Jigsaw: Indoor Floor Plan Reconstruction via Mobile Crowdsensing *

Ruipeng Gao[1], Mingmin Zhao[1], Tao Ye[1], Fan Ye[1,2], Yizhou Wang[1],
Kaigui Bian[1], Tao Wang[1], Xiaoming Li[1]
EECS School, Peking University, Beijing 100871, China[1]
ECE Department, Stony Brook University, Stony Brook, NY 11794, USA[2]
{gaoruipeng, zhaomingmin, pkuleonye, yefan[1], yizhou.wang, bkg, wangtao, lxm}@pku.edu.cn,
fan.ye@stonybrook.edu[2]

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. Our experiments on 3 stories of 2 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 is 100% correct.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

Keywords

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 [2], 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*, 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. This opens up the possibility of fast and scalable floor plan reconstruction.

The concept of mobile crowdsensing [12] is more and more popular. Recent work has used crowdsensed data to localize users [34] and reduce the calibration efforts of WiFi signatures [23, 38]. Among others [14, 22, 26, 27], CrowdInside [3] 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 POIs (e.g., 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. Thus 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

*This work is done at Peking University and supported in part by China NSFC Grants 61370056, 61231010, 61121002, 61201245, 61272340 and 973 Grant 2014CB340400.

MobiCom'14, September 7-11, 2014, Maui, Hawaii, USA.

http://dx.doi.org/10.1145/2639108.2639134.

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 in a common floor plane. Then user mobility traces from another type of micro-task are used to obtain the hallway connectivity, orientation and room shapes/sizes, using combinatorial optimization and probabilistic occupancy techniques.

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. This is 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. Thus 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 in 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 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% correct isle topology connectivity, 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.

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

2. DESIGN OVERVIEW

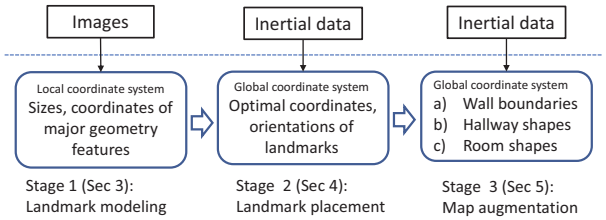


Figure 1: Jigsaw contains 3 stages: landmark modeling, landmark placement and map augmentation. Each stages uses image or inertial data and output from the previous stage.

Jigsaw utilizes images, acceleration and gyroscope data. The reconstruction consists of three stages: landmark modeling, placement and augmentation (Figure 1). First, two computer vision techniques, Structure from Motion (SfM) [29] and vanishing line detection [18], 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). Finally, 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 sizes/shapes from inertial user traces (Section 5).

Different from opportunistic data gathering adopted in most existing work [3, 23, 34, 38], we assume the users proactively take a bit effort to conduct different data-gathering *micro-tasks*. Each micro-task defines one or a few actions to gather different data in certain spatial areas and temporal durations. Examples include: 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 while taking a photo immediately before/after the walk. Such micro-tasks allow us to gather data useful in specific stages. We assume service providers have certain incentive mechanisms [37] to reward users of their efforts, and we do not consider intentional spam in this work.

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

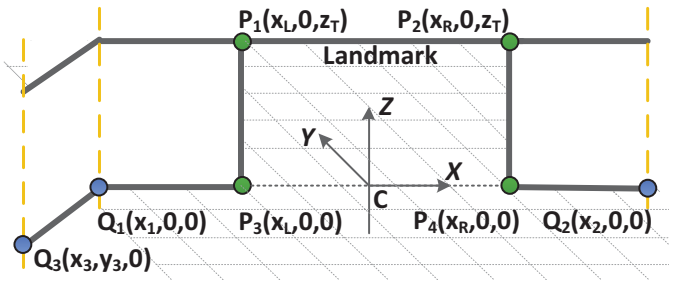


Figure 2: The model of this exemplary store entrance has 4 geometric vertices $P_1 \sim P_4$ and 3 connecting points of wall segments $Q_1 \sim Q_3$ in its local coordinate system.

We use a very simple model to describe the major geometry features of a landmark. As illustrated in Figure 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) [29] and vanishing line detection [18], to obtain the coordinates of P, Q from landmark images.

