CrowdMap: Accurate Reconstruction of Indoor Floor Plans from Crowdsourced Sensor-Rich Videos

Si Chen, Muyuan Li, Kui Ren, Chunming Qiao
Computer Science and Engineering Department
SUNY at Buffalo
Email: {schen23, muyuanli, kuiren, qiao}@buffalo.edu

***Abstract*—Lack of an accurate and low-cost method to reconstruct indoor maps is the main reason behind the current sporadic availability of digital building floor plans. The conventional approach using professional equipment is very costly and only available in the most popular areas. In this paper, we propose and demonstrate CrowdMap, a crowdsourcing system utilizing sensor-rich video data from mobile users for indoor floor plan reconstruction with low-cost. The key idea of CrowdMap is to first jointly leverage crowdsourced sensory and video data to track user movements, then use the inferred user motion traces and context of the image to produce an accurate floor plan. In particular, we exploit the sequential relationship between each consecutive frame abstracted from the video to improve system performance. Our experiments in three college buildings show that CrowdMap achieves a precision of hallway shape around 88%, a recall around 93% and a F-measure around 90%. In addition, we achieve on average 9.8% room area error and on average 6.5% room aspect ratio error. The evaluation result demonstrates a significant improvement of accuracy compared with other crowdsourcing floor plan reconstruction systems.**

I. Introduction

A building floor plan succinctly illustrates spatial correlations of rooms, hallways and other features of the architecture from a top-down view over a floor. It plays an essential role in many indoor mobile applications, such as localization and navigation [1]–[3]. However, unlike outdoor environment, acquiring digital indoor floor plan information is very challenging. The state-of-the-art Google Indoor Maps [4] only have 10,000 locations available on-line, which is not in a position to compete with the total number of indoor environments around the world. The complexity of the indoor environment is the major obstacle to achieve ubiquitous coverage. Existing centralized collection and on-site calibration techniques demand professional devices and multi-party coordination, which are time consuming, inconvenient and costly.

Recently, the wide availability of smartphones and wearable devices (e.g. google glasses) equipped with built-in visual and inertial sensors makes the crowd easier than ever to devote themselves to crowdsourcing. Following this trend, there have been several studies trying to explore the possibility of using crowdsourced data to generate an indoor floor plan automatically. Among others [5]–[10], CrowdInside [10] first utilizes crowdsourced inertial sensory data to automatically construct user motion traces, and then aggregate the information to reconstruct indoor pathways. Jigsaw [11] takes one step further as their method leverages both image and inertial data to reconstruct an indoor floor plan.

However, current crowdsourcing floor plan reconstruction systems are not able to produce accurate enough results. This is partially due to the fact that most of existing indoor floor plan reconstruction systems heavily rely on sensory data [6], [7], [10], [11], which is only able to provide users' moving information for an unknown indoor space. For instance, CrowdInside [10], Walkie-Markie [6], Jigsaw [11] and the work in [7] all primarily depend on aggregated user motion traces derived from inertial data to determine the shape of hallway and room. The premise of their work is that users would be able to move across all edges and corners in an indoor environment. Due to the fact that the edge of an indoor scene is usually blocked by furniture or other objects, that assumption, however, may not be realistic in practice when reconstructing a complex indoor environment like rooms. Moreover, some restricted areas in an indoor environment are also inaccessible for most of the users, which may lead to some significant errors for the crowdsourced results. Unlike the sensory information, visual information should preserve more context information for an unknown indoor environment, such as the geometric information, color information, lighting conditions and text information. Therefore, visual information based approaches may provide more accurate geometric (shape, coordinates and orientations) information compared with the sensor-only approaches.

In this paper, we propose CrowdMap, an accurate indoor floor plan reconstruction system based on sensor-rich videos. CrowdMap generates indoor floor plans by cross-fuse visual, inertial (gyroscope, accelerometer and compass) and spatial (geo-location) information crowdsourced from the users. We jointly utilize computer vision and mobile techniques in a complementary way to manage the noisy crowdsourced data. The key idea of our system is to leverage the sequential relationship between each consecutive frame of the crowdsourced video. We employ advanced computer vision algorithms, which are able to furnish the consistent video frame relation to generate accurate spatial information of the indoor environment. Compared with uncorrelated images, CrowdMap shows that the spatio-temporal continuous video frames are able to provide more valuable information with the same amount of data in an indoor crowdsourced setting.

We solve two challenges in the design of CrowdMap. First, the sensor-rich videos uploaded by the crowd are usually

not captured with floor plan generation in mind, since we cannot assume that every user is professionally trained to our crowdsourcing task. We solve this challenge by i) utilizing inertial sensor and visual data to track the position of the smartphone's camera and ii) designing a multi-layer system based on the "divide and conquer" method to gradually filter out unqualified data. Second, existing crowdsourcing based floor plan reconstruction approaches are unable to provide room layout with a good quality. We solve this challenge by leveraging the consecutiveness in the sensor-rich video data to generate 360° room panorama. Then, we process the panorama to generate room layout with a high accuracy.

The proposed CrowdMap uses a combination of appropriate computer vision and mobile sensing techniques to accurately reconstruct indoor floor plans. We summarize our contributions as follows:

- We design crowdsourcing data collecting tasks to collect several forms of geo-spatial, visual and inertial data from the crowd.
- We select suitable computer vision techniques to exploit the sequential relationship from the video data and consequently improve the quality of the result.
- We develop a prototype CrowdMap system and evaluate it on a real-world scenario. The result shows that CrowdMap achieves a hallway shape precision around 88%, a recall around 93% and a F-measure around 90%. Moreover, we achieve on average 9.82% room area error and on average 6.5% room aspect ratio error. These evaluation results demonstrate a significant improvement in accuracy compared with other crowdsourcing floor plan reconstruction systems.

The rest of the paper is organized as follows: We begin with the problem formulation in Section II, followed by the design details in Section III. Section IV and V describe the implementation of a prototype of CrowdMap and the evaluation procedures, respectively. The limitations of CrowdMap are discussed in Section VI. In Section VII, we compare our work with related works. Section VIII concludes this paper.

