Multi-Story Indoor Floor Plan Reconstruction via Mobile Crowdsensing

Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Guojie Luo, *Member, IEEE*, Yizhou Wang, Kaigui Bian, *Member, IEEE*, Tao Wang, *Senior Member, IEEE*, and Xiaoming Li, *Senior Member, IEEE*

Abstract—The lack of floor plans is a critical reason behind the current sporadic availability of indoor localization service. Service providers have to go through effort-intensive and time-consuming business negotiations with building operators, or hire dedicated personnel to gather such data. In this paper, we propose Jigsaw, a floor plan reconstruction system that leverages crowdsensed data from mobile users. It extracts the position, size, and orientation information of individual landmark objects from images taken by users. It also obtains the spatial relation between adjacent landmark objects from inertial sensor data, then computes the coordinates and orientations of these objects on an initial floor plan. By combining user mobility traces and locations where images are taken, it produces complete floor plans with hallway connectivity, room sizes, and shapes. It also identifies different types of connection areas (e.g., escalators and stairs) between stories, and employs a refinement algorithm to correct detection errors. Our experiments on three stories of two large shopping malls show that the 90-percentile errors of positions and orientations of landmark objects are about $1 \sim 2m$ and $5 \sim 9^{\circ}$, while the hallway connectivity and connection areas between stories are 100 percent correct.

Index Terms—Multi-story indoor floor plan reconstruction, mobile crowdsensing

1 Introduction

In contrast to the almost ubiquitous coverage outdoors, localization service is at best sporadic indoors. The industry state-of-the-art, Google Indoor Maps [1], covers 10,000 locations worldwide, which is only a small fraction of millions of indoor environments (e.g., airports, train stations, shopping malls, museums, and hospitals) on the Earth. One major obstacle to ubiquitous coverage is the lack of indoor floor plans. 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.

In this paper, we propose *Jigsaw*[2], which leverages crowdsensed data from mobile users to construct the floor plans of complex indoor environments. It avoids the intensive effort and time overhead in the business negotiation process for service providers. They do not need to talk to building owners/operators one by one, or hire dedicated personnel to measure indoor environments inch by inch. Jigsaw opens up the possibility of fast and scalable floor plan reconstruction. The concept of mobile crowdsensing [3] has become increasingly popular. Recent work has used crowdsensed data to localize users [4] and reduce the calibration efforts of WiFi signatures [5], [6]. Among others [7], [8], [9], [10], CrowdInside [11] pioneers the efforts of constructing hallway/room shape and connectivity of floor plans. It uses inertial data to build and combine user mobility traces to derive the approximate shape of accessible areas of floor plans.

Nevertheless, there exists much space for improvements. Inertial data do not give the accurate coordinates and orientations of indoor places of interests (POIs, such as store entrances in shopping malls, henceforth called *landmarks*), which are critical to guide users. Due to error accumulation in dead reckoning, "anchor points" (e.g., entrances/exits of elevators/escalators/stairs and locations with GPS reception) with unique sensing data signatures are needed to correct the drift in mobile traces. But in many large indoor environments, such anchor points can be too sparse to provide sufficient correction. Therefore, both over- and under-estimation of accessible areas can easily happen, e.g., when a trace drifts into walls, or there exist corners users seldom walk into.

Jigsaw combines computer vision and mobile techniques, and uses optimization and probabilistic formulations to build relatively complete and accurate floor plans. We use computer vision techniques to extract geometric features (e.g., widths of store entrances, lengths and orientations of adjoining walls) of individual landmarks from images. We then design several types of data-gathering *micro-tasks*, each a series of actions that users can take to collect data specifically useful for building floor plans. We derive the relative spatial relationship between adjacent landmarks from inertial data of some types of micro-tasks, and compute the optimal coordinates and orientations of landmarks on a common floor plane. Then user mobility traces from another type of micro-task are used to obtain the hallway connectivity,

- *R. Gao, M. Zhao, T. Ye, G. Luo, Y. Wang, K. Bian are with the EECS School, Peking University, Beijing 100871, China. E-mail: {gaoruipeng, zhaomingmin, pkuleonye, gluo, yizhou.wang, bkg}@pku.edu.cn.*
- *F. Ye is with the ECE Department, Stony Brook University, Stony Brook, NY 11794. E-mail: fan.ye@stonybrook.edu.*
- *T. Wang and X. Li are with the EECS School, Peking University, Beijing 100871, China, and the Collaborative Innovation Center of High Performance Computing, NUDT, Changsha, China. E-mail: {wangtao, lxm}@pku.edu.cn.*

Manuscript received 17 Sept. 2015; revised 23 Mar. 2016; accepted 27 Mar. 2016. Date of publication 4 Apr. 2016; date of current version 3 May 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2016.2550040

orientation and room shapes/sizes, using combinatorial optimization and probabilistic occupancy techniques. After reconstruction of each single-story floor plan, inertial data and WiFi/cellular signatures and images are also used to detect different types of connection areas between stories (e.g., stairs, escalators and elevators) to finally produce a multi-story floor plan.

Jigsaw design is based on the realization that computer vision and mobile techniques have complementary strengths. Vision ones can produce accurate geometric information when the area has stable and distinct visual features. They are suitable for landmarks where logos, decorations constitute rich features, and detailed information about their positions/orientations is desired. Mobile techniques give only rough sketches of accessible areas with much lower computing overhead, which is suitable for in-between sections such as textureless or glass walls where much fewer stable features exist, while less detailed information is required. Therefore, we leverage "expensive" vision techniques to obtain more accurate and detailed information about individual landmarks, and use "cheap" inertial data to obtain the placement of landmarks on a large, common floor plane, and derive the less critical hallway and room information at lower fidelity. The optimization and probabilistic formulations give us more solid foundations and better robustness to combat errors from data.

We make the following contributions in this work:

- We identify suitable computer vision techniques and design a *landmark modeling* algorithm that takes their output from landmark images to derive the coordinates of major geometry features (e.g., store entrances and adjoining wall segments) and camera poses in their local coordinate systems.
- We design micro-tasks to measure the spatial relationship between landmarks, and devise a *landmark placement* algorithm that uses a Maximum Likelihood Estimation (MLE) formulation to compute the optimal coordinates, orientations of landmarks on a common floor plane.
- We devise several *augmentation algorithms* that reconstruct wall boundaries using a combinatorial optimization formulation, and obtain hallway connectivity and orientation, room size/shape using probabilistic occupancy maps that are robust to noises in mobile user traces. We also reconstruct three types of connection areas between different floors and general multi-story floor plans.
- We develop a prototype and conduct extensive experiments in three stories of two large complex indoor environments. The results show that the position and orientation errors of landmarks are about $1 \sim 2m$ and $5^\circ \sim 9^\circ$ at 90-percentile, with 100 percent correct isle topology connectivity and connection areas between stories, which demonstrate the effectiveness of our design.

Note that we do not claim novelty in developing new computer vision techniques. Our main contribution is the identification and combination of appropriate vision and mobile techniques in new ways suitable for floor plan construction, and accompanying mathematical formulations and solutions that yield much improved accuracy despite errors and noises from image and inertial data sources.

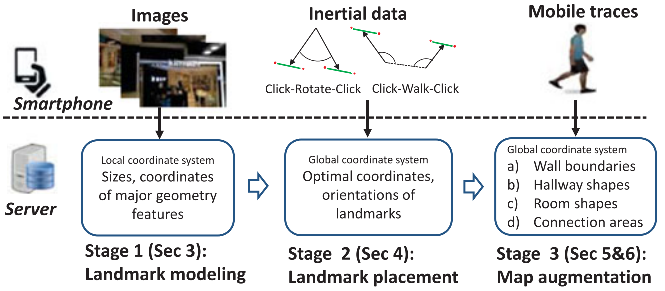


Fig. 1. Jigsaw contains three stages: landmark modeling, landmark placement, and map augmentation. Each stage uses image or inertial data and output from the previous stage.

The rest of the paper is organized as follows: We give an overview (Section 2), then present the design of the landmark modeling, placement and augmentation algorithms (Sections 3, 4, 5, and 6). We also conduct experimental evaluation of our design and demonstrate its effectiveness in Section 7. After a discussion (Section 8) of limitations, comparison to related work (Section 9), we conclude the paper (Section 10).

2 Design Overview

Similar to existing work [7], [8], [9], [10], [11], Jigsaw requires data collected using commodity smartphones from users. We assume that upon proper incentives (e.g., cash rewards [12], [13]), users willing to conduct simple *micro-tasks* can be recruited. They will follow guidelines and gather data in required form and manner: e.g., taking a single photo of a store entrance; taking a photo of one store and then spinning the body to take a photo of another store; walking a certain trajectory on the floor or across stories while taking a photo immediately before/after the walk. Such micro-tasks allow us to gather data for specific elements in floor maps. Given successful industrial precedences [12], [13] where users accomplish tasks in exchange for rewards, and plenty of research [14] on how incentives influence user behavior, we argue such a paradigm is feasible and practical. We leave the exact design of incentive form as future work.

Jigsaw utilizes images, acceleration and gyroscope data. The reconstruction consists of three stages: landmark modeling, placement, and augmentation (Fig. 1). First, two computer vision techniques, Structure from Motion (SfM) [15] and vanishing line detection [16], are used to obtain the sizes and coordinates of major geometry measurements of each landmark in its local coordinate system (Section 3). SfM also produces the location and orientation of the camera for each image, effectively localizing the user who took the picture. Next, two types of micro-tasks, Click-Rotate-Click (CRC) and Click-Walk-Click (CWC), are used to gather gyroscope and acceleration data to measure the distances and orientation differences between landmarks. The measurements are used as constraints in an MLE formulation to compute the most likely coordinates and orientations of landmarks in a global coordinate system (Section 4). Then, a combinatorial optimization is used to connect landmarks' adjoining wall segments into continuous boundaries, and probabilistic occupancy maps are used to obtain hallway connectivity, orientation and room

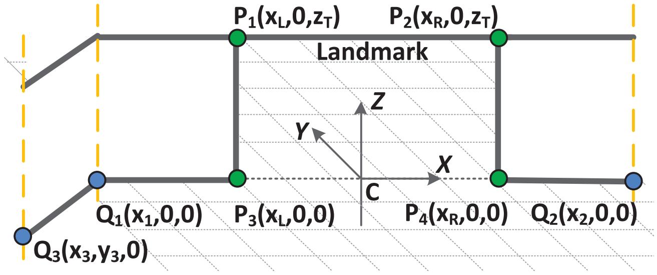


Fig. 2. The model of this exemplary store entrance has four geometric vertices $P_1 \sim P_4$ and three connecting points of wall segments $Q_1 \sim Q_3$ in its local coordinate system.

sizes/shapes from inertial user traces (Section 5). Finally, inertial data with images and WiFi/cellular signatures are used to identify connection areas between stories, and a refinement algorithm is employed to correct detection errors, thus we generate the multi-story floor plan (Section 6).