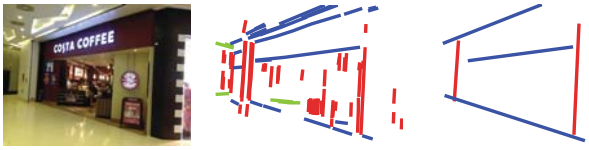
Structure from Motion is a mature computer vision technique commonly used to construct the 3D models of an object. Given a set of images of the same object (e.g., a building) from different viewpoints, it produces: 1) a "point cloud" consisting of many points in a local 3D coordinate system. Each point 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.

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 [18] to produce line segments for each image of the same landmark (Figure 3b). We merge co-linear

[1]To be more exact, each point represents a "feature point" as detected by certain feature extractor algorithms (e.g., [4, 20]).



(a) (b) (c)

Figure 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.

and parallel segments close to each other into long line segments (Figure 3b). This 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 [29]. 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 Figure 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 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 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 Figure 2, marked as two diamonds in Figure 4), then find the two ends (e.g., marked as two crosses in Figure 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. Figure 4 shows three wall connecting points discovered.

3.4 Example

Figure 4 shows the point cloud of one store entrance projected onto the floor plane and SfM produced camera

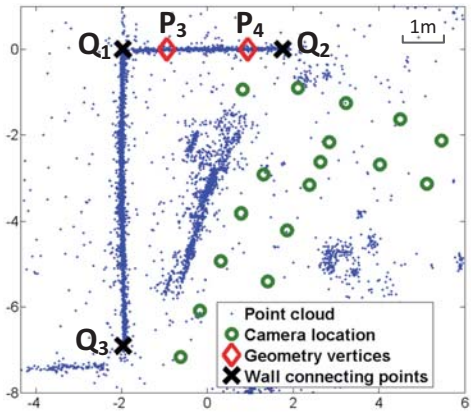


Figure 4: A landmark's point cloud projected to the floor plan, with camera locations, critical contour line (P_3 and P_4) and connecting points of wall segments (Q_1 , Q_2 and Q_3).

locations. We mark the geometry vertices (diamonds) and wall connecting points (crosses). In this example, the width of the entrance has an error of 0.086m (4.78% of the true width 1.8m). We also detect two external wall segments along the hallway, and their intersection angle error is 0.08° out of 90° (0.09%). We find that the 176 camera locations produced by SfM (only some of them are shown) are quite accurate. The localization error is within $1.2m$ at 90% percentile, and maximum error is $1.5m$. 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%) images are localized, with 90% error of $1.8m$ and maximum error of $5.2m$. We will present more systematic evaluation results in Section 6.

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.

4.1 Notations

Suppose there are n local coordinate systems corresponding to n landmarks $l_1, l_2, ..., l_n$. $X_i = (x_i, y_i) \in \mathbb{R}^2$ 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 $\boldsymbol{X} = (X_1, ..., X_n)$, $\boldsymbol{\phi} = (\phi_1, ..., \phi_n)$.

$R_i = R(\phi_i) = \begin{bmatrix} cos\phi_i & -sin\phi_i \\ sin\phi_i & cos\phi_i \end{bmatrix}$ is the rotation matrix used in coordinate transformation between the global and local coordinate systems of landmark l_i. $X_j^i = (x_j^i, y_j^i) = R(\phi_i)^T(X_j - X_i)$ and $\phi_j^i = \phi_j - \phi_i$ are the x-y coordinates and orientation of landmark l_j in the local coordinate system of landmark l_i, respectively.

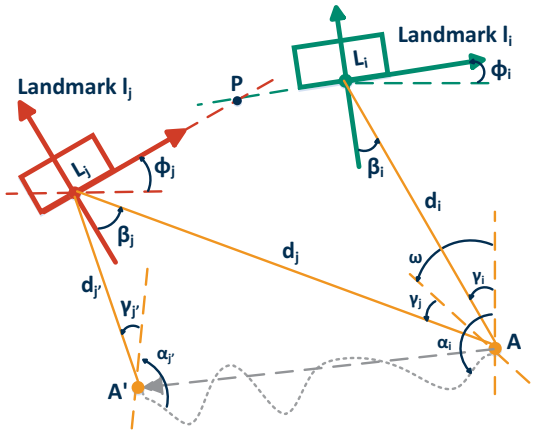


Figure 5: Click-Rotate-Click and Click-Walk-Click: A is where two photos of landmark l_i and l_j are taken in CRC. (d_i, β_i, γ_i) are the length of AL_i, angle formed by line AL_i and normal direction of L_i, angle formed by line AL_i and direction of camera, respectively. P is the intersection point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

4.2 Spatial Relation Acquisition

The spatial relationship between two adjacent landmarks l_i, l_j are X_j^i and ϕ_j^i, the coordinates and orientation of landmark l_j in the local coordinate system of landmark l_i (or vice versa, illustrated in Figure 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 Figure 5), then spins the body and camera for a certain 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 [23, 34].