II. CrowdMap: Problem Formulation

CrowdMap leverages crowdsourced sensor-rich videos to reconstruct accurate indoor digital floor plans without any building information known as a priori. The system consists of two components based on client-cloud platform structure. The first component is mobile front-end which allows user to contribute the spatial, video and inertial data by capturing sensor-rich videos. The other component is cloud backend which processes the received crowdsourced data and reconstructs floor plan.

a) At the mobile front-end, we design two data-collecting tasks to collect the spatial, video and inertial data from each individual user. The inertial and video data gathered from Stay-Rotate-Stay (SRS) and Stay-Walk-Stay (SWS) aim to track user movements and generate user trajectories.

b) At the cloud backend, we divide the floor plan generation process into three sub-processes and address each of them separately. The first sub-process is in charge of the indoor pathway reconstruction. It manipulates the sequence-based key-frame matching algorithm to aggregate user trajectories uploaded by the mobile front-end. We then reconstruct the indoor pathway by projecting the aggregated trajectories to an occupancy grid map. The purpose of the second sub-process is to reconstruct the room layout. It utilizes crowdsourced visual data to generate room panorama and applies advanced computer vision algorithm to process it. The panorama is a 360° image, which contains enough context information to generate accurate room layout. The third sub-process combines the indoor pathway and room layout to completely reconstruct the indoor floor plan.

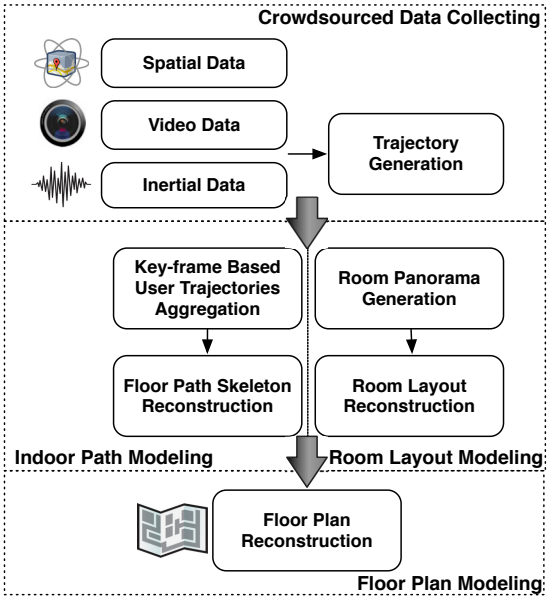


Fig. 1. The architecture of CrowdMap.

Furthermore, the CrowdMap system architecture can be categorized based on its functions. As shown in Fig. 1, CrowdMap consists of four modules:

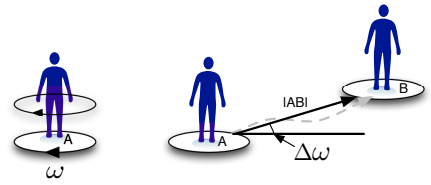
- i) crowdsourced data collection module (Section III. A)
- ii) indoor path modeling module (Section III. B)
- iii) room layout modeling module (Section III. C)
- iv) floor plan modeling module (Section III. D)

Similar to Jigsaw [11], CrowdMap is a proactive crowdsourcing system. We assume that users actively get involved in the data collecting tasks. One example could be: a user opens our mobile application and inputs the floor information (task 1); starts capturing the room environment by spinning his/her body (SRS task); then, walks towards the hallway (SWS task). We assume that several incentive mechanisms will be further developed before deploying our system in reality.

III. CrowdMap: Design Details

A. Crowdsourced Data Collecting Module

We pay careful attention to the design of the CrowdMap mobile front-end, and thereby, creating a prototype of the



(a) Stay-Rotate-Stay (SRS) Task (b) Stay-Walk-Stay (SWS) Task

Fig. 2. Video and inertial data collection task: (a) Stay-Rotate-Stay (SRS) (b) Stay-Walk-Stay (SWS)

application that runs on the Android operating system. Users are required to download and install our application on their mobile devices before they access our service. To crowdsource several forms of geo-spatial information (such as building location and floor number), visual and inertial data from the user, we design the following two data-gathering tasks:

Task 1: Geo-spatial Information Acquisition. In order to obtain the building location, we leverage the last known GPS position to estimate the user's current location. A pinpoint will be shown on a map to indicate the last known GPS location. If it is incorrect, user can correct it by simply dragging and dropping the pinpoint inside the outline of the building. Once finished, we prompt a new window asking users to input their current floor number. The geolocation data helps CrowdMap to uniquely identify the floor in a specific building. Moreover, there is no additional infrastructure needed during the first task.

Task 2: Video and Inertial Data Collection

In this task, we allow smartphone users to utilize our mobile application for capturing their preferred indoor scenes using the back camera. Our mobile application records both the video data and inertial sensor data simultaneously. During the capturing phase, the user can either move or rotate their body freely. Since the video data taken by the phone highly depends on the height of the user as well as the angle of taking the video. It requires the user to hold the phone in front of their free hand and keep the phone steady while capturing. We further decompose this task by designing two micro data-gathering tasks and modeling each of them separately (shown in Fig.2):

- **Task 2.a Stay-Rotate-Stay (SRS):** In this micro-task, the user records the video from location A, and then holds the smartphone and spins the body for a certain angle ω. According to [2], [3], the value of ω is accurate by reading the relative orientation changes from the gyroscope.
- **Task 2.b Stay-Walk-Stay (SWS):** In this micro-task, the user records the video from location A, and then holds the smartphone and walks to another location B over a period of time t. This movement can be described using a triple (x_i, y_i, t_i) to represent the location (x_i, y_i) in a local coordinate system for the user at time t_i. Therefore, through using a sequence of such triple $\{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_i, y_i, t_i), ..., (x_n, y_n, t_n)\}$, we can describe the movement of the user, which is called as the *trajectory* of the user.
  The walking distance $|AB|$ is calculated by the step counting method, which is widely applied in existing works [2], [6]. In addition, the direction change of each step $\Delta\omega$ is calculated by jointly using compass, gyroscope and accelerometer [12]. Thus, by using the inertial sensor data, we are able to reconstruct the trajectory of the user when they perform the SWS task.

B. Indoor Path Modeling Module

In order to reconstruct the path of the building floor, we first aggregate multiple user trajectories generated from the crowdsourced data collection module (SRS and SWS task) through the key-frame based user trajectories aggregation algorithm. Then, we reconstruct the floor path skeleton using the floor path skeleton reconstruction algorithm.