3 Landmark Modeling

In this section, we describe how we extract sizes and coordinates of major geometry features (e.g., widths of store entrances, lengths/orientations of adjoining walls) of landmarks from their images.

3.1 The Landmark Model

We use a very simple model to describe the major geometry features of a landmark. As illustrated in Fig. 2, a landmark is denoted by $L = (P, Q)$, where P are the main geometric vertices of the landmark (e.g., the four corners $P_1 \sim P_4$ of a store entrance), and Q are connecting points of adjoining wall segments on the floor (e.g., $Q_1 \sim Q_3$ for two wall segments). Each landmark has a local coordinate system, and we place its origin C at the center of the store's entrance line $\overline{P_3P_4}$. The X-axis is co-linear with $\overrightarrow{CP_4}$, the X-Y plane is the ground floor, and the three axes follow the right-hand rule.

We leverage the output of two computer vision techniques, Structure from Motion(SfM) [15] and vanishing line detection [16], to obtain the coordinates of P, Q from landmark images.

Structure from Motion is a mature computer vision technique commonly used to construct the 3D models of an object. Given a set of images of the same object (e.g., a building) from different viewpoints, it produces: 1) a "point cloud" consisting of many points in a local 3D coordinate system. Each point represents a physical point on the object[1]; and 2) the pose (i.e., 3D coordinates and orientations) of the camera for each image, which effectively localizes the camera/user taking that image.

Using SfM only and as-is, however, may not be the best match for indoor floor plan reconstruction. First, SfM relies on large numbers of evenly distributed stable and distinctive image features for detailed and accurate 3D model reconstruction. Although landmarks themselves usually enjoy rich features due to logos, decorations, many in-between sections have too few (e.g., textureless walls), interior (e.g., transparent glass walls) or dynamic (e.g., moving customers) features, which SfM may not handle well. Second, the "point cloud" produced by SfM is not what we need for constructing floor maps. We still have to derive the coordinates of those geometric features in our model, e.g., the corners of an entrance.

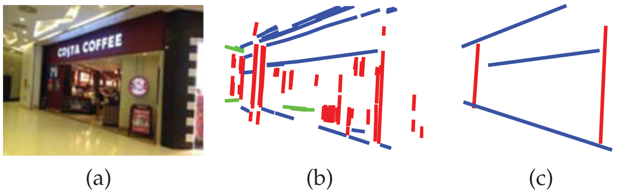


Fig. 3. Geometric vertices detection work flow: (a) Original image. (b) Detect line segments parallel to the three orthogonal axes. (c) Merged long line segments corresponding to the landmark's major contour lines. Different colors represent different dimensions.

3.2 Coordinates of Geometric Vertices

To obtain the coordinates of major geometry vertices needed in the model, we explore a two-phase algorithm. First, we use an existing vanishing line detection algorithm [16] to produce line segments for each image of the same landmark (Fig. 3b). We merge co-linear and parallel segments close to each other into long line segments (Fig. 3b). This method is done using an intersection angle threshold and a distance threshold between two line segments, and both thresholds are set empirically. The merging is repeated for all line segment pairs until no further merging is possible. We filter out the remaining short segments and leave only the long ones.

Next, we project merged 2D long lines from each image back into the 3D coordinate system using transformation matrices produced by SfM [15]. We then use an adapted k-means algorithm to cluster the projected 3D lines into groups according to their distance in 3D, and merge each cluster into a 3D line segment. This gives the likely 3D contour lines of the landmark. The intersection points of them are computed for major geometry vertices.

One practical issue that the above algorithm addresses is images taken from relatively extreme angles. Long contour lines (e.g., $\overline{P_1P_2}$ in Fig. 2) may become a short segment on such pictures. Because the majority of images are taken more or less front and center, real contour lines will have sufficient numbers of long line segments after the merging and projection. Thus the second phase clustering can identify them while removing "noises" from images of extreme angles. Due to the same reason, we find that the coordinates of wall segment connecting points farther from the center are not as accurate. This is simply because most images would cover the center of the landmark (e.g., store entrance) but may miss some peripheral areas farther away. Next we use a more reliable method to derive coordinates of wall connecting points.

3.3 Connecting Points of Wall Segments

We project the 3D point cloud of the landmark onto the floor plane, and search for densely distributed points in a line shape to find wall segments and their connecting points. This is because the projection of feature points on the same vertical plane/wall would fall onto the joining line to the

[1] To be more exact, each point represents a "feature point" as detected by certain feature extractor algorithms (e.g., SIFT [17]).

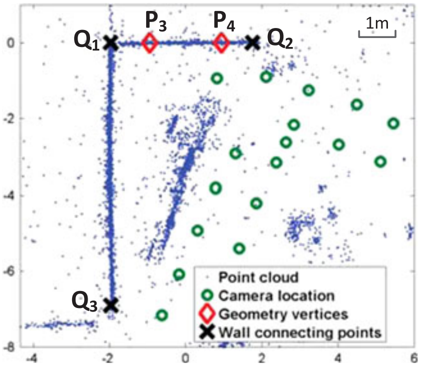


Fig. 4. A landmark's point cloud projected to the floor plan, with camera locations, critical contour line (P_3 and P_4) and connecting points of wall segments (Q_1, Q_2, and Q_3).

floor (e.g., $\overline{P_3Q_1}$ of the wall segment adjoining the entrance on left).

We start from some geometry vertices computed previously (e.g., $\overline{P_3P_4}$ gives the projected line of the entrance wall in Fig. 2, marked as two diamonds in Fig. 4), then find the two ends (e.g., marked as two crosses in Fig. 4) of this wall. From each end the search for the next connecting point continues, until no lines consisting of densely distributed points can be found. Fig. 4 shows three wall connecting points discovered.

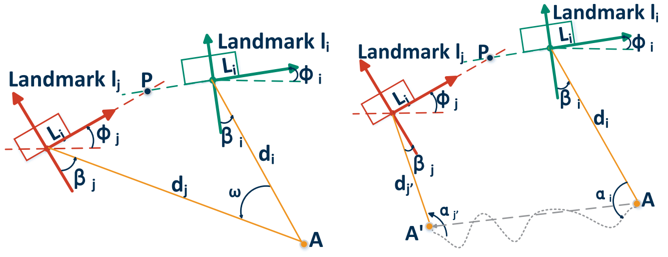
3.4 Example

Fig. 4 shows the point cloud of one store entrance projected onto the floor plane and SfM produced camera locations. We mark the geometry vertices (diamonds) and the wall connecting points (crosses). In this example, the width of the entrance has an error of 0.086 m (4.78 percent of the actual width 1.8 m). We also detect two external wall segments along the hallway, and their intersection angle error is 0.08 degree out of 90 degree (0.09 percent). We find that the 176 camera locations produced by SfM (only some of them are shown) are quite accurate. The localization error is within 1.2 m at 90 percent percentile, and maximum error is 1.5 m. We also test how the number of images impacts SfM's localization performance. As we vary the number of photos from 20 to 160, we find that about 80 images are sufficient for camera localization: 75 (94 percent) images are localized, with 90 percent error of 1.8 m and maximum error of 5.2 m. We will present more systematic evaluation results in Section 7.

4 Landmark Placement

In this section, we estimate the *configuration* of landmarks, which is defined as the coordinates and orientations of landmarks in the global 2D coordinate system. We also derive the global coordinates of locations where photos are taken. To this end, we first obtain the spatial relationship between adjacent landmarks from inertial and image data. The determination of the configuration is formulated as an optimization problem that finds the most likely coordinates and orientations of landmarks that achieve the maximal consistency with those pairwise relationship observations.

Once the landmarks' global coordinates are known, the global positions where photos are taken is a simple coordination transformation of the camera location in each landmark's local coordinate system (described in Section 3) to the global one. Such camera positions play an important role in the *augumentation algorithm* for the occupancy map in Section 5.



(a) Click-Rotate-Click (b) Click-Walk-Click

Fig. 5. Micro-tasks: A is where two photos of landmark l_i and l_j are taken in CRC. (d_i, β_i) are the length of AL_i, and the angle formed by line AL_i and normal direction of L_i, respectively. P is the intersection point of the two x-axes of the two local coordinate systems. A' is where the walk ends in CWC.

4.1 Notations

Suppose there are n local coordinate systems corresponding to n landmarks $l_1, l_2, \ldots, l_n$. $X_i = (x_i, y_i) \in \mathbb{R}^2$ and $\phi_i \in [-\pi, \pi)$ are the x-y coordinates and orientation of landmark l_i in the global coordinate system, respectively. $\boldsymbol{\theta} = (\boldsymbol{X}, \boldsymbol{\phi})$ is the configuration of landmarks to be determined, where $\boldsymbol{X} = (X_1, \ldots, X_n)$, $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$. $R_i = R(\phi_i) = \begin{bmatrix} cos\phi_i & -sin\phi_i \\ sin\phi_i & cos\phi_i \end{bmatrix}$ is the rotation matrix used in coordinate transformation between the global and local coordinate systems of landmark l_i. $X^i_j = (x^i_j, y^i_j) = R(\phi_i)^T (X_j - X_i)$ and $\phi^i_j = \phi_j - \phi_i$ are the x-y coordinates and orientation of landmark l_j in the local coordinate system of landmark l_i, respectively.

4.2 Spatial Relation Acquisition

The spatial relationship between two adjacent landmarks l_i, l_j are X^i_j and ϕ^i_j, the coordinates and orientation of landmark l_j in the local coordinate system of landmark l_i (or vice versa, illustrated in Fig. 5). It is difficult to obtain such measurements directly from users because they do not carry tools such as tapes. We design two data-gathering micro-tasks where the user takes a few actions to gather inertial and image data, from which we compute the pairwise relationship observations.

Click-Rotate-Click (CRC): In this micro-task, a user clicks to take a photo of a landmark l_i from position A (shown in Fig. 5a), then spins the body and camera for a particular angle (e.g., ω degrees) to take another photo of a second landmark l_j. The angle ω can be obtained quite accurately from the gyroscope [4], [5]. (d_i, β_i) represents the distance between camera A and landmark l_i, and the angle formed by line L_iA and the normal line of landmark l_i, respectively. They can be derived from the camera pose (i.e., coordinates and orientation in l'_is location coordinate system) as produced by SfM (Section 3). Similar is (d_j, β_j). P represents the intersection point of the two x-axes in the two landmarks' local coordinate systems.