(d_i, β_i, γ_i) represents the distance between camera A and landmark l_i, angle formed by line L_iA and the normal line of landmark l_i, angle formed by line L_iA and the direction of camera A, respectively. They can be derived from the camera pose (i.e., coordinates and orientation in $l_i's$ location coordinate system) as produced by SfM (Section 3). Similar is (d_j, β_j, γ_j). P represents the intersection point of the two x-axes in the two landmarks' local coordinate systems.

From plane geometry, quadrangle AL_iPL_j is uniquely determined given $(d_i, \beta_i, \gamma_i), (d_j, \beta_j, \gamma_j)$ 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. This 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 [23], 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 [24] 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 figure 5.

The two camera locations in CWC can be used as "anchor points" to calibrate the trace. Due to well-known error accumulation [3] 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(Figure 6) describing the conditional dependence structure among variables (denoted as nodes), where each variable only directly depends on its predecessors.

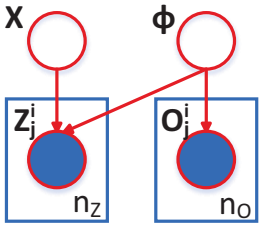


Figure 6: Bayesian belief network representation of our problem. X is the coordinates while ϕ is the orientations of all the landmarks. $\boldsymbol{\theta} = (\boldsymbol{X}, \boldsymbol{\phi})$ is the hidden variable we need to estimate based on measurements. Z_j^i, O_j^i measures the coordinates and 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.

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

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

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

As is standard in probabilistic mapping literature [7], we assume Gaussian measurement models that give further transformation into:

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

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:

$$\underset{\boldsymbol{\phi}, \boldsymbol{X}}{\text{minimize}} \quad \sum_{O_j^i} \|\phi_j^i - O_j^i\|^2 + \sum_{Z_j^i} \|X_j^i - Z_j^i\|^2 \quad (2)$$

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

4.4 Optimization Algorithm

Let's denote problem (2) as:

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

since the two terms in (2) are functions of $\boldsymbol{\phi}$ and $(\boldsymbol{\phi}, \boldsymbol{X})$.

Careful examination [13] shows that each term in $g(\boldsymbol{\phi}, \boldsymbol{X})$ is linear square of $\boldsymbol{X}$, thus $g(\boldsymbol{\phi}, \boldsymbol{X})$ is a typical linear least squares of $\boldsymbol{X}$ with a closed form solution. We denote the the minimum as $h(\boldsymbol{\phi})$. Thus problem (3) is equivalent to:

$$\underset{\boldsymbol{\phi}}{\text{minimize}} \quad f(\boldsymbol{\phi}) + h(\boldsymbol{\phi}) \quad (4)$$

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

STEP 1: Find $\boldsymbol{\phi'}$ given measurements $\boldsymbol{O}$.

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

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 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 in the orientation dependence graph where edges are relationship observations between landmarks. This problem $f_{\boldsymbol{MST}}(\boldsymbol{\phi})$ can be easily solved because adjusting the orientation of one landmark has a one-way effects 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

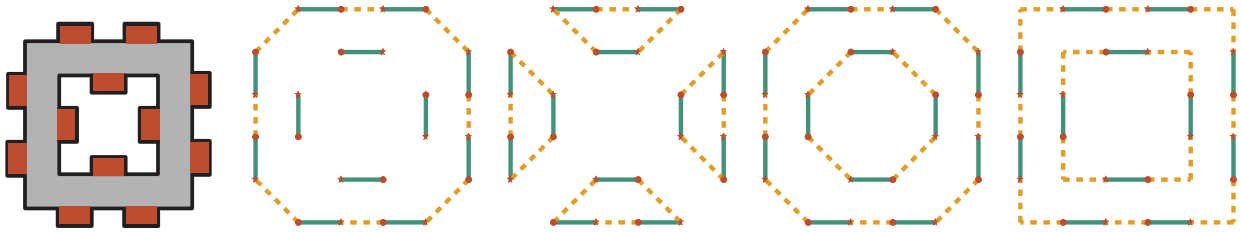


Figure 7: Comparison between different algorithms: (a) Example scenario (b) convex hull of all wall segments (c) one possible output of the greedy method (d) minimal weight matching using distance as weight (e) our minimal weight matching method.

be in near neighborhood of the true optimum ϕ'. Then we perform gradient descent from ϕ'_{MST} find a minimum likely to be ϕ'. In reality we do find them are usually in close neighborhood.