B.I Key-frame Based User Trajectories Aggregation: We aggregate multiple user trajectories using the video frames as "anchor points". The main challenge of this module is to achieve robust performance across a large variety of the frames that captured by different users, with different smartphone models or in different indoor environments. In order to overcome this challenge, CrowdMap adopts a sequence-based approach that we use multiple video frames along the user trajectory to aggregate multiple user trajectories. In other words, the aggregation of two or more user trajectories is determined from multiple frames over certain period of time instead of single frame comparison.

Video Key-frame Selection. During the early stages of our experimentation, we found that the bottleneck of CrowdMap is the process of crowdsourced video data, especially when using the SURF [13] algorithm to match two video frames. Therefore, this single step approach is not feasible for handling a rapidly growing influx of crowdsourced data. In order to remove the extremely similar frames and keep the frames with noticeable camera motion, we adopt the Histogram of Oriented Gradients (HOG) [14] descriptor computing algorithm.

After applying the HOG algorithm, the pairs of feature points in different gradient directions are filtered out. Next, we quantify the similarity value of two consecutive frames by the normalized cross-correlation score S_{cc}. To remove extremely similar frames, we only keep the sequence of frames that have the cardinality above a given threshold h_g. The remaining frames are called the *key-frames*.

Key-frame Comparison. We conduct video frame feature detecting and matching for the key-frame comparison. CrowdMap adopts a hierarchical approach, which takes place in two steps. In the first step, CrowdMap uses three off-the-shelf computer vision algorithms to compare two candidate video frames from three aspects, which are Color Indexing Histograms [15], [16], Shape Matching [17] and Wavelet Decomposition [18]. We assign a weight for each of the algorithm and use a linear combination of the weights to calculate the similarity score. If the similarity score S_1 is less than h_s, these two key-frames are not identical, and thereby, the two trajectories cannot be merged. This is significant

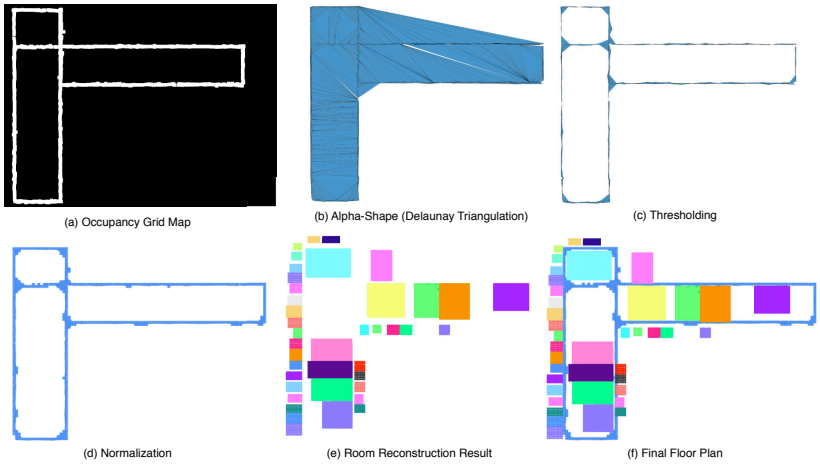


Fig. 3. Floor plan reconstruction process: (a)-(d) Floor path skeleton reconstruction, (e) room layout reconstruction, (f) the final floor plan

to prevent wrong trajectories aggregation, which impairs the accuracy of the whole system. In the second step, we select the state-of-the-art SURF [13] descriptors to precisely match two candidate key-frames. SURF algorithm is robust and fast enough in real-time processing. In our model, if it is given a pair of key-frames (I_1, I_2), we perform a match in the following manner: i) Extract two sets of descriptors $\{F_1\}$ and $\{F_2\}$ through the SURF algorithm. ii) Match these descriptors by using algorithm 1:

Algorithm 1 Key-frame Comparison Algorithm using SURF Feature

Given two sets of SURF descriptors $\{F_1\}$ and $\{F_2\}$
for $\forall f_1 \in \{F_1\}$ **do**
  $f_2 \Leftarrow NearestNeighbor(f_1, \{F_2\})$
  $f^* \Leftarrow NearestNeighbor(f_2, \{F_1\})$
  if $f^* == f_1$ **then**
    if $d(f_1, f_2) < h_d$ **then**
      add pair (f_1, f_2) to array $\mathscr{A}$
return $\mathscr{A}$

Where function $NearestNeighbor(f, \{F'\})$ returns descriptor $f' \in \{F'\}$ nearest to given f. We use Euclidean distance d as a distance metric and set distance threshold h_d for computing the quantity of good matches. Then, the similarity score is calculated using equation 1.

$$S_2(F_1, F_2) = \frac{|\mathscr{A}|}{|F_1 \bigcup F_2|} \tag{1}$$

As shown in equation 1, the distance is measured by computing the similarity score of sets F_1 and F_2. We consider F_1 matches with F_2, if the similarity score S_2 between two sets is larger than threshold h_f.

Sequence-based Aggregation. In our sequence-based aggregation algorithm, we use multiple key-frames to determine whether the two user trajectories can be merged. Our aggregation algorithm is based on the assumption that the user does not abruptly increase her walking speed above a certain limit. According this assumptions, if there is a match between the two trajectories generated in the same floor, there should be a common path between them in a high probability. Hence, we use the longest common subsequence to capture this notion. Let $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$ be the two user trajectories with length of i and j, respectively. We define the longest common subsequence metric as follows:

$$L\left(\boldsymbol{T}_{a,i}, \boldsymbol{T}_{b,j}\right) \begin{cases} 0, if\ i = 0\ or\ j = 0; \\ 1 + L(\boldsymbol{T}_{a,i-1}, \boldsymbol{T}_{b,j-1}), \\ \quad if\ d(\vec{t}_{a,i}, \vec{t}_{b,j}) \leq \epsilon\ and\ |i-j| < \delta; \\ max(L(\boldsymbol{T}_{a,i}, \boldsymbol{T}_{b,j-1}), L(\boldsymbol{T}_{a,i-1}, \boldsymbol{T}_{b,j})), \\ \quad otherwise; \end{cases}$$

where parameter δ represents the maximum length difference between two user trajectories and ϵ is the distance threshold. The similarity score S_3 for two user trajectories based on [19] is defined as follows:

$$S_3 = max_{f \in \mathscr{F}} \frac{L(\boldsymbol{T}_a, f(\boldsymbol{T}_b))}{min(i, j)} \tag{2}$$

similar to [19], $\mathscr{F}$ represents a set of all possible translations. Two user trajectories are able to be aggregated only if the similarity score S_3 for T_a and T_b is larger than h_l.