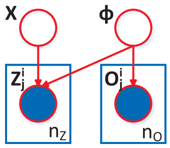


Fig. 6. Bayesian belief network representation of our problem. X is the coordinates while ϕ is the orientations of all the landmarks. $\theta = (X, \phi)$ is the hidden variable we need to estimate based on measurements. Z_j^i, O_j^i measures the coordinates and orientation of landmark j in the coordinates system of landmark i. Measurements of each kind are aggregated together with the total number of that kind denoted by n_Z, n_O.

From plane geometry, quadrangle AL_iPL_j is uniquely determined given $(d_i, \beta_i), (d_j, \beta_j)$ and ω. Thus we can calculate an observation of one landmark's coordinates and orientation in the other's local coordinate system (and vice versa), namely, observations of (ϕ_j^i, X_j^i), (ϕ_i^j, X_i^j) denoted as (O_j^i, Z_j^i) and (O_i^j, Z_i^j).

Click-Walk-Click (CWC): In this micro-task, a user clicks to take a photo of landmark l_i, then walks to another location A' to take another photo of a second landmark l_j (shown in Fig. 5b). It is useful when two landmarks are farther away and finding one location to take proper photos for both is difficult. The distance $|AA'|$ could be calculated from step counting method [5], and the angle between the direction when user takes a photo and his/her walking direction, i.e., (α_i, α'_j) at two locations A and A', could be obtained from placement offset estimation method [18] and gyroscope readings. Measurements calculation here is similar to that of Click-Rotate-Click except that the quadrangle is replaced by a pentagon as illustrated in Fig. 5a.

The two camera locations in CWC can be used as "anchor points" to calibrate the trace. Due to well-known error accumulation [11] in inertial tracking, many methods use anchor points (places of known locations such as entrances/exits of escalators, elevators, stairs) to pinpoint the trace on the floor. In environments with large open space, such anchor points may be sparse. CWC addresses the sparsity issue because users can take photos almost anywhere.

Nevertheless, we use CWC between two landmarks only when CRC is difficult to conduct, because the accuracy of step counting based inertial tracking is limited compared to that of the gyroscope in CRC. Jigsaw utilizes both types of measurements while considering their varying qualities, by assigning different confidences to each type in a common optimization problem (described next in Section 4.3).

4.3 Problem Formulation

We use Maximum Likelihood Estimation (MLE) to formulate the optimal configuration problem. Our problem is represented as a Bayesian belief network (Fig. 6) describing the conditional dependence structure among variables (denoted as nodes), where each variable only directly depends on its predecessors.

We denote the maximum likelihood estimation of θ as θ^*. The intuition for maximizing $P(Z, O|X, \phi)$ is that we try to find a configuration of landmarks $\theta^* = (X^*, \phi^*)$ under which those measurements Z, O (i.e., observations of X, ϕ) are most likely to be observed.

We have the following equations based on the conditional dependence in the graphical model:

$$\begin{aligned}\theta^* &= \arg\max_{\theta} P(Z, O|X, \phi) = \arg\max_{\theta} P(O|\phi)P(Z|\phi, X)\\ &= \arg\min_{\theta} -\sum_{O_j^i} \log P\big(O_j^i|\phi\big) - \sum_{Z_j^i} \log P\big(Z_j^i|\phi, X\big).\end{aligned}$$

As is standard in probabilistic mapping literature [19], we assume Gaussian measurement models that give further transformation into:

$$\theta^* = \arg\min_{\theta} \sum_{O_j^i} \frac{\|\phi_j^i - O_j^i\|^2}{\sigma_O^2} + \sum_{Z_j^i} \frac{\|X_j^i - Z_j^i\|^2}{\lambda_Z^2}. \tag{1}$$

where σ_O, λ_Z are covariances of normally distributed zero-mean measurement noises for different kinds of measurements. As noted in Section 4.2, we assign small σ_O, λ_Z for CRC measurements to give them predominance over those of CWC.

Without losing generality, we can simply use variable substitution to yield an equivalent nonlinear least squares formulation:

$$\underset{\phi, X}{\text{minimize}} \sum_{O_j^i} \|\phi_j^i - O_j^i\|^2 + \sum_{Z_j^i} \|X_j^i - Z_j^i\|^2. \tag{2}$$

The intuition is that we try to find a configuration of landmarks $\theta^* = (X^*, \phi^*)$ such that the aggregate difference between ϕ_j^i, X_j^i derived from (X^*, ϕ^*) and their measurements O_j^i, Z_j^i is minimized.

4.4 Optimization Algorithm

Let's denote problem (2) as:

$$\underset{\phi, X}{\text{minimize}}\ f(\phi) + g(\phi, X), \tag{3}$$

since the two terms in (2) are functions of ϕ and (ϕ, X). Careful examination [20] shows that each term in $g(\phi, X)$ is linear square of X, thus $g(\phi, X)$ is a typical linear least squares of X with a closed form solution. We denote the minimum as $h(\phi)$. Thus problem (3) is equivalent to:

$$\underset{\phi}{\text{minimize}}\ f(\phi) + h(\phi), \tag{4}$$

We solve this problem based on an observation: minimizing $f(\phi)$ gives the most likely orientation ϕ' of landmarks with orientation relationship observations only. Due to relatively accurate gyroscope data, ϕ' would be very close to the global optimal ϕ^* that minimizes $f(\phi) + h(\phi)$. Thus we find the optimum of $f(\phi)$ as the initial value, then use stochastic gradient descent (SGD) to find the global minimum ϕ^*.

STEP 1: Find ϕ' given measurements O.

$$\underset{\phi}{\text{minimize}}\ f(\phi) = \sum_{O_j^i} \|\phi_j^i - O_j^i\|^2. \tag{5}$$

Note that this is not a linear least squares problem since the result of the subtraction on angles is periodic with a period of 2π. What adds to the difficulty is the loop dependence of the orientations of different landmarks. The effect of

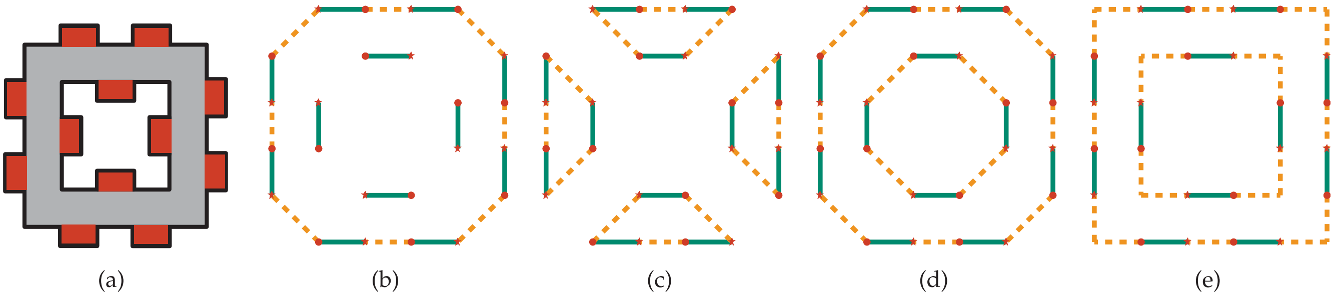


Fig. 7. Comparison between different algorithms: (a) Example scenario, (b) convex hull of all wall segments, (c) one possible output of the greedy method, (d) minimal weight matching using distance as weight, and (e) our minimal weight matching method.

adjusting the orientation of one landmark would propagate along pairwise relationship observations, eventually back to itself.

We solve this problem as follows: First, we find the maximum spanning tree of the orientation dependence graph where edges are relationship observations between landmarks. This problem $f_{MST}(\boldsymbol{\phi})$ can be easily solved because adjusting the orientation of one landmark has a one-way effect on its decedents only. Again, due to the accuracy of gyroscope and relatively small number of removed edges (i.e., relationship observations), the resulting $\boldsymbol{\phi}'_{MST}$ would be in near neighborhood of the true optimum $\boldsymbol{\phi}'$. Then we perform gradient descent from $\boldsymbol{\phi}'_{MST}$ find a minimum likely to be $\boldsymbol{\phi}'$. In reality, we do find them are usually in the close neighborhood.

STEP 2: perform stochastic gradient descent (SGD) from $\boldsymbol{\phi}'$ to find $\boldsymbol{\phi}^*$. Based on the intuition explained earlier that $\boldsymbol{\phi}'$ is close to $\boldsymbol{\phi}^*$, we perform SGD which is known to be able to climb out of "local minima" to find the global minimum with higher probability.

5 Map Augmentation

After obtaining the optimal coordinates and orientations of the landmarks, we need more details for a relatively complete floor plan: 1) wall reconstruction for external boundaries of the hallway; 2) hallway structure; and 3) rough shapes of rooms. Next, we describe how to construct such details.

5.1 Wall Reconstruction

Connecting wall segments between adjacent landmarks in manners "most consistent" with the likely architectural structure of buildings is not trivial. Naive methods such as using a convex hull to cover all segments produce an external boundary but may not connect those segments "inside" the hull (Fig. 7b).

To formally define the problem, we represent a wall segment as a line segment with its normal direction pointing to the hallway, and denote the endpoints on its left/right side as L and R (shown in Fig. 8). Therefore, k wall segments have two sets of endpoints $\boldsymbol{L} = \{L_1, L_2, \ldots, L_k\}$ and $\boldsymbol{R} = \{R_1, R_2, \ldots, R_k\}$. We need to add new wall segments connecting each endpoint in $\boldsymbol{L}$ to one in $\boldsymbol{R}$.

Every possible solution corresponds to a perfect matching π, where π is a permutation of $(1, 2, \ldots, k)$, indicating $L(i)$ and $R(\pi(i))$ are linked for $i = 1, 2, \ldots, k$. Thus the problem becomes a combinatorial optimization problem that finds the perfect matching with the minimal weight (i.e., most likely connection manner) in a bipartite graph.

A simple greedy algorithm uses distance as weight and connects every endpoint in set $\boldsymbol{L}$ to the closest (i.e., least distance) one in set $\boldsymbol{R}$ directly. The drawback is that the result depends on the order of connecting endpoints, and 90 degree corners commonly seen in buildings may be missing. For example, Figs. 7c and 7d show two possible results, where one is incorrect while the other does not have 90 degree corners.

To address the above issues, we consider the two following options for linking two adjacent wall segments. Each option carries a weight, which can be computed given two endpoints in $\boldsymbol{L}$ and $\boldsymbol{R}$. The weight represents the likelihood of the option: a smaller one indicates a more likely linking manner.