STEP 2: perform stochastic gradient descent (SGD) from ϕ' to find ϕ^*. Based on the intuition explained earlier that ϕ' is close to ϕ^*, 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 produces an external boundary but may not connect those segments "inside" the hull (Figure 7(b)).

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 Figure 8). Thus k wall segments have two sets of endpoints $\boldsymbol{L} = \{L_1, L_2, ..., L_k\}$ and $\boldsymbol{R} = \{R_1, R_2, ..., R_k\}$. We need to add new wall segments connecting each endpoint in $\boldsymbol{L}$ to one in $\boldsymbol{R}$ (shown in Figure 8).

Every possible solution corresponds to a perfect matching π, where π is a permutation of $(1, 2, ..., k)$, indicating $L(i)$ and $R(\pi(i))$ are linked for $i = 1, 2, ..., k$. 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° corners commonly seen in buildings may be missing. E.g., Figure 7(c) and 7(d) show two possible results, where one is incorrect while the other does not have 90° corners.

To address the above issues, we consider the two following options of linking two adjacent wall segments. Each option carries a weight, which can be computed given two endpoints

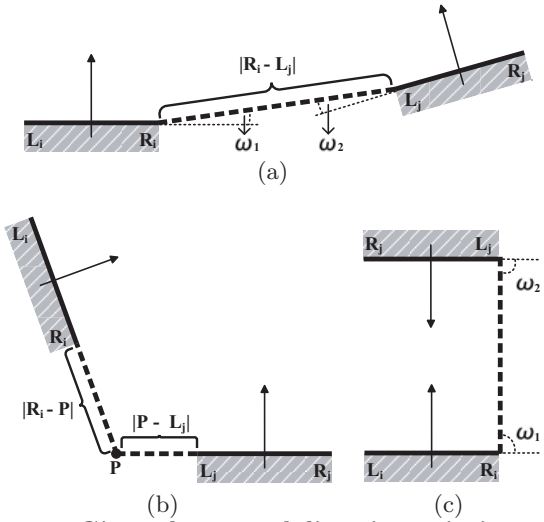


Figure 8: Given the normal direction pointing to the hallway, two endpoints of a wall segment are labeled L and R. New wall segments must link endpoints with different labels. Three cases of connection are shown: (a) two nearly collinear segments; (b) two nearly perpendicular segments; and (c) two nearly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

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:

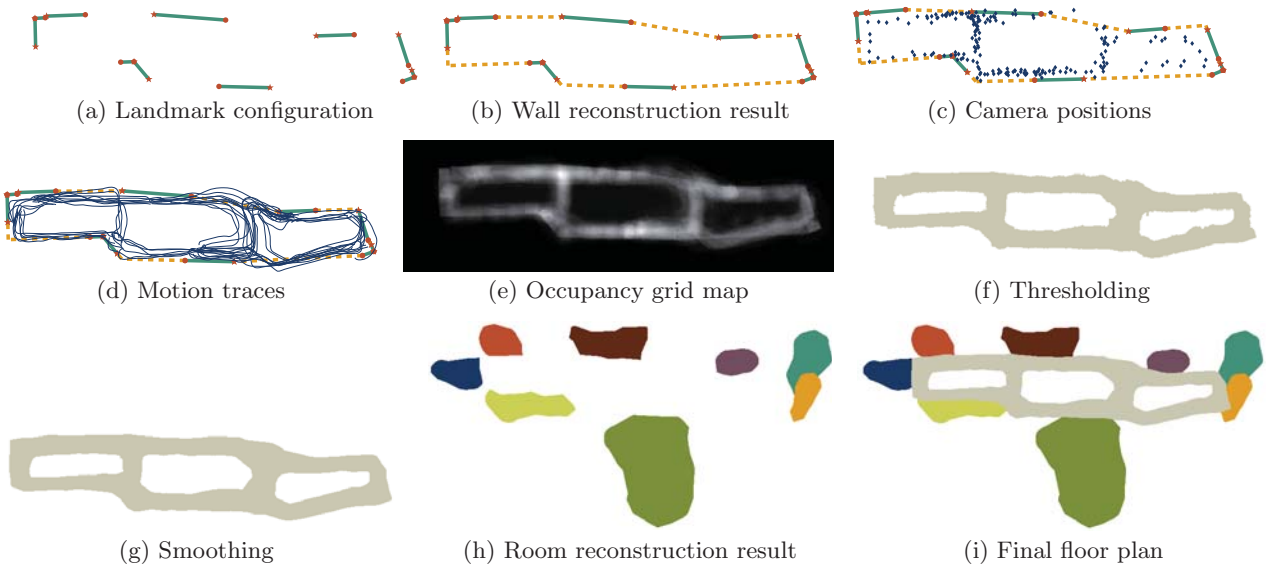
$$w_{ij}^{(1)} = |R_i - L_j|(\omega_1 + \omega_2) \tag{6}$$