B.II Floor Path Skeleton Reconstruction: One common technique for indoor floor path representation is to use the occupancy grid [20] to approximate the environment. The occupancy grid contains a grid of square cells with fixed dimensions, which discretize the continuous 2D indoor space to represent the indoor path. Each cell is assigned with a probability value that represents how accessible the location is.

CrowdMap leverages the accessible cells to express the path skeleton of each floor. Base on this model, the floor path

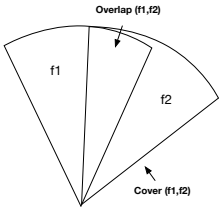


Fig. 4. The model of overlap and cover between two key-frames f1,f2

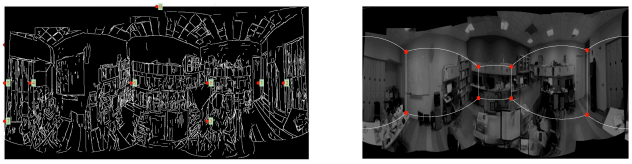


Fig. 5. Processing the room panorama to generate room layout: (a) line segments detection (b) room corner detection

skeleton reconstruction processes can be further divided into six steps. First of all, we initialize the occupancy grid map as a matrix full of zeros. Second, we map the aggregated user trajectories onto the grid map to estimate the access probability of each cell. If a cell is mapped by more than one trajectory, the probability of the cell increases. Third, a binarization technique [21] is applied to automatically calculate an optimal threshold, and then set the threshold to every cell $c_{i,j}$ (i.e. in the i^th row and j^th column of the occupancy grid map) in the map to filter out cells with low access probability. This step is used for eliminating the errors and outliers introduced by the crowdsourced data (shown in Fig. 3a). To further mark the boundaries of the floor skeleton, in the fourth step, we choose the α-shape algorithm [22], which uses Delaunary triangulation to estimate shape (shown in Fig. 3b). After that, we apply an α-threshold h_α to find the the regularized boundaries of the indoor path skeleton (shown in Fig. 3c). In the last step, we normalize the regularized boundaries by repairing the unconnected paths. The output result is the reconstructed floor path skeleton (shown in Fig. 3d).

C. Room Layout Modeling Module

In this module, we utilize crowdsourced images to create the panorama for each room, and then use computer vision techniques to process the panorama, and thereby, generate the room layout.

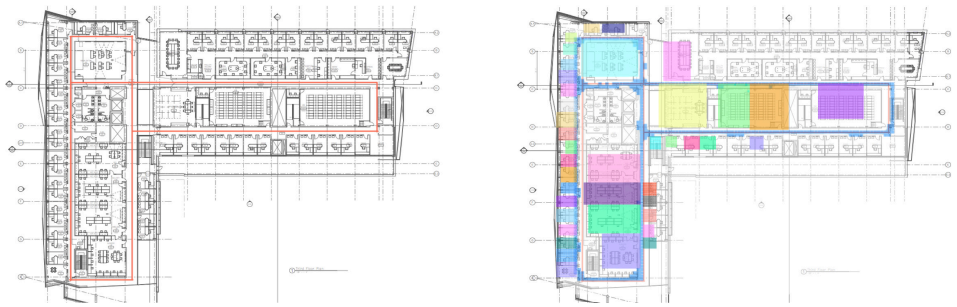
C.I Indoor Panorama Generation: In indoor path modeling module, we aggregate multiple users' trajectories using video key-frames and generate floor path skeleton based on the occupancy grid model. After aggregation, it is possible to have more than one key-frames for a certain cell $C_{i,j}$ on our path skeleton. This is due to either i) user records the scene through SRS micro-tasks at that position or ii) two or more user trajectories are merged at that position. The key-frames inside the cell $C_{i,j}$ should be geographically close to each other. Therefore, we can utilize a point panorama model to approximate their relationship, as shown in Fig. 4. In this model, given two key-frames f1 and f2, we define that the Overlap(f1,f2) is the intersecting viewing angle and the Cover(f1,f2) is the union of the viewing angle. For each key-frame, the viewing angle depends on the camera lens properties. For example, a standard 35mm lens smartphone back-facing camera has the visible angle of 54.4° for landscape mode. Therefore, through checking the direction change $\Delta\omega$ (obtained from SRS or SWS task by leveraging inertial data) of each key-frame in cell $C_{i,j}$, we are able to select a series of overlapping key-frames in different directions for cell $C_{i,j}$. For generating a 360° panorama, the candidate key-frames should satisfy the following criteria: i) every two adjacent selected key-frames should have the overlap part. ii) the selected key-frames should cover the scene in 360°. If the candidate key-frames in cell $C_{i,j}$ satisfies these two conditions, an off-the-shelf program *AutoStitch* is applied to them to generate panoramic images. The parameter set we use in CrowdMap panorama generation pipeline is JPEG quality 90 and 2048×1024 resolution in AutoStich.

C.II Room Layout Generation: A full view room panorama as the input to essentially generate the room layout in CrowdMap. It provides 360° whole room contextual information which is sufficient to reconstruct the shape of the entire room. CrowdMap chooses a 2D rectangular model for the room layout. We further discuss how to reconstruct non-rectangular shaped rooms in Section VI. According to 2D rectangular model, for each room, we need to detect the corner (edge) to generate the layout. Through using generated panoramic images, we first detect line segments with the line segment detection algorithm [23] (Shown in Fig.5a), followed by applying the Hough Transform [24] to the panorama to find the vanishing lines of these line segments. Then, five line segments are selected along the vanishing direction as the room corners (shown in Fig. 5b) to form the room layout models in 3D. We repeat the previous step to generate 20,000 room layout models for each panorama. Ultimately, the best model is selected by checking the pixel-wise surface-consistency metric [25]. Fig. 3 (e) shows the result of room reconstruction.

D. Floor Plan Modeling Module

The objective in this module is to merge the indoor path skeleton with rooms and reconstruct the building floor plan. To optimize the room layout in the floor plan, a well-established force-directed room arrangement algorithm is applied.

Force-directed Room Arrangement: Similar to [7], we choose the force-directed algorithms [26] to determine the location of each room center with the least crossing edges. This algorithm uses a spring-like model to assign attractive force F_s and repulsive force F_r between the two neighboring rooms R_a and R_b. Moreover, it gradually adjust the center of each room until the room experience net zero force. Fig. 3 (f) shows the final output of the floor plan modeling module.