Linking with another segment directly. Two segments (L_i, R_i) and (L_j, R_j) are linked by another segment between L_i and R_j directly. The weight is defined as:

$$w_{ij}^{(1)} = |R_i - L_j|(\omega_1 + \omega_2), \tag{6}$$

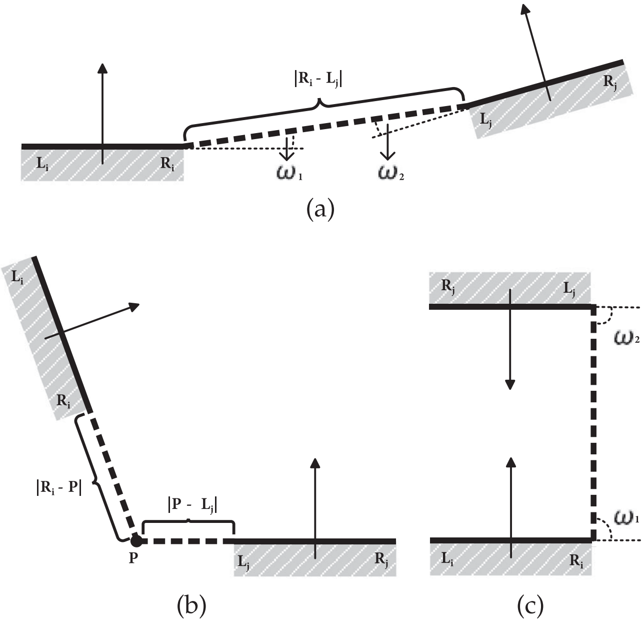
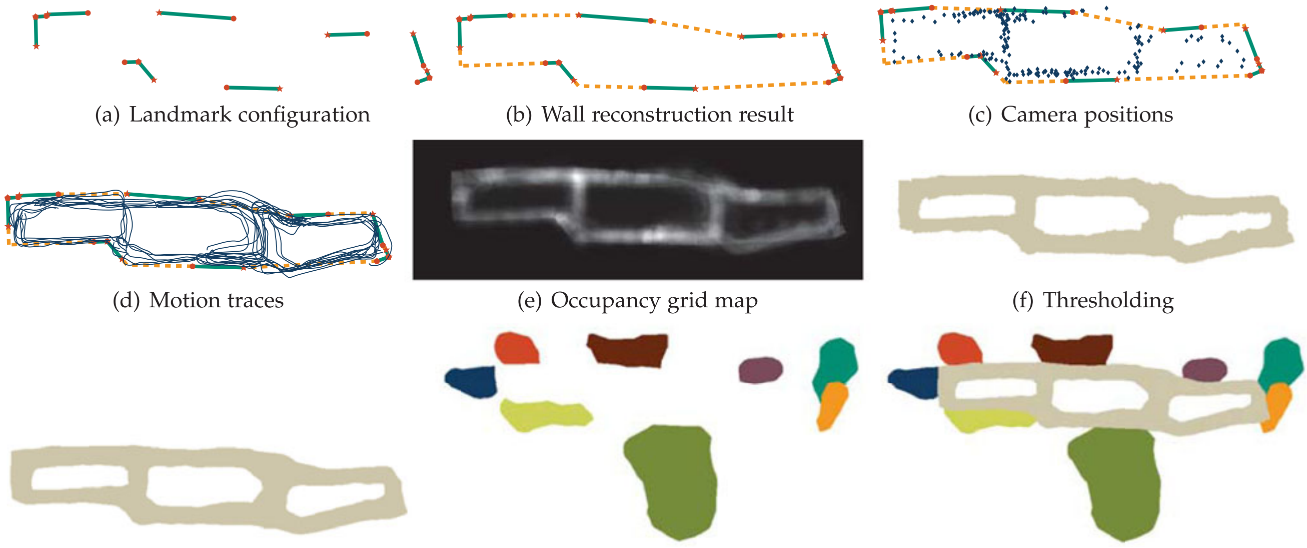


Fig. 8. Given the normal direction pointing to the hallway, two endpoints of a wall segment are labeled L and R. New wall segments must link endpoints with different labels. Three cases of connection are shown: (a) two nearly collinear segments; (b) two nearly perpendicular segments; and (c) two nearly opposite segments.



(a) Landmark configuration (b) Wall reconstruction result (c) Camera positions

(d) Motion traces (e) Occupancy grid map (f) Thresholding

(g) Smoothing (h) Room reconstruction result (i) Final floor plan

Fig. 9. Augmentation process: (a) Shows landmark configuration results. (b) Depicts hallway external boundary after wall reconstruction. (c) and (d) Show camera positions and motion traces. Combining the above, occupancy grid map is shown in (e), followed by thresholding (f), and smoothing (g). (h) Depicts room reconstruction results and the final floor plan is shown in (i).

where $|R_i - L_j|$ is the distance between two endpoints R_i and L_j and ω_1, ω_2 are the turning angles from segments (L_i, R_i), (L_j, R_j) to the newly added segment (illustrated in Figs. 8a and 8c). Such direct linking is more likely when two adjacent segments are collinear or facing each other.

Extending to an intersection. If the two segments are not parallel, extending them from endpoints R_i and L_j reaches a point of intersection. This is another possibility and its weight is defined as:

$$w_{ij}^{(2)} = \frac{|R_i - P| + |P - L_j|}{2}, \tag{7}$$

where P is the point of intersection and $|R_i - P|$ and $|P - L_j|$ are the distances among them (illustrated in Fig. 8b). For two (close to) perpendicular segments, the above equation produces a smaller weight, ensuring proper connection for 90 percent corners.

Given the configuration of landmarks estimated in Section 4, we calculate $w_{ij}^{(1)}$ and $w_{ij}^{(2)}$ for each pair of wall segments based on (6) and (7). We define the weight w_{ij} of linking L_i and R_j as the smaller of the two:

$$w_{ij} = \begin{cases} \min(w_{ij}^{(1)}, w_{ij}^{(2)}), & i \neq j; \\ \infty, & i = j. \end{cases}, \tag{8}$$

the weight is ∞ if $i = j$ since the two endpoints of the same segment is already connected.

Given all the weights, we can find the perfect matching π^* to minimize the total weight as follows:

$$\underset{\pi}{\text{minimize}} \sum_{i=1}^{k} w_{i\pi(i)}. \tag{9}$$

While a naive exhaustive search needs factorial time, we recognize that finding the perfect matching with minimal weight in a bipartite graph can be solved efficiently by Kuhn-Munkres algorithm [21] in polynomial time ($O(n^3)$) where n is the number of landmarks, which is usually a small number (e.g., tens of stores for one floor of a mall). Fig. 7e shows the correct result produced by our algorithm and Fig. 9b illustrates the outcome in a real environment.

5.2 Hallway Reconstruction

To reconstruct the structure of the whole hallway, we first build the occupancy grid map [22], which is a dominant paradigm for environment modeling in mobile robotics. Occupancy grid map represents environments by fine-grained grid cells each with a variable representing the probability that the cell is accessible.

In Jigsaw, it can be regarded as a confidence map that reflects the positions accessible to people. This confidence map is initialized as a matrix full of zeros. We add confidence to a cell if there is evidence that it is accessible, and the scale of the confidence we add depends on how much we trust the evidence. We fuse three kinds of cues to reconstruct the occupancy grid map.

External boundary of the hallway: This is re-constructed in Section 5.1. Due to obstacles (e.g., indoor plants placed next to the wall), the accessible positions are not equivalent to the region bounded by the external boundary. Since the area in front of landmarks is often the entrance, it is always accessible and we assign higher confidence. Places in front of a newly added wall are usually accessible but obstacles may exist. Thus, we assign less confidence to such places.

Positions of cameras: Positions of cameras can be computed given the configuration of landmarks and the relative position between cameras and landmarks. Such positions are obviously accessible. So we add confidence to places around every camera's position. Fig. 9c depicts positions of cameras with the result of wall reconstruction.

Motion traces in the hallway: The shape of motion traces can be computed using methods such as [5], [18]. The traces can be calibrated by taking photos and using their locations as anchor points. Given such information, we can correct the step length, which is one primary source of error in

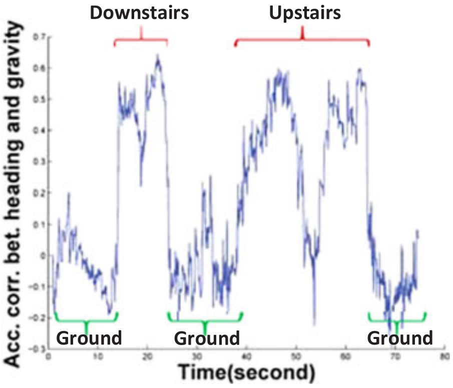


Fig. 10. Acceleration correlation between heading and gravity directions.

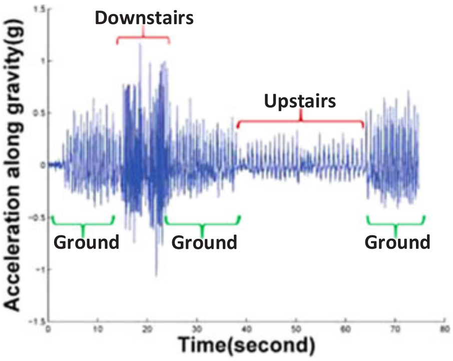


Fig. 11. Acceleration along gravity direction.

step-counting based tracking. Such traces in the hallway add confidence to positions along them. Because motion traces usually carry higher errors, we assign less confidence along motion traces comparing to positions of cameras. Fig. 9d depicts motion traces in the hallway with the result of wall reconstruction.

The final occupancy grid map is shown in Fig. 9e. We use an automatic threshold-based binarization technique [23] to determine whether each cell is accessible, thus creating a binary map indicating which cells are accessible. The accumulation of evidences makes our method robust to noises and outliers in crowdsensed input: a cell is considered accessible only when there is enough evidence. The result of thresholding is depicted in Fig. 9f. To further improve the result, we implement a smoothing algorithm based on *alpha-shape* [11], which is a generalization of the concept of convex hull. Fig. 9g shows the final result of hallway reconstruction after smoothing.

5.3 Room Reconstruction

We use the same confidence map technique to fuse two kinds of cues for robust estimation of the shape of rooms.

Wall segments in landmark models: These wall segments are not only part of the external boundary of the hallway, but also the boundary of the room. Therefore, the places inside the detected wall segments are part of the room with high confidence.

Motion traces inside the room: We have a data-gathering micro-task similar to CWC to collect data for rooms. A user takes a photo of a landmark, then walks into this room. After walking for a while, the user exits and takes another photo. The photos are used to determine the initial/final locations of the trace, and the area along the trace receives confidence. We perform similar thresholding of the cumulated confidence to determine the accessibility of each cell, producing room reconstruction results similar to that shown in Fig. 9h. The final floor plan at the end of map augmentation is in Fig. 9i.