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 Figure 8(a) and 8(c)). 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 Figure 8(b)). For two (close to) perpendicular segments, the



(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

Figure 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).

above equation produces a smaller weight, ensuring proper connection for 90° 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:

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

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:

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

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 effectively by Kuhn-Munkres algorithm [16] 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). Figure 7(e) shows the correct result produced by our algorithm and Figure 9(b) 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 [30], 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 reconstructed 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 are often the entrance, it is always accessible and we assign higher confidence. Places in front of a newly added wall is 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 relative position between cameras and landmarks. Such positions are obviously accessible. So we add confidence to places around every camera's position. Figure 9(c) 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 [23, 24]. 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 main source of error in 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. Figure 9(d) depicts motion traces in the hallway with the result of wall reconstruction.

The final occupancy grid map is shown in Figure 9(e). We use an automatic threshold based binaryzation technique [21] 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 accessibly only when there is enough evidence. The result of thresholding is depicted in Figure 9(f). To further improve the result, we implement a smoothing algorithm based on

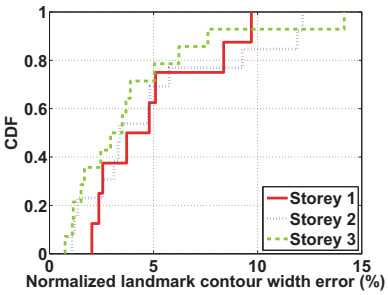


Figure 10: Normalized store entrance width error.

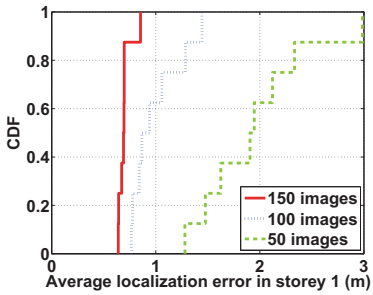


Figure 11: Impact of image quantities

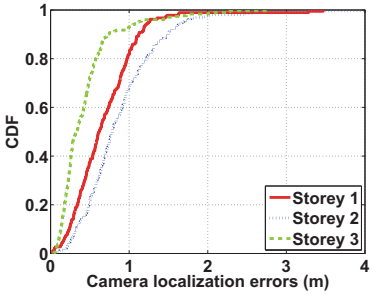


Figure 12: Image localization errors.

alpha-shape [3], which is a generalization of the concept of convex hull. Figure 9(g) 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. Thus the places inside the detected wall segments are part of the room with a 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 Figure 9(h). The final floor plan at the end of map augmentation is in Figure 9(i).

6. PERFORMANCE EVALUATION

6.1 Methodology

We use iPhone 4s to collect images and motion sensor data in three environments: two storeys of a $150 \times 75m$ shopping mall (labeled storey 1 and 2) of irregular shape, and one storey of a $140 \times 40m$ long and narrow mall comprised of two parts connected by two long corridors (labeled part I and II of storey 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" to take two images of two nearby landmarks, and 24 "Click-Walk-Click" to take two images of two far away landmarks in different parts in storey 3. We also collect 96, 106 and 73 user traces along the hallway of each environment, and about 7 traces inside each store.

6.2 Landmark Modeling

First we examine the accuracy of estimated widths of store entrances using normalized error (error divided by the true width). As Figure 10 shows, 90-percentile error is about 10%, 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% in three indoor environments, respectively; while the precision (i.e., fraction of detected ones that are correct) are all 100%. This shows 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° 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 increase steadily from 74.5%, 95% to 99.3%. When there are sufficient images (e.g., $100 \sim 150$), nearly all of them are localized.