(a) The Ground Truth Floor Plan

(b) Reconstructed Floor Plan and The Ground Truth

Fig. 6. The ground truth and the reconstructed floor plan

IV. CrowdMap: Implementation

A prototype of CrowdMap has been implemented in several testbeds using Google (LG) Nexus 5, Google (LG) Nexus 4, and SAMSUNG Galaxy Nexus, running Android 4.4 KitKat and four 16-core memory intensive (A9) Ubuntu Linux cloud severs at Microsoft cloud platform Microsoft Azure (a total of 64-core Intel(R) Xeon(R) E5-2670 @ 2.60GHz CPU and 448 GB RAM).

This prototype CrowdMap can be divided into two part: i) a mobile front-end prototype for Android and ii) a cloud backend, which is deployed on an infrastructure-as-a-service (IaaS) cloud using Apache Spark.

1) CrowdMap Mobile Front-end. The mobile front-end allows users to record and upload sensor-rich videos annotated with building geo-spatial information. We design a simple graphical user interface (GUI) for mobile users for letting them capturing the indoor scenes and upload data to the cloud server. The datasets are zipped and then separated into 5MB chunks for transmitting. We transmit data only when the users are using Wi-Fi connections as a default.

2) CrowdMap Cloud Backend. The CrowdMap cloud backend has two main functionalities: i) handling incoming crowdsourced data. ii) processing crowdsourced visual and inertial data and generating building floor plan. To implement the cloud backend, CrowdMap uses a set of virtual machine instances (VMIs) in the cloud configured with Ubuntu Linux and Apache Spark [27] for large-scale crowdsourced data processing.

Handling Incoming Data. We utilize a Tornado web server to handle incoming Http request. Tornado is a high performance asynchronous web server, which is capable of receiving data from a large number of users simultaneously. Our mobile clients send zipped data to Tornado via Web Sockets, a technology that allows the dataset to be sent to the cloud server in real-time.

Data Parallel-processing Pipeline. We first unzip the received data and store the raw data into MongoDB, using a non-blocking asynchronous driver to communicate with Tornado web server. Then an Advanced Python Scheduler (APScheduler) will load the data and feed it to a cascade pipeline as what we described in the previous section. Moreover, we leverage PySpark with MLlib (interoperates with NumPy) to accelerate the process of user trajectories aggregation. The reconstructed building floor plan can be downloaded directly from the website.

V. CrowdMap: Performance Evaluation

We evaluate our CrowdMap prototype in the following scenario: untrained and uncorrelated volunteers use our mobile front-end capturing indoor scenes in a typical college building to reconstruct building floor plan. We collect data on the college buildings at different times of day, and over a period of six months. Before conducting the experiment, we collect 61,243 key frames of three different buildings (Lab1 dataset, Lab2 dataset and Gym dataset) from 301 sensor-rich video sequences successfully uploaded by 25 users. Some places were captured multiple times.

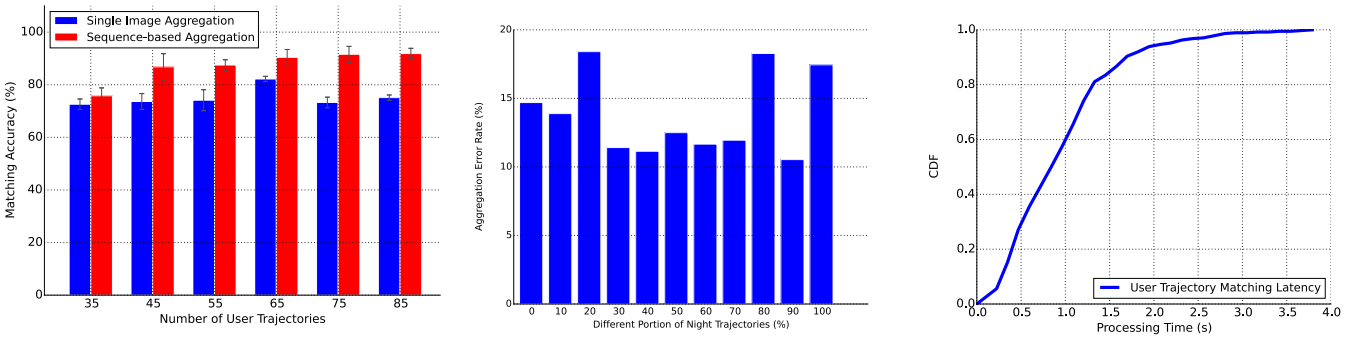
Fig. 6 demonstrates the comparison of the final reconstructed result for lab1 dataset and the ground truth. We further evaluate the performance of indoor path modeling (Section A), room layout modeling (Section B) and floor plan modeling (Section C) in the CrowdMap system. Moreover, the differences between our work and Jigsaw [11] are discussed in Section D.

A. Indoor Path Modeling Performance

Hallway Shape. We evaluate the similarity between the hallway path skeleton generated by CrowdMap with the ground truth. First, the reconstructed indoor path skeleton is overlaid onto the ground truth to achieve maximum cover area by moving and rotating the center point of the reconstructed indoor path skeleton. Then, we manually cut off part of the skeleton that belongs to the room path. In order to assess the performance of CrowdMap, the metrics below are used,

$$\mathcal{P} = \frac{|\mathcal{S}_{gen} \bigcap \mathcal{S}_{true}|}{|\mathcal{S}_{gen}|} \tag{3}$$

$$\mathcal{R} = \frac{|\mathcal{S}_{gen} \bigcap \mathcal{S}_{true}|}{|\mathcal{S}_{true}|} \tag{4}$$



(a) Impact of User Trajectories Quantities (b) Tolerance of Changes in Lighting and Exposure (c) User Trajectory Matching Latency

Fig. 7. CrowdMap Indoor Path Modeling Module Evaluation

TABLE I
HALLWAY SHAPE EVALUATION

	Precision ($\mathcal{P}$)	Recall ($\mathcal{R}$)	F-Measure ($\mathcal{F}$)
Lab 1	87.5%	93.3%	90.3%
Lab 2	92.2%	95.9%	94.0%
Gym	84.3%	88.8%	86.5%

$$\mathcal{F} = 2 * \frac{\mathcal{P} * \mathcal{R}}{\mathcal{P} + \mathcal{R}} \tag{5}$$

where $\mathcal{P}$ presents the precision of the hallway shape and we define it as the ratio of the size of the overlapped area to the whole area of the reconstruction hallway skeleton. $\mathcal{R}$ is the recall of the hallway shape, which is expressed as the ratio of the size of the overlapped area to the whole area of the ground truth hallway skeleton. $\mathcal{F}$ stands for the harmonic mean of the precision and recall. The performance evaluation results are shown in Table I, which exhibit that CrowdMap is able to achieve a precision around 88%, a recall around 93% and a F-measure around 90%. These three assessments reveal that our visual and inertial hybrid method is more robust to errors and outliers. As we select occupancy grid to approximate the indoor environment, the area of the hallway skeleton is larger than the ground truth. Therefore, we tend to have a higher recalls compared with the precision values.