6 Connection Area Detection

So far we have generated the floor plan of a single floor. However, typical indoor environments always contain multiple floors, along with connection areas between adjacent floors (e.g., stairs, elevators and escalators). We find that the inertial and wireless signals (e.g., WiFi and cellular) have distinctive patterns when a user passes through such areas. We use unsupervised classification to detect such patterns without training process, and develop refinement algorithm to correct detection errors.

6.1 Types of Connection Areas

Stairs generate repetitive jolts, hence periodic acceleration fluctuations in the gravity direction when a user climbs. Note that going upstairs or downstairs may cause different jolting patterns, and we need to recognize each of them correctly. We also need to distinguish stairs from walking on the same flat floor. A significant clue is that WiFi signatures always change dramatically between two different floors. We use *WiFi cosine distance* [24], i.e., the cosine value of the angle between two vectors of WiFi signatures, to represent their similarity. Higher cosine distance indicates similar WiFi signatures, and vice versa. From large amounts of experiment data, we observe that WiFi cosine distance between stairs is mostly between $0.65 \sim 0.75$, apparently lower than that of walking on the same floor ($0.8 \sim 0.85$). Detailed evaluation is shown in Fig. 21a.

Inertial patterns are also differentiated between walking on floor and stairs. Fig. 10 indicates that the acceleration correlation between heading direction and gravity direction is much lower on the floor than that of stairs. Fig. 11 shows the acceleration along gravity direction is much lower for upstairs than downstairs: the reason is that gravity is impeding/helping user motion when walking up/down stairs.

Escalators and elevators. Users always stand still in elevators/escalator while their absolute positions change dramatically. To distinguish from standing on the floor, we observe that the WiFi cosine distance between beginning and end of escalator/elevator rides are always significantly smaller ($0.65 \sim 0.8$), compared to standing on the ground for a similar duration (~ 0.95). Detailed results are shown in Fig. 21b. This observation can be used to distinguish them from standing on the floor. Furthermore, elevators can be easily detected via obvious fading of cellular signals (more than 30dbm based on our measurements).

To tell the moving direction (up or down), we observe that that there are temporary decrease and increase of acceleration along gravity direction at the beginning/end of the ride (Fig. 12), and the opposite for going up. We compute the difference of 5 s time window average for the beginning/end of the ride to detect the direction.

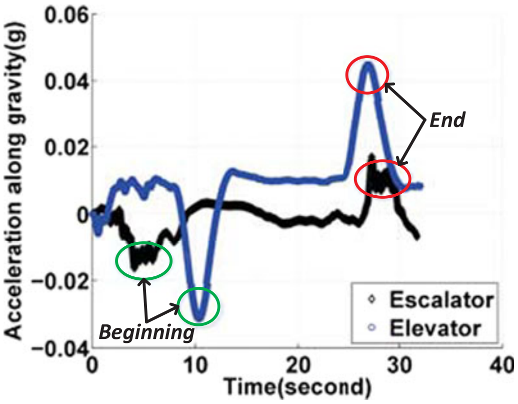


Fig. 12. Acceleration along gravity direction for downward escalator/elevator, after Butterworth filter.

6.2 Features

We extract the following features from inertial, WiFi, and cellular data for recognize the type of connection area.

(1) *Acc MNAC* and *Acc STD*: Acc MNAC denotes the maximum normalized auto-correlation of acceleration along gravity direction, which computes the period of repetitive walking patterns; Acc STD is the standard deviation of that acceleration, which implies user movements. They have been used to tell whether a user is standing still or walking [5].

(2) *Acc COR* and *Acc PV*: Acc COR is the acceleration correlation between heading and gravity directions; it helps to identify stairs; Acc PV are peak values of acceleration along gravity direction; they are used to distinguish their up/down cases [11].

(3) *WiFi CD*: WiFi cosine distance between two endpoints of a time window. This feature tells whether a user passes through floors. The intuition is that WiFi cosine distance between two floors is obviously smaller than staying on the same floor for a short time window (e.g., $\sim$15 s of passing across adjacent floors).

(4) *Cellular SD*: Cellular signal declination. This feature helps to identify the elevator since the closed metal environment dramatically attenuates signals.

6.3 Unsupervised Classification

Next we propose an unsupervised classification algorithm to identify different types of connection areas. We avoid learning techniques that require training that is difficult in practice, especially for crowdsourced mobile users, and training models in various environments always differ a lot. Instead, we automatically cluster the features into different categories and develop an unsupervised classification algorithm via majority voting.

Step 1: Walking detection. Zee [5] computes walking periods on each trace but it relies on hard thresholds of Acc MNAC and Acc STD. We observe that those thresholds change dramatically for different user walking styles and different smartphones, which makes uniform settings impossible. We propose a pre-task in data collection: before walking a CWC micro-task, a user stands still for around 5 seconds, then walks with the phone held steady. We leverage k-means algorithm [25] ($k=2$) to generate the thresholds for each user, and obtain similar results in [5] with step counting errors only at start/end of a trace.

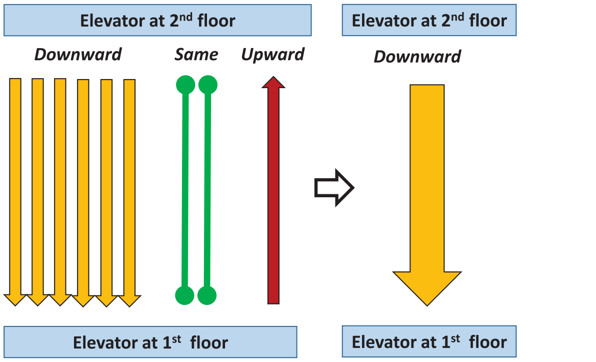


Fig. 13. Landmark connection edges of an elevator between two floors. Majority voting is used to correct detection errors.

Step 2-1: Stairs detection: if a user is walking, we identify whether he/she walks on the floor or stairs. We set the time window as 15 s (approximate time for walking to an adjacent floor via stairs), leverage Acc COR and WiFi CD features, and use the k-means algorithm for recognition (with $k=2$ for ground/stairs). For stairs, we use Acc PV feature and k-means to identify the up/down direction.

Step 2-2: Escalator/elevator detection: if a user is standing still, we identify whether it is an escalator/elevator or the floor. We compute WiFi CD and cellular SD features for the whole standing still period and use k-means to identify each of them (with $k=3$ for escalator/elevator/ground). To detect up/down, we compare the average acceleration along gravity direction at the start/end of the ride, i.e., larger at the end than the start of the ride when going down, and vice versa. The above detection achieves very high accuracy (close to 100 percent) and details are in Table 3.

6.4 Refinement and Placement

While most of the above detection results are found to be correct, we observe that occasional errors can happen (e.g., a CWC data on the ground is incorrectly recognized as passing stairs, or an upward escalator is detected as downward). We use a simple majority voting to correct those minority errors.

We abstract a connection graph where nodes are landmarks, and each CRC/CWC constraint forms an edge between two landmarks, with a weight for its constraint type. Multiple edges may exist between two nodes when different constraints were measured. To correct erroneous edges, we use majority voting from all edges between two nodes to decide their connection relation (e.g., on the same story, upward or downward) and remove incorrect detections (Fig. 13).

We also need to generate locations of different types of connection areas on the reconstructed floor plan. For stairs, we place them via the 90-degree turn at its start/end points; for escalator and elevator, they are placed via user standing still locations on traces.

The final connection graph for mall 1 is shown in Fig. 14, which shows 100 percent correct detection of four types of landmark connection relations: the same floor, stairs, escalator, and elevator connection. The floor plans with connection areas are shown on Figs. 19a and 19b.

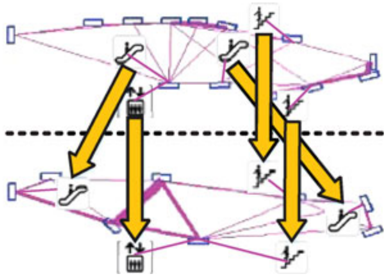


Fig. 14. Landmark connection graph of two floors. Line segments represent landmark connection on the same floor, with width for inputs quantity; arrows represent downward connections.

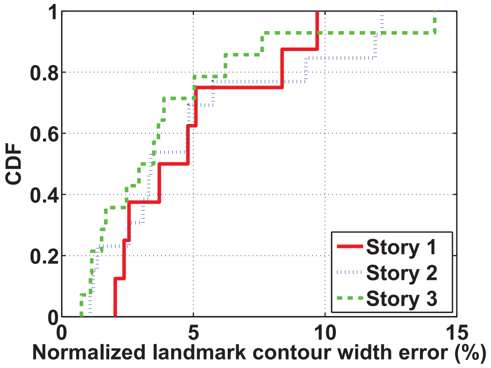


Fig. 15. Normalized store entrance width error.

7 Performance Evaluation

7.1 Methodology

We use iPhone 4s to collect images and motion sensor data, and Samsung Galaxy S II for WiFi scans.[2] We conduct experiments in three environments: two stories of a 150 m × 75 m shopping mall (labeled story 1 and 2) of irregular shape, and one story of a 140 m × 40 m long and narrow mall comprised of two parts connected by two long corridors (labeled parts I and II of story 3). In these environments, we select 8, 13 and 14 store entrances as landmarks and collect about 150 photos at different distances and angles for each landmark. In each environment, we have 182, 184 and 151 locations where users conduct "Click-Rotate-Click"(CRC) to take two images of two nearby landmarks, and 24 "Click-Walk-Click"(CWC) to take two images of two far away landmarks in different parts in story 3. We also collect 96, 106 and 73 user traces along the hallway of each environment, and about seven traces inside each store. To connect two stories of the first shopping mall, we observe that there are two stairs, two escalators and one elevator connecting them. Thus we also conduct 40 CWC measurements between two stories passing up/down for each stair, and 14 CWC measurements passing up/down for each escalator and elevator.

During data collection, users follow simple guidelines: 1) choose landmarks as large physical objects on the wall, such as store entrances and posters; 2) during walking, hold the phone steady; 3) take photos with the landmark in the center, and without obstructions and moving people on the image. These guidelines help users gather data of higher quality, and user feedbacks suggest the guidelines are easy to follow in practice.

7.2 Landmark Modeling

First, we examine the accuracy of estimated widths of store entrances using normalized error (error divided by the actual width). As Fig. 15 shows, 90-percentile error is about 10 percent, which indicates that the inferred parameters of major geometry vertices are quite accurate. We also find that large errors are caused by obstructions such as pillars or hanging scrolls.

We evaluate wall segment detection accuracy, and observe that the recall (i.e., detected wall segments in all existing segments) is 91.7, 88.2 and 100 percent in three indoor environments, respectively; while the precision (i.e., fraction of detected ones that are correct) are all 100 percent. We find that the wall connecting point detection is quite accurate as well. Those segments not detected are due to extreme angles (e.g., less than 15 percent difference) to the entrance wall, which are considered part of the same segment by the algorithm.