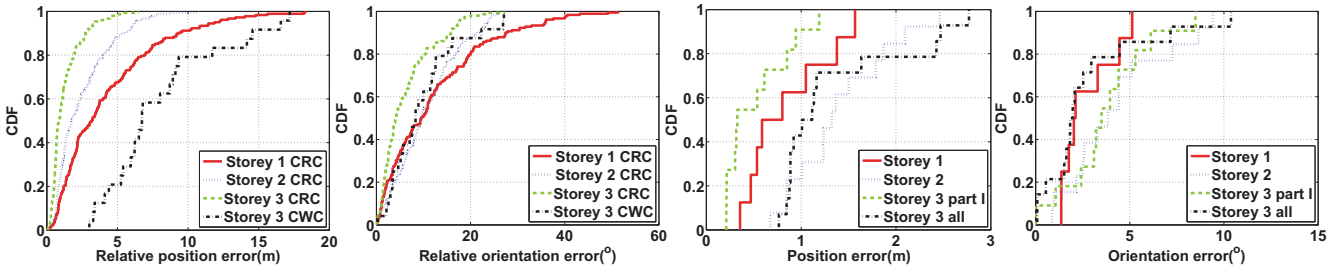
We also observe similar trend in the average localization error for each landmark in storey 1 (shown in Figure 11; the other two are similar). When there are sufficient images, the average error is less than 2m. Thus $100 \sim 150$ images for each landmark would be an appropriate amount.

Finally we examine image localization accuracy (shown in Figure 12). We observe that localization errors are about $1 \sim 1.5m$ 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 storey 3 has the least error, due to its smaller size so images are distributed more densely and thus appear similar to each other.

6.3 Landmark Placement

Measurements accuracy. We first evaluate the relative position and orientation errors as derived from pairwise measurements (Section 4.2) between adjacent landmarks. Relative position error is the distance between a landmark's derived position and its true 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 storey 3 between its two parts. Figure 13(a) and 13(b) 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 $10 \sim 20°$, both of which have quite some inaccuracies. CWC measurements have worse position errors (80-percentile around 10m) but comparable



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

Figure 13: 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; (d) orientation error of proposed algorithm for landmark level mapping.

orientation errors (80-percentile at 15°). 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 respective ground truth to examine errors in the derived configuration. Figure 13(c) and 13(d) show the CDFs of position and orientation errors. Since CRC measurements alone cannot join the two parts of storey 3, we give CRC accuracy for part I (containing most stores). Compared with respective errors in measurements (shown in Figure 13(a) and 13(b)), both position and orientation errors improve (e.g., $1 \sim 2m$ and $5 \sim 9°$ 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., 2.5m at 90-percentile) due to relatively 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.

6.4 Floor plan performance

The reconstructed floor plans and their respective ground truths are shown in Figure 14(a), 14(b), 14(c) and Figure 14(d), 14(e), 14(f).

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 $X_i^{test} = (x_i^{test}, y_i^{test})$, $i = 1, 2, ..., n$, 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 storey 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 accurate CWC data.

Table 1: RMSE of floor plans (m)

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

Hallway shape. We also evaluate how close the shapes of constructed hallways resemble 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%, a recall around 90% and a F-score around 84% for the first two storeys. This shows the effect of the calibration of traces by camera locations, and probabilistic occupancy maps that are more robust to errors and outliers. The reason that recalls are higher than precisions (as shown in Figure 14) is that reconstructed hallway is a little thicker than the ground truth due to errors in traces. Storey 3 has a relative lower performance because only CWC data can be used to connect the left and right parts.

Table 2: Evaluation of hallway shape

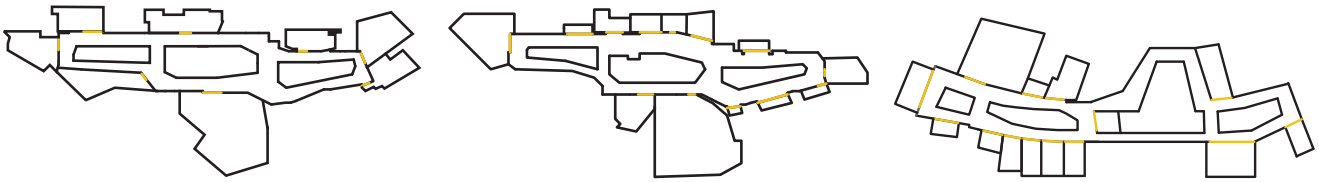
	Precision	Recall	F-score
Storey 1	77.8%	92.0%	84.3%
Storey 2	81.2%	93.3%	86.9%
Storey 3	74.5%	86.0%	79.8%

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% respectively for 3 storeys. 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.