Impact of User Trajectories Quantities. CrowdMap has the ability to aggregate crowdsourced user trajectories based on sequence-based aggregation method. Our method not only uses single image as an "anchor point" to fuse trajectories, but also checks several nearby frames and finds the longest common subsequence between the two trajectories. Fig. 7(a) illustrates the matching accuracy of both single image aggregation method and sequence-based aggregation method with different number of user trajectories. Obviously, our sequence-based aggregation method performs better than single image aggregation method. We also find that when the number of user trajectories data reaches above 65, the accuracy of single image aggregation method actually decreases. This is due to the reason that indoor scenes in the same floor have a high similarity. Hence, using single image only as an anchor point is insufficient and leads to errors.

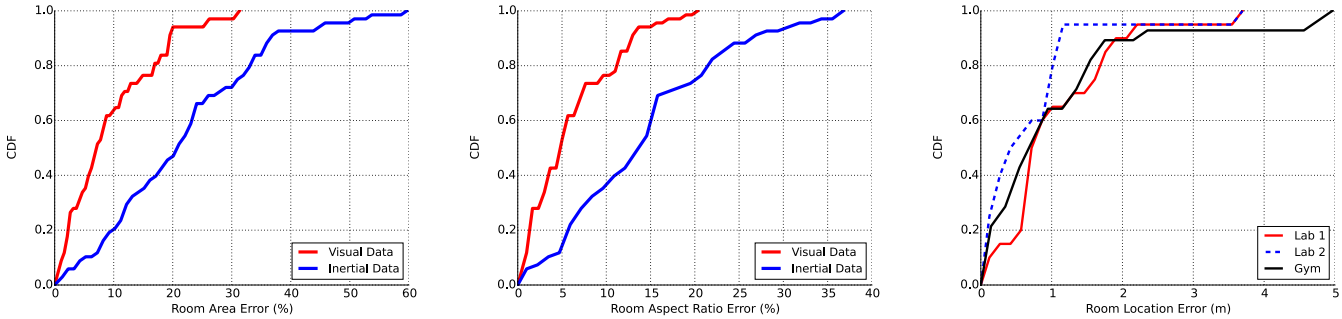
Tolerance of Changes in Lighting and Exposure. The crowdsourced video data uploaded by the users are captured at different time of a day under different lighting and exposure conditions. To assess the performance of CrowdMap under different lighting conditions, we manually choose user upload data and classify them into two categories: daylight group (primarily lighting source: sunlight, lux range: 100-500 lux) and night group (primarily lighting source: Incandescent lamp, lux range: 75-200 lux) by checking video data and time stamp. We keep the size of the two groups to be equal.

The following experiments have been performed: First, the aggregation error rate is evaluated from the daylight group. Next, we randomly switch 10% of the night group data into the daylight group, and then conduct the aggregation again. We keep doing this process until all the daylight data are switched out and the dataset becomes all night. Fig. 7(b) shows the aggregation error rate with different portion of night trajectories. The result exhibits that CrowdMap is robust to changes in lighting and exposure.

Computational Latency. Indoor path modeling module heavily relies on the process of the advanced computer vision algorithm. Fig. 7(c) plots the CDF graph of the computational latency for matching user trajectories. The result shows that our method have an average running time of 0.8 seconds for matching two key-frames (on a single-threaded setting).The majority of this time is spent on SURF feature matching. Performing the complete user trajectory aggregation algorithm (with multiple image comparisons per frame) takes around 40-50 seconds, which depends on the number of key-frames. The performance of our matching algorithm is comparable to the state-of-the-art.

B. Room Layout Modeling Performance

Room Area. CrowdMap utilizes visual information to generate the room layout. For assessing the performance of CrowdMap, room area is selected as one of our evaluation metrics. We first calculate the room area by multiplying room length and width. Then, the room area error is calculated, which is defined as the area difference between the generated room layout and the ground truth divided by the ground truth



(a) Room Area Error (b) Room Aspect Ratio Error (c) Room Location Error

Fig. 8. CrowdMap Room Layout Modeling Module and Floor Plan Modeling Module Evaluation

room area. Fig. 8(a) shows the CDF graph of room area error. The result demonstrates that our visual method achieves an average error of 9.8%. Comparing with the result of using inertial data (average 22.5% room area error), our method obtains a significant improvement. This is because our method does not require user to move across edges and corners of the room. Moreover, the wide filed-of-view panorama generated by video is able to provide enough context information to reconstruct the room layout.

Room Aspect Ratio. The room aspect ratio represents the shape of a room. We define the room aspect ratio as the length of the room divided by the width of the room, and the room aspect ratio error as the difference between generated room aspect ratio and the ground truth ratio divided by the ground truth ratio. Fig. 8(b) presents the CDF graph of room aspect ratio error. The result illustrates that our room layout generation method achieves an average error of 6.5%, which generate more accurate result compared with the inertial data (an average 15.1% room aspect ratio error).

C. Floor Plan Modeling Performance

Positions of the Room. In this module, we map the room to the indoor pathway, and thereby, build the floor plan. The performance of this floor plan generation process is quantified by using the room location error. Fig. 8(c) shows the CDF graph of room location error for three datasets. The result reveals that the average room location error for Lab 1, Lab 2 and Gym dataset is 1.2m, 1.5m, and 1.2m, respectively. Note that for the Gym environment, it has a sporadic distribution of rooms. Therefore, it is very challenging to accurately locate the room center. Hence, one room has a maximum room location error of 5m.