We measure how the quantity of images impact localization accuracy to understand its impact on SfM performance. For each environment, we randomly select 50, 100 and 150 images for each landmark. We can see that the fraction of localized images increases steadily from 74.5, 95 to 99.3 percent. When there are sufficient images (e.g., 100 ~ 150), nearly all of them are localized.

We also observe a similar trend in the average localization error for each landmark in story 1 (shown in Fig. 16; the other two are similar). When there are sufficient images, the average error is less than 2 m. Thus, 100 ~ 150 images for each landmark would be an appropriate amount.

Finally, we examine image localization accuracy (shown in Fig. 17). We observe that localization errors are about 1 ~ 1.5 m at 90-percentile. The large errors are due to "isolated" images taken from extreme distances (e.g., too faraway) or angles (e.g., almost parallel to the entrance wall), which cannot find enough matching feature points with the majority of images taken more front and center. We observe that story 3 has the least error due to its smaller size, so images are distributed more densely and thus appear similar to each other.

7.3 Landmark Placement

Measurements accuracy. We first evaluate the relative position and orientation errors as derived from pairwise measurements (Section 4.2) between adjacent landmarks. The relative position error is the distance between a landmark's

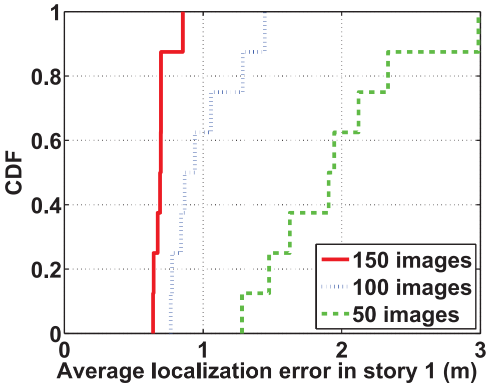


Fig. 16. Impact of image quantities.

[2] iOS public API does not give WiFi scan results.

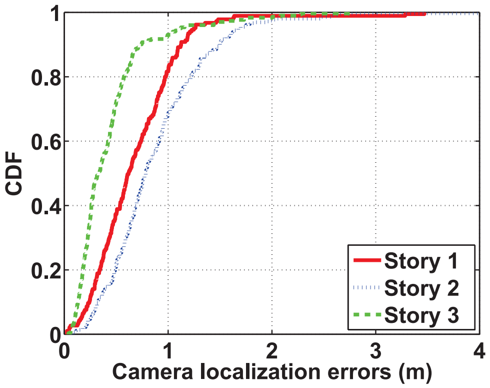


Fig. 17. Image localization errors.

derived position and its actual position, both in the other landmark's coordinate system. Relative orientation error quantifies how close the derived orientation difference is to ground truth.

We use $182, 184, 151$ CRC measurements in three environments, and 24 CWC measurements in story 3 between its two parts.

Figs. 18a and 18b show the cumulative distribution functions (CDF) of relative position and orientation errors in three environments. We can see that for CRC measurements, the 80-percentile relative position errors are about $2 \sim 7$ m, while that of relative orientation about $10 \sim 20^\circ$, both of which have quite some inaccuracies. CWC measurements have worse position errors (80-percentile around 10 m) but comparable orientation errors (80-percentile at 15 percent). This is because of errors in stride length estimation, but the gyroscope remains accurate.

Landmark configuration. We compare the computed landmark positions and orientations to the respective ground truth to examine errors in the derived configuration. Figs. 18c and 18d show the CDFs of position and orientation errors. Since CRC measurements alone cannot join the two parts of story 3, we give CRC accuracy for part I (containing most stores). Compared with respective errors in measurements (shown in Figs. 18a and 18b), both position and orientation errors improve (e.g., $1 \sim 2$m and $5 \sim 9^\circ$ at 90-percentile). This is because errors in measurements are statistically symmetric; thus, the impacts tend to cancel out each other.

After CWC measurements are combined, there is not much change in orientation but slight degradation in positions (e.g., 2.5 m at 90-percentile) due to lower accuracy of CWC position measurements. This shows that CWC may impact the accuracy. Thus, we use them only when CRC alone cannot establish the spatial relationship between faraway landmarks. The relative positions in each part do not change much, due to different weights assigned to CRC and CWC measurements. In summary, the landmark placement algorithm successfully combines all measurements for better estimation of the most likely configuration.

7.4 Floor Plan Performance

The reconstructed floor plans and their respective ground truths are shown in Figs. 19a, 19b, 19c and Figs. 19d, 19e, 19f.

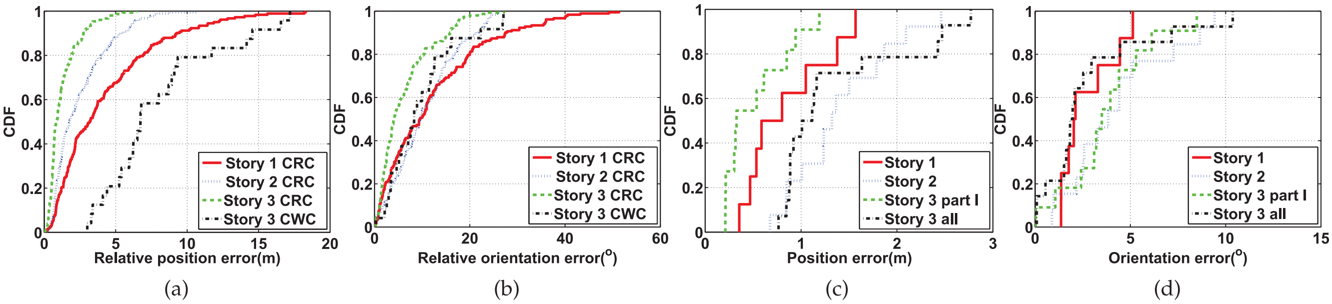
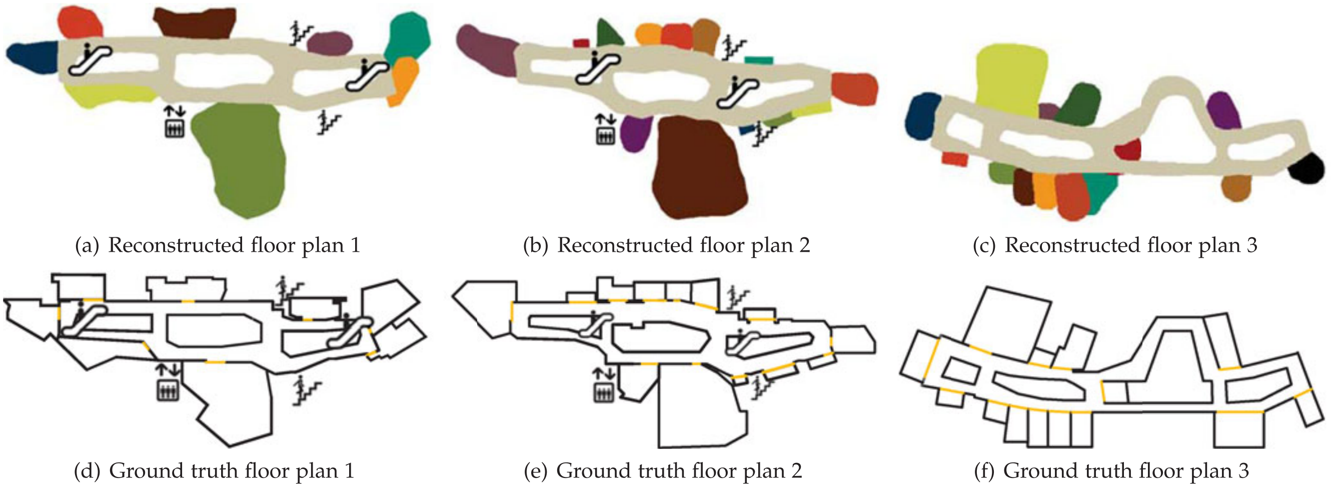


Fig. 18. CDFs of landmark placement evaluation: (a) relative position error extracted from crowdsourced data; (b) relative orientation error extracted from crowdsourced data; (c) position error of proposed algorithm for landmark level mapping; and (d) orientation error of proposed algorithm for landmark level mapping.



(a) Reconstructed floor plan 1 (b) Reconstructed floor plan 2 (c) Reconstructed floor plan 3

(d) Ground truth floor plan 1 (e) Ground truth floor plan 2 (f) Ground truth floor plan 3

Fig. 19. Reconstructed floor plans and ground truth floor plans.

TABLE 1
RMSE of Floor Plans (m)

	Landmarks	Intersections
Story 1	0.94	1.25
Story 2	1.49	1.80
Story 3	0.61/0.15	0.91/0.49

Positions of feature points. We evaluate the quality of floor plans using the root mean square error (RMSE). Given n feature positions on a floor plan with 2D coordinates $X_i^{map} = (x_i^{map}, y_i^{map})$, and their corresponding ground truth coordinates $X_i^{test} = (x_i^{test}, y_i^{test})$, $i = 1, 2, \ldots, n$, the RMSE is calculated by

$$e_{RMS} = \sqrt{\frac{\sum_{i=1}^{n} (X_i^{map} - X_i^{test})^2}{n}}. \quad (10)$$

For each environment, we select two sets of feature positions, one for landmarks, the other for center points of hallway intersections. We can see that RMSEs of landmarks are small (e.g., $< 1.5m$) while those for intersections are slightly larger (Table 1). Note that for story 3, we calculate the RMSEs for the left and the right part separately since each part was reconstructed using relatively accurate CRC data while the connecting hallway between them uses less precise CWC data.

Hallway shape. We also evaluate how close the shapes of constructed hallways resemble the respective ground truth. We overlay the reconstructed hallway onto its ground truth to achieve maximum overlap by aligning both the center point and the orientation. Precision is the ratio of the size of the overlap area to the whole reconstructed hallway, and recall is that to the ground truth hallway. F-score is the harmonic average of precision and recall. The results are shown in Table 2. We can see that Jigsaw achieves a precision around 80 percent, a recall around 90 percent and a F-score around 84 percent for the first two stories. This shows the effect of the calibration of traces by camera locations, and probabilistic occupancy maps are more robust to errors and outliers. The reason that recalls are higher than precisions (as shown in Fig. 19) is that reconstructed hallway is a little thicker than the ground truth due to errors in traces. Story 3 has a relative lower performance because only CWC data can be used to connect the left and right parts.

Room size. We use the error of reconstructed room size as the metric. Jigsaw achieves an average error of 25.6, 28.3 and 28.9 percent respectively for three stories. Given the fact that some part of room is not accessible, the errors are relatively small since camera localization provides accurate anchor points to calibrate the errors of inertial traces and the probabilistic occupancy map provides robustness to outliers.