6.5 Comparison with CrowdInside

We compare the reconstruction performance of Jigsaw to that of CrowdInside [3]. 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

(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

Figure 14: Reconstructed floor plans and ground truth floor plans

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 storey 2 there are only 3 inertial anchor points and no GPS reception; traces that do not pass through (e.g., start/end at) these 3 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 is hard to recognize.

Figure 15: Constructed floor plan of storey 2 by CrowdInside++, which has several artificial improvements that are not always possible in reality.

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% correct. We call such an artificially improved version CrowdInside++.

The landmark positions of CrowdInside++ have an RMSE of 6.26m, and the maximum error 8.92m; the RMSE of intersections is 7.36m and maximum error is 9.84m. 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% recall, a 64.0% precision and a 55.0% F-score, which are much worse than those of Jigsaw shown in Table 2. The average error for room sizes is 42.7%, also much larger than that of Jigsaw. Note such performance is achieved after several artificial improvements which may not always be 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.

7. DISCUSSION

In practise, the performance of SfM may be affected by multiple factors, some of which have been studied in literature. When the number of images is insufficient, the point cloud may not be accurate enough. There has been work [8] that guides the selection of images such that a small quantity can still produce reasonably good output. Such technique may help improve the performance on camera localization. The scale-invariant features extracted by SIFT [20] gives SfM robustness against differences in resolution, orientation, and certain illumination conditions (e.g., noon vs. morning sunlight). Random variations such as moving customer flows, obstructions such as pillars constitute disturbances to the appearances of landmarks. Thus more images of stable features would be needed to offset the impact of such disturbances.

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 [35] on automated classification that can achieve high accuracy. Our landmark model is quite simple and most suitable for store or room entrances on a frontal wall. Although this captures a great portion of indoor landmarks, there are others (e.g., a Christmas tree) that do not fit this model, and more advanced model would be needed.

Accurate user trajectories are shown quite a challenge [23, 24] because inertial data is impacted by many factors such as the make/model, position of the device (e.g., in hand/pocket), relative movement to the human body (e.g., holding still vs. 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 takes some energy. The power consumption for accelerometer and gyroscope sampling is pretty low (e.g., about 30mW [1]), while people customarily take tens of or more photos during a trip. Jigsaw uses downsized images of 800×600 resolution, each about 100kB. Based on WiFi radio transmission power around 700mW [5] and practical speed of 2MB/s, uploading 20 photos would cost about 0.7 Joule. Compared to the battery capacity of 20k Joules, we believe the capturing and uploading of a few tens images, and collecting a few traces, do not constitute any signification power consumption for an user. The photos may contain customer faces, which may need to be blurred (e.g., like Google street view) for privacy. On the other hand, we find that store owners welcome such exposure because they view the appearance of their logos on maps as a form of advertisement.

8. 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 [3] leverages inertial data from accelerometer, gyroscope and compass to reconstruct users' mobile trajectories, and use "anchor points" with unique sensing data such as elevators, stairs, escalators 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 [34]).

Jiang et. al. [14] 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. Walkie-Markie [26] also leverages WiFi signals, but 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. MapGenie [22] uses mobile trajectories as well. But instead of the sensors in smartphones, it leverages foot-mounted IMU (Inertail Measurement Unit) which is less affected by different positions of the phone.

Compared to the above, 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.

SLAM. Learning maps in an unexplored environment is the famous SLAM (Simultaneous Localization And Mapping) problem in robotics [10, 11]. 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, gyroscope, 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/stearo 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 of them are interactive [6,28] and need continuous user intervention; some are automatic [9,36] and exploit prior knowledge about the relations among building primitives (e.g., walls, doors and windows); some take advantage of laser ranger data to produce very detailed and accurate exterior models [33].

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.

Vision-based localization. Computer vision techniques have been used for localization as well. Some work [15,17,32] compares a test image against a database of pre-captured benchmark images, finds the "closest" match and uses its location. Sextant [31] leverages photos and gyroscope on smartphones to measure users' relative positions to physical objects, thus localizing users. Some first reconstruct 3D model of the scene using depth information (e.g., from laser rangers) or by crawling large numbers of Internet images, and then computes the most likely location of an image against the 3D scene model [19, 25]. We simply leverage the ability of SfM to compute the pose, thus the location of the camera taking the image.