D. Comparison with Jigsaw

Indoor Path Modeling. For reconstructing the indoor pathway shape, CrowdMap and Jigsaw are both utilizing the visual and inertial data to aggregate multiple user trajectories, and thereby, form its shape. The difference is that: Jigsaw requires user to take photos of each landmark, and the images shoot for the same landmark are grouped together. Then, they process the bundled images to calculate the camera position

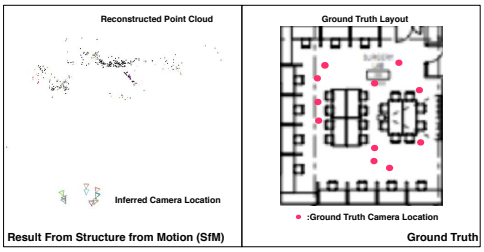


Fig. 9. Structure from Motion (SfM) Result compared with Ground Truth

(using Structure-from-Motion) and the geometry features of the pathway (using vanishing line detection). They reconstruct the pathway by fusing the camera position, geometry information of the landmarks and user trajectories together. However, CrowdMap requires user to shoot a video, and it exploits the sequential relationship reside in the video data to reconstruct indoor pathway with high accuracy. We process multiple continuous key-frames to calibrate the drift error residing in the trajectories, and then aggregate these trajectories to generate indoor floor plan.

As pointed out by [28], the state-of-the-art Structure-from-Motion (SfM) technique is not reliable when used in a highly cluttered and featureless indoor environment. Participants must have extensive experiences in shooting photos (e.g. avoid featureless objects). As shown in Fig.9, the camera locations inferred by SfM are not accurate in our dataset (a lab room inside a college building). As our video-based indoor path reconstruction method achieves the excellent performances consistently even in featureless indoor environment (shown in Table I), therefore, CrowdMap is more reliable than Jigsaw in general.

Room Layout Modeling. Jigsaw and CrowMap utilize different approach to reconstruct room layout. Jigsaw applies aggregated user trajectories to determine the room layout. Our system manipulates panorama generated by the video frame to reconstruct the room layout. As shown in Fig. 8 (a) (b), our method delivers doubled performances in terms of room area error and room aspect ratio error compared with Jigsaw. This is due to the reason that some parts of a room

are not accessible, thereby, relying only on the user motion data should not provide accurate results. However, the wide field-of-view image (360° panorama) is able to provide more context information of the scene [25], including the room layout. It is also capable to infer the edge of a room even if it is occluded by objects.

VI. Discussions and Limitations

Energy Consumption. CrowdMap front-end runs on a user's smartphone. It takes some energy when user starts capturing the indoor environment by shooting sensor-rich videos. The inertial sensor (accelerometer, compass and gyroscope) only consumes about 30mW when sampling. Recording video takes an average of 350mW [29] for a one minute recording with a resolution setting of 480p. However, unlike CrowdInside, our mobile application does not require users to run a daemon process in the background. Therefore, several rounds of data collecting tasks should not constitute significant power consumption for an user.

Reconstruct Multi-Floors in Single Round. CrowdMap only focuses on 1-floor building floor plan reconstruction. The task of constructing multiple floors can be decomposed into multiple 1-floor map constructions. One possible solution is to use stairs, elevators and escalators as special reference points and connect multiple 1-floor maps at these reference points. According to Skyloc [30], different floors can be distinguished by GSM fingerprints. We may also jointly use the acceleration patterns to tell apart corridors and stairs [2] or elevators [10].

Reconstruct Non-Rectangular Shaped Room. Our room layout reconstruction method is based on the assumption that each room fits in a rectangular shape. Because according to [31], around 90% of modern buildings have a rectangular contour. The room layout for the rectangular building tends to be consistent, and thereby, most of them are also rectangular. However, for non-rectangular room, our visual-based approach is either not working or may provide less accurate result. As our future work, we propose the following solutions: i) we jointly use user trajectories and visual-based approach to determine the room layout. ii) when encounter non-rectangular room, we ask users to label the edges of the room. We believe these two solutions could enhance the accuracy of reconstructing the layout of non-rectangular rooms.

VII. Related Work

Digital Floor Plan Construction. Digital floor plan construction is a relatively new topic in mobile computing. Most of the existing approaches focus primarily on inertial data aggregations [6], [7], [10], [32]. CrowdInside [10] is a crowdsourcing-based system for automatically and transparently construct digital indoor floor plans. It leverages smartphone inertial sensor data from accelerometer, gyroscope and compass to generate user motion traces. Also, it uses anchor points with unique sensory pattern such as elevators, stairs, escalators and locations with GPS reception to eliminate accumulated errors. The mapping algorithm is highly dependent on the crowdsourced motion traces's accuracy. Yifei Jiang et al. [7] propose a system for automatic floor plan construction using Wi-Fi signature similarities between different rooms and hallway segments. Walkie-Markie [6] also leverages Wi-Fi signals to reconstruct the room layout. However, their system fully relies on the availability of Wi-Fi fingerprints. Jigsaw [11] utilizes both image and sensory data to reconstruct the indoor floor plan and achieves a better performance. However, Jigsaw only uses images to infer the wall segments of the room entrance and still uses aggregated user motion trace and camera position to determine the shape of the room. As mentioned before, we cannot assume all edges and corners of the room could be covered with user traces as it may be inaccessible to users (e.g. blocked by desk).

Indoor Parameters Acquisition. Various studies have utilized sensor data to facilitate the determination of user parameters such as location, heading and speed in an indoor environment [1], [3], [33]–[36]. A few feature zero configuration requirements, thereby, ideal for using crowd sourced data. Location is one of the most important parameters when we are modeling an indoor environment. [34] leveraged off-the-shelf Wi-Fi infrastructure to aid the indoor localization of smart phones. Based on existing infrastructures, it requires little effort for deployment. [3] proposed a system that enables training data to be crowdsourced without any explicit effort on the part of users, and thereby, made calibration require zero effort. Another key parameter is the heading information. One traditional way to obtain this is aggregating readings from a device's accelerometers, gyroscope and compass [37]. Because this method can easily result in the accumulation of errors, it requires extensive work to determine effective calibration methods. [38] used the camera to detect heading changes by calculating the vanishing points in consecutive images. [35], on the other hand, makes the assumption that internal line-shaped objects, such as light-tubes, are either parallel or perpendicular to the building outlines. By adjusting front-camera image projection with pose estimation, it can accurately determine user headings in real time. These techniques serve as building blocks in our system, and some of their ideas are reflected in our final design.

Indoor Scene Reconstruction. Indoor scene reconstruction remains an active area of research. Work [39] used smartphones to capture a panorama of indoor scenes and allowed the users to manually fit lines onto the edges of walls. It then adopted a line-fitting algorithm based on the Manhattan World assumption to reconstruct room shapes.