TABLE 2
Evaluation of Hallway Shape

	Precision	Recall	F-score
Story 1	77.8%	92.0%	84.3%
Story 2	81.2%	93.3%	86.9%
Story 3	74.5%	86.0%	79.8%

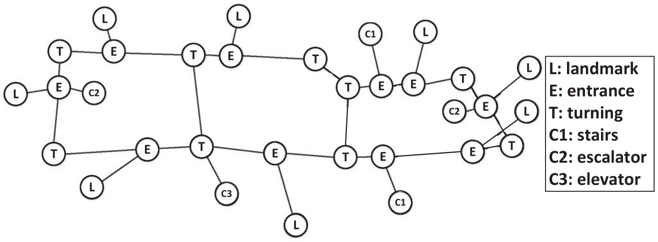


Fig. 20. Topological map of reconstructed floor plan for story 1, where nodes "L" denote rooms of landmarks, nodes "E" denote entrances along hallway, nodes "T" denote turnings along hallway, and nodes "C1/C2/C3" denote three types of connection areas.

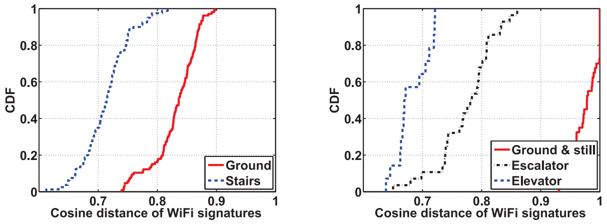
Topological structure. For indoor navigation, the topological structure of a floor plan is more important than its shape/size. We use a topological map where nodes are regions and an edge between two nodes denotes the adjacency of corresponding regions. Nodes can be the intersection points of hallways, or between hallways and landmarks. A landmark's room is also a node. Such a topological map can be used to find the navigation route to a given destination.

To evaluate the topology of the reconstructed map, we extract and compare the topological maps from the grid maps for both ground truth and reconstructed floor plans. Fig. 20 shows the topological maps of the floor plan on story 1 produced by Jigsaw. We observe that its topological structure is the same as that of its ground truth (the same for storys 2 and 3).

7.5 Connection Area Detection

We collect WiFi CD (cosine distance) to evaluate how well it can detect whether a user stays on the same floor or goes to another floor. Fig. 21a shows the WiFi cosine distance for each 15 s sliding window with 1 s step for walking users. We observe that when users walk on the same floor, the majority (20-percentile to 80-percentile) of WiFi cosine distance is around $0.8 \sim 0.85$, and that of users passing stairs is much less ($0.65 \sim 0.75$), due to different APs on different floors.

Next, we examine stairs detection via WiFi CD and Acc COR features. Table 3 shows stairs detection accuracy of them individually and together. We observe that WiFi CD achieves better performance than Acc COR, and their combination significantly improves each alone, with 97.4 percent precision, 100 percent recall and 98.7 percent f-score values. Finally, we use Acc PV feature and a k-means algorithm to identify up/down stairs cases, with 100 percent detection accuracy.



(a) Cosine distance of WiFi signatures along 15s walking.

(b) Cosine distance of WiFi signatures for standing still.

Fig. 21. Cosine distance of WiFi signatures between start/end points: (a) along 15 s walking on the ground/stairs; (b) standing still on the ground/escalator/elevator.

TABLE 3
Evaluation of Stairs Detection

	Precision	Recall	F-score
WiFi CD	86.9%	91.3%	89.1%
Acc COR	78.2%	88.8%	83.2%
WiFi CD + Acc COR	97.4%	100.0%	98.7%

For escalator/elevator detection, we leverage WiFi CD and cellular SD features for the whole standing still period. They achieve 100 percent detection accuracy for both escalator and elevator. Then we use the difference of average acceleration along gravity direction of two 5s time windows at the beginning/end of the ride to identify up or down, achieving 92.9 percent accuracy for escalators, and 100 percent for elevators.

Finally, we use the refinement algorithm to correct residual detection errors, and achieve 100 percent detection accuracy for each connection area and each up/down case. The reconstructed and ground truth connection areas are shown on Figs. 19a, 19b and Figs. 19d, 19e.

7.6 Comparison with CrowdInside

We compare the reconstruction performance of Jigsaw to that of CrowdInside [11]. Jigsaw utilizes vision techniques and incurs more overhead in collecting and processing images, producing detailed positions and orientations of individual landmarks. CrowdInside is much lighter weight and uses mostly mobile traces. Its design is based on several assumptions: 1) sufficient numbers of anchor points (e.g., locations with GPS reception or special inertial data signature such as escalators/elevators/stairs) for calibrating traces; 2) sufficient amount of traces that pass through these anchor points; 3) distinctive WiFi signatures in different rooms.

In reality, we find that they may not always hold in all environments. For example, in story 2 there are only three inertial anchor points and no GPS reception; traces that do not pass through (e.g., start/end at) these three anchor points cannot be placed relative to other traces on the common ground plane; variations in WiFi signatures may cause incorrect room classification. As a result, the direct application of CrowdInside where these requirements do not hold may generate “unusable” floor plans that are hard to recognize.

To deal with these conditions, we make several artificial improvements to CrowdInside: 1) We double the number of anchor points and assume they are all GPS-based, thus more accurate global coordinates can be used to calibrate traces; 2) we make all traces pass through adjacent anchor points so they can be placed on the common floor plane; 3) we manually classify room traces so that their labeling is 100 percent correct. We call such an artificially improved version CrowdInside++.

The constructed floor plan by CrowdInside++ is shown in Fig. 22. The landmark positions of CrowdInside++ have an RMSE of 6.26 m, and the maximum error 8.92 m; the RMSE of intersections is 7.36 m and the maximum error is 9.84 m. All of these are four times larger than those of Jigsaw. We also notice that CrowdInside++ does not detect a few small-sized stores due to the ambiguity differentiating hallway and store traces, while Jigsaw uses images and can always detect such stores. The hallway shape of CrowdInside++ has a 48.2 percent recall, a 64.0 percent precision and a 55.0 percent F-score, which are much worse than those of Jigsaw shown in Table 2. The average error for room sizes is 42.7 percent, also much larger than that of Jigsaw. Note such performance is achieved after several artificial improvements which may not always be possible in reality.

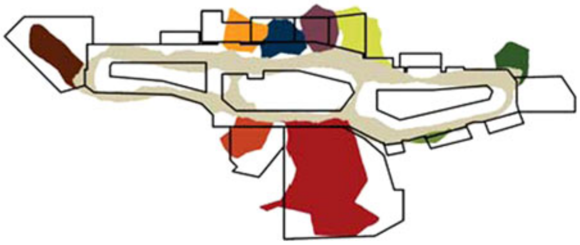


Fig. 22. Constructed floor plan of story 2 by CrowdInside++, which has several artificial improvements that are not always possible in reality.

The above shows that inertial-only approach cannot handle error accumulation well when there are not sufficient anchor points, while Jigsaw can use any camera location to calibrate traces. The landmark placement optimization and the probabilistic occupancy map also make Jigsaw much more robust to errors and outliers, whereas the deterministic alpha-shape in CrowdInside cannot tolerate outliers.

8 Discussion

Photo-taking operations involve more user efforts, and they provide more accurate geometry information of landmarks than inertial data. We have invited more than 30 users to collect data, and found that if a user is paid (e.g., $\sim$ \$20), he is willing to spend a few minutes practicing data collection following simple guidelines. We observe that they do exhibit more attention on gathering both images and inertial data after receiving the rewards, and their feedbacks suggest the guidelines are easy to follow in practice.

We have tried Jigsaw in other types of buildings where the environments are homogeneous, e.g., an office and a lab building. We find that all components in Jigsaw perform well except SfM, because it relies on abundant feature points matching among images. In office and lab, landmarks (e.g., doors on blank walls) have similar appearances and feature points are much less. Thus, SfM could not create the point cloud needed for landmark modeling. If we replace SfM with other techniques that do not rely on feature points, we could still create maps for those building types.

Our landmark model needs image classification so that images of the same landmark are used as input. With proper and sufficient incentives, users may be willing to tag photos to help ease the classification. There also has been study [26] on automated classification that can achieve high accuracy, and the scale-invariant features extracted by SIFT [17] gives robustness against image differences in resolution, orientation, and illumination conditions. Since there might be several landmarks with similar appearances, one future work is to combine image similarity with WiFi signatures and mobile trajectory to identify those landmarks.

Accurate user trajectories are shown quite challenges [5], [18] because inertial data is impacted by many factors such as the make/model, the position of device (e.g., in hand/pocket), the relative movement of human body (e.g., holding still versus swinging arms). Some of these may change during the user movements. In light of that, we assign a relatively lower confidence to such trajectories and use a probabilistic model when using them to build hallway and room occupancy maps.

The collection of image and inertial data consumes energy. Jigsaw uses downsized images of 800×600 resolution, each about 100 kB. We use Monsoon Power Monitor [27] (a standard tool for power measurement on a mobile device) to measure the energy cost of micro-tasks. Taking the largest micro-task in our experiments as an example, which consists of two photos taken at the start/end of 2-minute walking, it costs around 174 Joules. Based on WiFi radio transmission power of 720 mW and practical speed of 700 kB/s [28], uploading all data (~ 780 kB) in this micro-task costs 0.8 Joule. Compared to the battery capacity of 19 k Joules [29], the largest micro-task constitutes a mere 0.92 percent energy consumption.

9 Related Work

Floor plan construction. Indoor floor plan construction is a relatively new problem in mobile computing. A few pieces of work has conducted very valuable initial investigation, using mostly inertial and WiFi data. CrowdInside [11] leverages inertial data from accelerometer, gyroscope and compass to reconstruct users' mobile trajectories, and use "anchor points" with unique sensing data such as stairs and locations with GPS reception to correct accumulated errors. The trajectories serve as hints about accessible areas, from which hallways, rooms can be identified. Such "anchor" points are also used for user localization (e.g., Unloc [4]). Compared to it, we combine vision and mobile techniques of complementary strengths, extracting detailed geometry information about individual landmarks from images, while inferring the structure and shapes of the hallway and rooms from inertial data. We also use optimization and probabilistic techniques so that the results are robust to errors and outliers in crowdsensed data.

