Y. Zhao, and Y. Liu, “Human mobility enhances global positioning accuracy for mobile phone localization,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 131–141, 2015.

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

[21] D. C. Lee, M. Hebert, and T. Kanade, “Geometric reasoning for single image structure recovery,” in *Proc. of IEEE CVPR*, 2009, pp. 2136–2143.

[22] “Monsoon power monitor,” http://www.msoon.com/LabEquipment/PowerMonitor/.

[23] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, “Energy characterization and optimization of image sensing toward continuous mobile vision,” in *Proc. of ACM MobiSys*, 2013, pp. 69–82.

[24] H. Shen, A. Balasubramanian, E. Yuan, A. LaMarca, and D. Wetherall, “Improving power effciency using sensor hubs without re-coding mobile apps,” in *Technical Report of University of Washington*, 2014, pp. 1–13.

[25] T. Zhou, B. Xiao, Z. Cai, M. Xu, and X. Liu, “From uncertain photos to certain coverage: a novel photo selection approach to mobile crowdsensing,” in *Proc. of IEEE INFOCOM*, 2018, pp. 1–9.

[26] L. Xiao, T. Chen, C. Xie, H. Dai, and V. Poor, “Mobile crowdsensing games in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1535–1545, 2017.

[27] X. Cheng, L. Fang, L. Yang, and S. Cui, “Mobile big data: the fuel for data-driven wireless,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1489–1516, 2017.

[28] X. Cheng, L. Fang, X. Hong, and L. Yang, “Exploiting mobile big data: Sources, features, and applications,” *IEEE Network*, vol. 31, no. 1, pp. 72–79, 2017.