9. CONCLUSION

In this paper, we propose Jigsaw, which combines vision and mobile techniques that take crowdsensed image and inertial data to produce floor plans for complex indoor environments. It addresses one key 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 storeys (two with irregular shapes) of two large shopping malls. The results demonstrate that Jigsaw can produce reasonably complete and accurate locations/orientations of landmarks, and structures/shapes of hallways and rooms.

10. REFERENCES

[1] Accelerometer, Gyro and IMU Buying Guide. `https://www.sparkfun.com/pages/accel_gyro_guide`.

[2] Google indoor maps availability. `http://support.google.com/gmm/bin/answer.py?hl=en\&answer=1685827`.

[3] M. Alzantot and M. Youssef. Crowdinside: Automatic construction of indoor floorplans. In *SIGSPATIAL*, 2012.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *Computer Vision and Image Understanding*, 2008.

[5] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *Proceedings of USENIX Annual Technical Conference*, 2010.

[6] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *SIGGRAPH*, 1996.

[7] F. Dellaert and M. Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[8] J. Delon and B. Rouge. Small baseline stereovision. *Journal of Mathematical Imaging and Vision*, 2007.

[9] A. Dick, P. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, pages 111–134, November 2004.

[10] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part I the essential algorithms. *Robotics and Automation Magazine*, 2006.

[11] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part II state of the art. *Robotics and Automation Magazine*, 2006.

[12] R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. *IEEE Communication Magzine*, 2011.

[13] S. Huang, Y. Lai, U. Frese, and G. Dissanayake. How far is slam from a linear least squares problem? In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3011–3016. IEEE, 2010.

[14] 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 *Ubicomp*, 2013.

[15] J. Kosecka, L. Zhou, P. Barber, and Z. Duric. Qualitative image based localization in indoors environments. In *IEEE CVPR*, 2003.

[16] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[17] P. Lamon, I. R. Nourbakhsh, B. Jensen, and R. Siegwart. Deriving and matching image fingerprint sequences for mobile robot localization. In *IEEE International Conference on Robotics and Automation*, 2001.

[18] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE CVPR*, 2009.

[19] J. Z. Liang, N. Corso, E. Turner, and A. Zakhor. Image based localization in indoor environments. In *IEEE COM.Geo*, 2013.

[20] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[21] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[22] D. Philipp, P. Baier, C. Dibak, F. DÍźrr, K. Rothermel, S. Becker, M. Peter, and D. Fritsch. Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data. In *Percom*, 2014.

[23] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Mobicom*, pages 293–304, 2012.

[24] N. Roy, H. Wang, and R. R. Choudhury. I am a smartphone and i can tell my users walking direction. In *Mobisys*, 2014.

[25] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *13th IEEE International Conference on Computer Vision(ICCV)*, 2011.

[26] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: Indoor pathway mapping made easy. In *NSDI*, 2013.

[27] H. Shin, Y. Chon, and H. Cha. Unsupervised construction of an indoor floor plan using a smartphone. *IEEE Transactions on Systems, Man, and Cybernetics*, 2012.

[28] S. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Polle-Feys. Interactive 3d architectural modeling from unordered photo collections. *ACM Transactions on Graphics*, 2008.

[29] N. Snavely, I. Simon, M. Goesele, R. Szeliski, and M. Seitzs. Scene reconstruction and visualization from community photo collections. *Proceedings of the IEEE*, 1998.

[30] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003.

[31] 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 *IEEE INFOCOM*, 2014.

[32] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *IEEE International Conference on Robotics and Automation*, 2000.

[33] C. A. Vanegas, D. Aliaga, and B. Benes. Automatic extraction of manhattan-world building masses from 3d laser range scans. *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[34] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *MobiSys*, 2012.

[35] S. Wang, J. Joo, Y. Wang, and S. C. Zhu. Weakly supervised learning for attribute localization in outdoor scenes. In *CVPR*, 2013.

[36] J. Xiao, T. Fang, P. Zhao, M. Lhuillier, and L. Quan. Image-based street-side city modeling. *ACM Transactions on Graphics*, pages 273–289, 2009.

[37] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *MobiCom*, 2012.

[38] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Mobicom*, pages 269–280, 2012.