MapGenie [10] uses mobile trajectories as well but it leverages foot-mounted IMU (Inertail Measurement Unit) which is less affected by different positions of the phone, while we use smartphones which are more suitable for crowdsensing. Walkie-Markie [8] leverages WiFi signals, and it uses locations where the trend of WiFi signal strength reverses direction as anchor points, which are found to be more stable than signatures themselves. However, it only constructs the rough hallway skeleton while we also construct both hallways and rooms with accurate geometry and shape. SmartSLAM [9] utilizes WiFi signals and applies dynamic Bayesian network on the smartphones, and it also focuses on just hallway skeleton instead of complete floor plans. Jiang et. al. [7] propose a series of algorithms to detect similarities in WiFi signatures between different rooms and hallway segments to find their adjacency, and combine inertial data to obtain hallway lengths and orientations to construct floor plans. However, they manually associate WiFi fingerprints with each room ID, and assume regular building layouts (e.g., hallways with straight segments and right turns, rooms are adjoined and in rectangle shapes), while we automatically cluster landmarks without any manual intervention, and our occupancy grid map technique is robust to arbitrary building layouts.

SLAM. Learning maps in an unexplored environment is the famous SLAM (Simultaneous Localization And Mapping) problem in robotics [30]. One has to estimate the poses (2D/3D locations and orientations) of the robot, and locations of landmarks from robot control and environment measurement parameters. Various sensors such as odometry, depth/stereo cameras and laser rangers are used.

We share similar goals with SLAM, but our input and problem have significant differences. First, crowdsensed data is not just noisy, but also piece-wise, collected from mostly uncoordinated users. While in SLAM a robot usually has special high precision sensors (e.g., laser ranges, depth/stereo cameras) and systematically explores all accessible areas. We use commodity mobile devices which do not have such sensors; the mobile trajectories are also highly noisy due to error accumulation. Second, we estimate landmarks' orientations as well, while SLAM does only their locations. The existence of loops in the dependence relationship of measurements also adds to the complexity of our problem.

3D construction. There has been significant amount of literature for reconstructing the 3D model of buildings in computer vision. They take different approaches and require different kinds and amount of data. Some use laser ranger data to produce very detailed and accurate exterior models [31]. Indoor floor plan is essentially a 2D model and we realize that indiscriminate and uniform details are not necessary. This insight enables us to use vision techniques for individual landmarks only while using much lighter weight mobile techniques for landmark placement, hallway and rooms. This approach greatly reduces the effort and overhead for capturing and processing large amount of data (some of which may require special hardware such as laser rangers not available on commodity mobile devices), yet still generate reasonably complete and accurate floor plans.

Indoor localization. LiFS [6] leverages the user motion to construct the signature map and crowdsources its calibration to users. Zee [5] tracks inertial sensors in mobile devices carried by users while simultaneously performing WiFi scans. Multidimensional scaling technique [32] is also used to locate WiFi APs from radio scans, so as to build a positioning system without the floor plan. These admirable work produce the signature map, while we construct the floor plan with geometry and shape/sizes of indoor elements such as hallways and rooms. Furthermore, our reconstructed floor plans can be used as constraints to improve localization accuracy (as used in Zee [5] and VeTrack [33]).

Computer vision techniques have been used for localization as well. Sextant [34] leverages photos and gyroscope on smartphones to measure users' relative positions to physical objects, thus localizing users. We simply leverage the ability of SfM to compute the pose, thus the location of the camera taking the image.

10 Conclusion

In this paper, we propose Jigsaw, which combines vision and mobile techniques that take crowdsensed images and inertial data to produce multi-story floor plans for complex indoor environments. It addresses one fundamental obstacle to the ubiquitous coverage of indoor localization service:

lack of floor plans at service providers. Jigsaw enables service providers to reconstruct floor plans at scale from mobile users' data, thus avoiding the intensive efforts and time needed in business negotiations or environment surveys. We have presented the detailed design and conducted extensive experiments in three stories (two with irregular shapes) of two large shopping malls. The results demonstrate that Jigsaw can produce complete and accurate locations/orientations of landmarks, and structures/shapes of hallways, rooms and connection areas.

ACKNOWLEDGMENTS

This work is supported in part by China 973-2015CB351800, NSFC 61572051, and 61231010, and Beijing NSF 4142022.

REFERENCES

[1] (2016). Google indoor maps availability [Online]. Available: http://support.google.com/gmm/bin/answer.py?hl=en&answer=1685827

[2] 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. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 249–260.

[3] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[4] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 197–210.

[5] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.

[6] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 269–280.

[7] 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. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 315–324.

[8] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: Indoor pathway mapping made easy," in *Proc. 10th USENIX Conf. Netw. Syst. Des. Implementation*, 2013, pp. 85–98.

[9] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *IEEE Trans. Syst., Man, Cybern.*, vol. 42, no. 6, pp. 889–898, Nov. 2012.

[10] D. Philipp, P. Baier, C. Dibak, F. Drr, K. Rothermel, S. Becker, M. Peter, and D. Fritsch, "MapGENIE: Grammar-enhanced indoor map construction from crowd-sourced data," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2014, pp. 139–147.

[11] M. Alzantot and M. Youssef, "Crowdinside: Automatic construction of indoor floorplans," in *Proc. 20th Int. Conf. Adv. Geographic Inform. Syst.*, 2012, pp. 99–108.

[12] (2016). Gigwalk [Online]. Available: http://www.gigwalk.com

[13] (2016). Zaarly [Online]. Available: https://www.zaarly.com

[14] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 173–184.

[15] N. Snavely, I. Simon, M. Goesele, R. Szeliski, and M. Seitzs, "Scene reconstruction and visualization from community photo collections," *Proc. IEEE*, vol. 98, no. 8, pp. 1370–1390, Aug. 2010.

[16] D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 2136–2143.

[17] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.

[18] N. Roy, H. Wang, and R. R. Choudhury, "I am a smartphone and i can tell my users walking direction," in *Proc. Int. Conf. Mobile Syst. Appl. Serv.*, 2014, pp. 329–342.

[19] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[20] S. Huang, Y. Lai, U. Frese, and G. Dissanayake, "How far is SLAM from a linear least squares problem?" in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 3011–3016.

[21] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, no. 1/2, pp. 83–97, 1955.

[22] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Auton. Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[23] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[24] (2016). Cosine distance [Online]. Available: https://en.wikipedia.org/wiki/Cosine_similarity

[25] (2016). K-means clustering [Online]. Available: https://en.wikipedia.org/wiki/K-means_clustering

[26] S. Wang, J. Joo, Y. Wang, and S. C. Zhu, "Weakly supervised learning for attribute localization in outdoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3111–3118.

[27] (2016). Monsoon power monitor [Online]. Available: https://www.msoon.com/LabEquipment/PowerMonitor

[28] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Conf. USENIX Annu. Techn. Conf.*, 2010, pp. 21–34.

[29] (2016). iphone 4s spec [Online]. Available: https://en.wikipedia.org/wiki/IPhone_4S

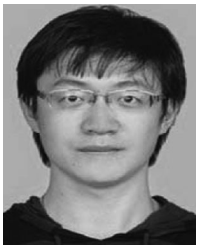
[30] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[31] C. A. Vanegas, D. Aliaga, and B. Benes, "Automatic extraction of Manhattan-world building masses from 3D laser range scans," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 10, pp. 1627–1637, Oct. 2012.

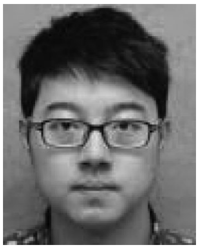
[32] J. Koo and H. Cha, "Autonomous construction of a wifi access point map using multidimensional scaling," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2011, pp. 115–132.

[33] M. Zhao, T. Ye, R. Gao, F. Ye, Y. Wang, and G. Luo, "Vetrack: Real time vehicle tracking in uninstrumented indoor environments," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 99–112.

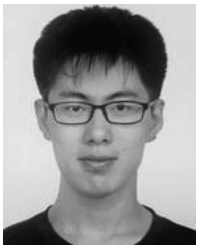
[34] Y. Tian, R. Gao, K. Bian, F. Ye, T. Wang, Y. Wang, and X. Li, "Towards ubiquitous indoor localization service leveraging environmental physical features," in *Proc. IEEE INFOCOM*, 2014, pp. 55–63.



Ruipeng Gao received the BE degree in communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010. He is currently working toward the PhD degree in computer science from Peking University, Beijing, China. His research interests include wireless communication and mobile computing.



Mingmin Zhao received the BE degree in computer science from Peking University in 2015. He is currently working toward the PhD degree in computer science at the Massachusetts Institute of Technology, Cambridge, MA. His research interests include mobile computing, wireless networking, and machine learning.



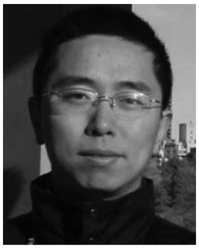
Tao Ye is currently working toward the BS degree in computer science at Peking University, Beijing, China. His research interests include mobile computing and artificial intelligence.

Fan Ye received the MS and BE degrees from Tsinghua University and the PhD from the Computer Science Department at the University of California, Los Angeles, CA. He is an assistant professor in the ECE Department, Stony Brook University. He has published more than 60 peer reviewed papers that have received more than 7,000 citations according to Google Scholar. He has 21 granted/pending US and international patents/applications. His research interests include mobile sensing platforms, systems and applications, Internet-of-Things, indoor location sensing, and wireless and sensor networks.

Guojie Luo (M'12) received the BS degree in computer science from Peking University, Beijing, China, in 2005, and the MS and PhD degrees in computer science from University of California, Los Angeles, CA, in 2008 and 2011, respectively. He is currently an assistant professor at Peking University. His research interests include physical design automation, scalable EDA algorithms, and advanced design technologies for 3D ICs. He obtained the 2013 ACM SIGDA Outstanding PhD Dissertation Award in electronic design automation. He is a member of the IEEE.

Yizhou Wang received the bachelor's degree in electrical engineering from Tsinghua University in 1996, and the PhD degree in computer science from the University of California, Los Angeles, CA, in 2005. He is a professor in the Computer Science Department, Peking University, Beijing, China. His research interests include computational vision, statistical modeling and learning, pattern analysis, and digital visual arts.

Kaigui Bian (M'11) received the PhD degree in computer engineering from Virginia Tech, Blacksburg, VA, in 2011. He is currently an associate professor with the Institute of Network Computing and Information Systems, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include mobile computing, cognitive radio networks, and network security and privacy. He is a member of the IEEE.



Tao Wang (SM'11) received the PhD degree in computer science from Peking University, Beijing, China, in 2006. He is currently an associate professor at Peking University, Beijing, China. His research interests include computer architecture, reconfigurable logic, wireless network architecture, and mobile cloud computing. He is a senior member of the IEEE.

Xiaoming Li (SM'03) received the doctoral degree in computer science from Stevens Institute of Technology, Hoboken, NJ, in 1986. He is currently a professor at Peking University, Beijing, China. His research interests include web search and mining and online social network analysis. He is an international editor of concurrency and computation. He is a senior member of the IEEE.