Some studies have focused on specialized hardware, such as laser scanners [40], depth sensors [41] and commercial devices, e.g. Kinect [42]. Lately, Structure.io [43] and Google Project Tango [44] also manufactured 3D sensor for mobile devices. The reconstruction results of these devices are impressive, however, we argue that they are not suitable for crowdsourcing scenarios as they require specialized equipments. Many studies by the computer vision community have also focused on reconstructing indoor 3D models solely from images [45], [46]. These studies used state-of-the-art techniques like Structure from Motion [47], [48] and Multiview-stereo

[49], [50] to generate 3D models from images. Some research has shown an increase in accuracy if user-labeling information is supplied [41]. As pointed out in [28], however, these methods recommend participants with extensive experience in shooting photos, which significantly restricts their validity in crowdsourcing scenarios.

VIII. Conclusion and Future Work

This paper presents CrowdMap, an indoor floor plan reconstruction system based on crowdsourced sensor-rich videos. Our solution uses the sequential relationship between consecutive frames to enhance the accuracy of the floor plan. The prototype of our system is readily deployable at a large scale. As our future work, we will focus on further processing of the room panorama to extract more context information of the room, such as object detection and object recognition. We also plan to further study several issues related to the proposed crowdsourcing based indoor mapping approaches, such as user incentive and privacy preservation mechanism. Once fully hardened, we believe that CrowdMap is able to extend existing digital map services to indoor environment on a world scale.

IX. Acknowledgement

We thank the helpful comments from the anonymous reviewers. This work was supported in part by US National Science Foundation under grants CNS-1421903, CNS-1318948, and CNS-1262275.

References

[1] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Mobicom*, 2010.
[2] 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.
[3] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Mobicom*, 2012.
[4] G. I. Map, https://www.google.com/maps/about/partners/indoormaps/.
[5] X. Zhang, Z. Yang, C. Wu, W. Sun, and Y. Liu, "Robust trajectory estimation for crowdsourcing-based mobile applications," 2013.
[6] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: indoor pathway mapping made easy," in *NSDI*, 2013.
[7] Y. e. a. Jiang, "Hallway based automatic indoor floorplan construction using room fingerprints," in *Ubicomp*, 2013.
[8] X. Zhang, Y. Jin, H.-X. Tan, and W.-S. Soh, "Cimloc: A crowdsourcing indoor digital map construction system for localization," in *ISSNIP*, 2014.
[9] D. e. a. Philipp, "Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data," in *PerCom*, 2014.
[10] M. Alzantot and M. Youssef, "Crowdinside: automatic construction of indoor floorplans," in *SIGSPATIAL GIS*, 2012.
[11] 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 *MobiCom*, 2014.
[12] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Mobisys*, 2014.
[13] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV*, 2006.
[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
[15] M. J. Swain and D. H. Ballard, "Color indexing," *International journal of computer vision*, vol. 7, no. 1, pp. 11–32, 1991.
[16] A. Vellaikal and C.-C. J. Kuo, "Content-based image retrieval using multiresolution histogram representation," in *Photonics East*, 1995.
[17] T. Kato, T. Kurita, N. Otsu, and K. Hirata, "A sketch retrieval method for full color image database-query by visual example," in *IAPR*, 1992.
[18] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995.
[19] Y. Zheng, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *WWW*, 2009.
[20] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.
[21] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.
[22] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *Information Theory*, vol. 29, no. 4, pp. 551–559, 1983.
[23] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, 2012.
[24] P. a. Hough, "Machine Analysis Of Bubble Chamber Pictures," *Conf.Proc.*, vol. C590914, pp. 554–558, 1959.
[25] Y. Zhang, S. Song, P. Tan, and J. Xiao, "Panocontext: A whole-room 3d context model for panoramic scene understanding," in *ECCV*, 2014.
[26] P. EADES, "A heuristics for graph drawing," *Congressus Numerantium*, vol. 42, pp. 146–160, 1984.
[27] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *HotCloud*, 2010.
[28] A. e. a. Furlan, "Free your camera: 3d indoor scene understanding from arbitrary camera motion," in *BMVC*, 2013.
[29] X. Chen, Y. Chen, Z. Ma, and F. C. Fernandes, "How is energy consumed in smartphone display applications?" in *Hotmobile*, 2013.
[30] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara, "The skyloc floor localization system," in *PerCom*, 2007.
[31] P. Steadman, "Why are most buildings rectangular?" *Architectural Research Quarterly*, vol. 10, no. 02, pp. 119–130, 2006.
[32] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *ITSMC*, 2012.
[33] J. Xiong and K. Jamieson, "Arraytrack: a fine-grained indoor location system," in *Usenix NSDI*, 2013.
[34] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *TPDS*, vol. 24, no. 4, pp. 839–848, 2013.
[35] Z. Sun, S. Pan, Y.-C. Su, and P. Zhang, "Headio: zero-configured heading acquisition for indoor mobile devices through multimodal context sensing," in *UbiComp*, 2013.
[36] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Mobicom*, 2012.
[37] J. Chon and H. Cha, "Lifemap: A smartphone-based context provider for location-based services," *IEEE Pervasive Computing*, no. 2, pp. 58–67, 2011.
[38] L. Ruotsalainen, H. Kuusniemi, and R. Chen, "Heading change detection for indoor navigation with a smartphone camera," in *IPIN*, 2011.
[39] A. Sankar and S. Seitz, "Capturing indoor scenes with smartphones," in *UIST*, 2012.
[40] J. Xiao and Y. Furukawa, "Reconstructing the worlds museums," in *ECCV*, 2012.
[41] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *ICCV*, 2013.
[42] R. A. e. a. Newcombe, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR*, 2011.
[43] Structure.io, http://structure.io/.
[44] P. Tango, https://www.google.com/atap/projecttango.
[45] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *CVPR*, 2009.
[46] Y. Furukawa, B. Curless, S. M. Seitz, and R. szeliski, "Reconstructing building interiors from images," in *ICCV*, 2009.
[47] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *IJCV*, vol. 80, no. 2, pp. 189–210, 2008.
[48] N. Snavely, S. M. Seitz, and R. szeliski, "Photo tourism: exploring photo collections in 3d," *TOG*, 2006.
[49] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *PAMI*, vol. 32, no. 8, pp. 1362–1376, 2010.
[50] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *CVPR*, 2010.